Monotone iterates for nonlinear singularly perturbed convection-diffusion problems

---

A thesis submitted in partial fulfilment

of the requirements of the degree for

Doctor of Philosophy

in

Mathematics

at Massey University, Palmerston North,

New Zealand

Sophie Pack

2010

Abstract

We are interested in monotone iterative algorithms for solving nonlinear singularly perturbed convection-diffusion problems. These problems arise in many physical phenomena. One of the most common sources of these problems is the linearization of Navier-Stokes equations with large Reynolds numbers, other sources include drift-diffusion equations of semi-conductor device modelling, financial modelling, modelling in mathematical biology, fluid dynamics and heat transport problems. Singularly perturbed convection-diffusion problems are characterized by thin areas of rapid change of solutions. Many of these problems can not be solved analytically but must instead be solved numerically. Classical numerical approaches for solving these problems do not always work and may show unsatisfactory behaviours. In this thesis, we focus on constructing monotone iterative methods for solving nonlinear singularly perturbed convection-diffusion problems. Monotone difference schemes have significant advantages: they guarantee that systems of algebraic equations based on such schemes are well-posed; the finite difference operators satisfy the discrete maximum principle.

We construct a uniform convergent difference scheme for solving a nonlinear singularly perturbed two-point boundary value problem of the convection-diffusion type with discontinuous data. The uniform convergence of this scheme is proven on arbitrary meshes. A monotone iterative method is applied to computing the nonlinear difference scheme.

In the past fifteen years, much interest has been shown in domain decomposition techniques for solving singularly perturbed convection-diffusion problems. In this thesis, we construct one- and two-level monotone domain decomposition algorithms based on the multiplicative and additive Schwarz algorithms. These algorithms are proven to converge to the exact solution of the problem.

We construct monotone relaxation methods by modifying the point and block ω-Jacobi

and successive underrelaxation methods. We prove that the point and block monotone relaxation methods converge to the exact solution of the problem.

We combine the monotone domain decomposition algorithms and relaxation methods to construct composite monotone domain decomposition algorithms. These algorithms are proven to converge to the exact solution of the problem.

Multigrid methods are generally accepted as fast efficient solvers. The standard multigrid method has been shown to be unsatisfactory when applied to singularly perturbed problems. We construct monotone multigrid methods for solving nonlinear singularly perturbed convection-diffusion problems. We prove that these methods converge to the exact solution of the problem.

Acknowledgements

Firstly, I would like to thank my supervisor Prof. Igor Boglaev for his guidance throughout my honours and doctoral degree. I would like to thank him for his guidance, support, knowledge, advise and patience. He has also provided me with ideas, shown me how to write academic papers and helped with proof reading.

I would also like to thank the Institute of Fundamental Sciences at Massey University, which has given me with support, encouragement and opportunities throughout all of my university study. The institute has also supplied me with financial assistance enabling me to attend overseas conferences.

I would like to show my appreciation to the Tertiary Education Committee (TEC). The TEC has provided me with financial assistance while studying at Massey University. They have helped me financially with both living cost and travel to enable me to attend conferences.

A huge thanks would go to my husband, Thomas, and our families. I would like to thank them for supporting me, for being there in those stressful times, and for being patient in those times where it feels like all I do is study.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Singularly perturbed convection-diffusion problems

'Imagine a river - a river flowing strongly and smoothly. Liquid pollution pours into the water at a certain point. What shape does the pollution stain form on the surface of the surface of the water?

Two physical processes operate here: the pollution *diffuses* slowly through the water, but the dominant mechanism is the swift movement of the river, which rapidly *convects* the pollution downstream. Convection alone would carry the pollution along a one dimensional curve on the surface; diffusion gradually spreads that curve, resulting in a long thin wedge shape '[58].

The above is a description of a convection-diffusion problem. Convection-diffusion equations occurs when the differential equations modelling the process contain both convective and diffusive terms. Some typical examples where convection-diffusion equations occur include pollution dispersal in water or in the atmosphere, Fokker-Planck equation, semi-conductor equations, groundwater transport problems [51].

In convection-diffusion problems, the second order derivatives are multiplied by a parameter ε. In standard problems where $\varepsilon = \mathcal{O}(1)$, diffusion is the dominant term and is the main influence on the numerical solution of the problem. In general, this results in a relatively smooth solution, and classical numerical methods are usually sufficient to solve such problems. When $\varepsilon << 1$, diffusion is no longer the dominant term and the convective term strongly influences the numerical solution. The diffusion term has a significant influence on the solution in a small area or layer. Within this thin layer

the gradient of the solution is large, hence, the dependent variable undergoes very rapid changes. This is a characteristic of singularly perturbed problems. As ε is the parameter which influences this behaviour, it is known as the perturbation parameter. These thin areas of rapid change are known as boundary layers if they occur near a boundary, or interiour layers if they occur within the domain not adjoining the boundary.

Singularly perturbed convection-diffusion problems arise in many physical phenomena. One of the most common sources of these problems is the linearisation of Navier-Stokes equations with large Reynolds number. These problems also occur in drift-diffusion equation of semi-conductor device modelling, financial modelling, modelling in mathematical biology and in fluid dynamics and heat transport problems. Many more examples may be found in [42], [43] and [51]. It is due to the regular occurrence of singularly perturbed convection-diffusion problems in applied mathematics and engineering that there is much interest in finding solutions to them.

For a two dimensional problem to be elliptic, the following inequality of the quadratic form must be satisfied

$$\sum_{i,j=1}^{2} a_{ij}\xi_i\xi_j \geq \kappa(|\xi_1|^2 + |\xi_2|^2), \text{ for all } \xi_i \text{ and } \xi_j,$$

where a_{ij} are the coefficients of the second derivatives and $\kappa > 0$ is the ellipticity constant. The differential operators in convection-diffusion problems with small values of ε stretch this definition, since the ellipticity constant is close to zero. Many classical approaches for solving these convection-diffusion problems are designed for elliptic problems. When applied to singularly perturbed convection-diffusion problems, these classical approaches do not always work and may show unsatisfactory behaviour such as instability [67], failure to converge and oscillatory behaviour within the boundary layers which are not physically realistic [51]. Therefore, there is much interest in investigating different methods for solving singularly perturbed convection-diffusion problems.

Prandtl first introduced the idea of boundary layers in 1904 in the study of fluid dynamics. However, he did not delve into the mathematical complexities of the problems. Prandtl proved that a flow (such as in water or air) around a body may be split into two regions, one which adjoins the body and one which does not. It is only in the region adjoining the body that friction has a prominent effect. The following in a translation of a section of [55]:

'The physical processes in the boundary layer between fluid and solid body can be calculated in a sufficiently satisfactory way if it is assumed that the fluid adheres to the walls, so that the total velocity there is zero or equal to the velocity of the body. If the viscosity is very small and the path of the fluid along the wall is not too long, the velocity will have again its usual value very near to the wall. In the thin transition layer the sharp changes of velocity, in spite of the small viscosity coefficient, produce noticeable effects' [1].

Although Prandtl introduced the idea of singularly perturbed problems but the term singularly perturbed was not introduced until 1946, where Friedrichs and Wasow [34] introduced it while researching relaxation oscillations of the Van der Pol type.

So far there have been two main approaches for finding approximate solutions for singularly perturbed problems, one using asymptotic expansions and the other is a numerical method approach. The approach using asymptotic expansions has grown since the mid 1960's. Asymptotic expansions prove valuable information about an exact solution for a problem. The expansions provide an approximation to the solution using simple known functions. It is therefore a qualitative approach, looking at gaining insight into the behavior of families of problems. Asymptotic expansions are useful for certain classes of problems, however, Roos et al. [58] state 'asymptotic expansions may be impossible to construct or may fail to simplify the given problem and the numerical approximations are often the only option'. Numerical methods are developed for a broad range of problems, whereas asymptotic methods are developed for a comparatively narrow range of problems. For these reasons, the study of the numerical method approach has steadily gained momentum since the 1970's. The first textbook in this area [30] was published in 1980. However, very few, if any, additional textbooks were published for another 16 years, after 1996, [32], [50], [51], [57] and [58] were published.

The focus on numerical methods for solving singularly perturbed problems is to find robust layer-resolving methods. The four key properties of the robust layer-resolving numerical method are the following [32]:

- it is defined at each point in the domain,
- it is point-wise accurate,
- it is parameter-uniform,

- it is monotone.

A numerical method is said to be parameter-uniform if there exist positive numbers N_0, C, and p which are independent of N and the parameter ε, such that for all $N > N_0$ [32]

$$\sup_{0<\varepsilon\leq 1} ||U - u||_{\overline{\Omega}} \leq CN^{-p}, \tag{1.1}$$

where U and u are the discrete numerical and exact solutions, respectively, $\overline{\Omega}$ is the computational domain and N is the number of mesh points.

Most contemporary numerical methods are not robust layer-resolving when applied to singularly perturbed problems. There are two main approaches in the construction of robust layer-resolving numerical methods, they are fitted operator methods and fitted mesh methods.

Fitted operator methods were first introduced in 1955 by Allen and Southwell [29] while researching steady laminar motion of an incompressible viscous fluid past a stationary cylinder: that is, motion at speeds such that neither inertia nor viscosity can be neglected. Fitted operator methods work by replacing a standard finite difference operator with a specially designed finite difference operator which takes into account the nature of the singularly perturbed problem. An extensive amount of fitted operator methods are published in [7], [8], [30], [32] and [65].

In [32], [62] and [63], it is proved that there are classes of problems in the two dimensional case, where there is no fitted operator method which is robust and layer-resolving on uniform mesh, hence, a special nonuniform mesh must be used.

The fitted mesh approach is the second main approach in constructing the robust layer-resolving numerical methods. The first special nonuniform mesh was introduced in 1969 by Bakhvalov [2], where the mesh is a graded mesh based on a logarithmic scale. It is very fine within the boundary layers and course elsewhere. The Bakhvalov-type meshes were generalised by Vulanović [72]–[77] and Boglaev [9],[10],[12], and are known as Bakhvalov-type meshes. In 1989, Shishkin [62] suggested a simple piecewise uniform mesh. The piecewise uniform or Shishkin mesh, is uniformly fine in the boundary layers and uniformly course outside of the boundary layers. The Shishkin mesh uses *a-priori* knowledge of the derivatives to determine the transition point between the course and fine meshes. The corresponding numerical methods have been shown to be robust and layer-resolving [32], [58]. Due to the simplicity of its construction, the Shishkin mesh has

been widely used. The drawbacks of the Shishkin-type meshes are that *a-priori* knowledge of the derivative is needed to determine the transition point and at this stage they have only been applied to problems on rectangular domains [66]. An alternative approach is to use an adaptive mesh approach. The adaptive mesh technique contains a function which identifies boundary layers and adapts the mesh to suit. To do rigorous analysis of the adaptive mesh method, an *a posteriori* bound on the error in the computed solution (that is, the error bound which is expressed entirely in terms of local data of the current computed solution) is needed [58]. Many numerical techniques used to estimate the error lead to problem-dependent constants for convection-diffusion problems. This may lead to the error estimate not being very reliable due the constants being large and dominating the error estimate [40]. The adaptive mesh approach for solving singularly perturbed problems has been developed in [4]–[6],[47], [54], [59] and [78].

1.2 Domain decomposition algorithms

Over the past 15 years, with an increase in high performance parallel computers, much interest has been shown in domain decomposition techniques to help reduce the processor time and the computer memory required for solving problems. Domain decomposition algorithms are techniques for solving differential equations where the computational domain is partitioned (or decomposed) into several subdomains. The subdomains may or may not overlap. Subproblems, which are of a reduced size, are then formulated and solved on each of the subdomains while enforcing suitable continuity requirements between neighbouring subdomains [28]. The domain decomposition techniques may be applied either directly to differential equations - the continuous approach, or to discrete approximations of differential equations - the discrete approach. The domain decomposition techniques enable the use of different numerical schemes and also different kinds of equations in different subdomains [56]. This is a reason that there is interest in the domain decomposition algorithms for solving singularly perturbed problems, where the solutions have diverse behavioral change in its boundary or interior layers. Recently much interest has been shown in Schwarz-type iterative domain decomposition algorithms. The Schwarz iterative domain decomposition algorithms can be classified into two classes: multiplicative and additive [28], [56], [64]. The overlapping Schwarz algorithm or multiplicative

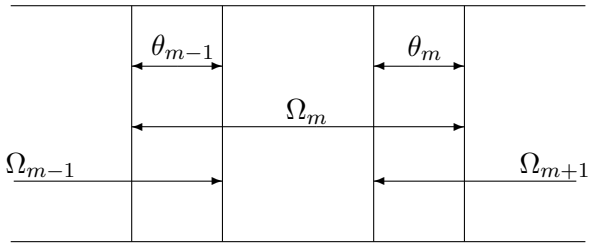

Figure 1.1: Illustration of the domain decomposition with overlapping subdomains.

Schwarz algorithm can be described in the following way. The domain Ω is split into M overlapping vertical strips Ω_m, $m = 1, \ldots, M$ (an example of this decomposition is shown in Figure 1.1). For the convection-diffusion problem, the strips are then solved in series starting at the left or the right boundary (according to upwind error propagation) [37]. This process is known as an outer iteration. The outer iterations are continued until a required accuracy is reached. The multiplicative Schwarz algorithm is a special instance of the classical or alternating Schwarz algorithm. In the alternating Schwarz algorithm, the overlapping regions are not necessarily solved in a serial fashion but may be solved in a different ordering. The other Schwarz-type iterative domain decomposition algorithm is based on nonoverlapping subdomains, which is known as the additive Schwarz algorithm. This decomposition consists of splitting the domain into M nonoverlapping strips Ω_m, $m = 1, \ldots, M$, and $M - 1$ interfacial subdomains θ_m, $m = 1, \ldots, M - 1$ (an example of this decomposition is shown in Figure 1.2) [37]. This domain decomposition algorithm has a natural way of constructing a parallel algorithm. The subdomains Ω_m, $m = 1, \ldots, M$, can be solved in parallel on their own processors, then the interfacial subproblems θ_m, $m = 1, \ldots, M - 1$, are solved in parallel on their own processors and used to update the boundary values between the strips.

We now describe the development of the domain decomposition approach for solving singularly perturbed convection-diffusion problems. A domain decomposition algorithm was first applied to a singularly perturbed problem in [31], where the discrete Schwarz algorithm is applied to one dimensional semilinear singularly perturbed problems. The continuous Schwarz algorithm on two subdomains is applied to one dimensional linear

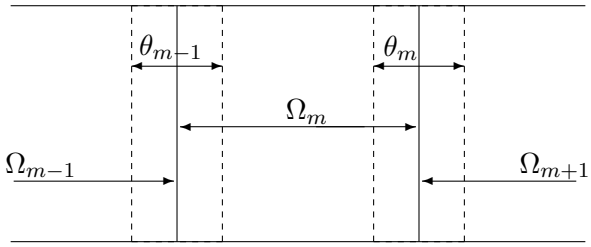

Figure 1.2: The domain decomposition with nonoverlapping subdomains.

convection-diffusion problems in [50]. The algorithm is proven to be parameter uniformly convergent. In [35] and [36], for solving two dimensional problems the asymptotic analysis is in use to determining the domain partitioning and for constructing the domain decomposition algorithm. In [49], a two dimensional linear convection-diffusion problem is solved by the continuous Schwarz algorithm on two subdomains, and the convergence rate is estimated in the maximum norm. In [16], [20] and [21], for solving two dimensional linear convection-diffusion problems, the domain decomposition algorithms are constructed on an arbitrary number of subdomains, and the convergence rate is estimated in the maximum norm. In [15], the domain decomposition algorithm on arbitrary number of subdomains is constructed and investigated for solving one dimensional nonlinear convection-diffusion problem. However, in [15] there is no discussion on solving the nonlinear problems on the subdomains.

There is not a large amount of research which has been carried out on monotone Schwarz algorithms, which combine the monotone approach with Schwarz algorithms. The advantages of the monotone Schwarz algorithms are that these algorithms solve only linear discrete systems at each iterative step of the iterative process and converge monotonically to the exact solution of the nonlinear problem. In [45] and [46], the theoretical convergence properties of the continuous monotone Schwarz algorithms applied to Poisson's equation and a nonlinear reaction-diffusion equation, respectively, are investigated. Neither of the problems in [45] and [46] are singularly perturbed and there is no discussion of the convergence properties of the discrete monotone Schwarz algorithms. In [48], it is shown that the convergence properties of the continuous Schwarz algorithm does not indicate the

same convergence properties for the corresponding discrete Schwarz algorithm. In [17], for solving two dimensional nonlinear convection-diffusion problems, the monotone Schwarz algorithm, based on the monotone approach and outer iterates from [37], is constructed, and the uniform convergence rate is estimated.

1.3 General overview of the thesis

The remainder of this chapter is as follows. In Section 1.4, we discuss uniformly convergent numerical methods for three two dimensional convection-diffusion problems. In Section 1.5, we construct a nonlinear difference scheme for solving nonlinear singularly perturbed convection-diffusion problems. The maximum principle is proven for a linear difference scheme. In Section 1.6, we construct a monotone iterative method and prove its monotone convergence. A simple method for finding initial upper or lower solutions without any prior knowledge of the solution is provided.

Chapter 2 deals with a uniform (in a perturbation parameter) convergent difference scheme for solving one dimensional nonlinear singularly perturbed two-point boundary value problem of the convection-diffusion type with discontinuous data. Uniform convergence of the proposed difference scheme on arbitrary meshes is proven. A monotone iterative method, which is based on the method of upper and lower solutions, is applied to computing the nonlinear difference scheme. Numerical experiments are presented.

Iterative domain decomposition algorithms for solving two dimensional nonlinear singularly perturbed convection-diffusion problems with a convective dominated term are discussed in Chapter 3. We consider one-level and two-level domain decomposition algorithms. We prove that the algorithms converge monotonically to the exact solution. The advantage of the two-level algorithm is that it has a potential for parallel computing. Results of numerical experiments are presented.

Chapter 4 deals with monotone relaxation methods based on the ω-Jacobi and successive underrelaxation methods. Point monotone ω-Jacobi, point monotone successive underrelaxation methods, block monotone ω-Jacobi and block monotone successive underrelaxation methods are constructed, and their monotone convergence are proven. We compare the convergence of the block monotone iterative methods. A comparison of the point monotone and block monotone iterative methods is given. Results of numerical

experiments are presented.

In Chapter 5, we describe composite monotone domain decomposition algorithms based on the Jacobi, Gauss-Seidel, block Jacobi and block Gauss-Seidel methods for solving nonlinear singularly perturbed convection-diffusion equations. These algorithms are combinations of the monotone domain decomposition algorithms presented in Chapter 3 and the monotone relaxation methods presented in Chapter 4. The advantages of these composite monotone domain decomposition algorithms are that the algorithms solve only linear discrete systems at each iterative step of the iterative process and converge monotonically to the exact solution of the nonlinear problems. Numerical experiments are presented.

In Chapter 6, we construct three monotone multigrid methods: the monotone multigrid method, the block monotone multigrid method and the two-level monotone multigrid algorithm. Monotone convergence of these algorithms is proven, and numerical experiments are presented.

Chapter 7 deals with the general overview of the conclusions from Chapters 1–6.

1.4 Uniform convergent numerical methods for two dimensional convection-diffusion problems

In this section, we consider three examples of two dimensional singularly perturbed convection-diffusion problems.

1.4.1 Convection-diffusion problem with elliptic boundary layers

$$-\varepsilon(u_{xx}+u_{yy})+b_1(x,y)u_x+b_2(x,y)u_y+f(x,y,u)=0, \quad (x,y)\in\Omega=[0,1]\times[0,1], \tag{1.2}$$

$$b_1\geq\beta_1=\text{const.}>0, \quad b_2\geq\beta_2=\text{const.}>0, \quad u(x,y)=g(x,y), \quad (x,y)\in\partial\Omega,$$

$$f_u\geq c_*=\text{const.}>0, \quad (x,y,u)\in\overline{\Omega}\times(-\infty,\infty), \quad (f_u\equiv\partial f/\partial u),$$

where $\partial\Omega$ is the boundary of $\overline{\Omega}$. Using the mean value theorem, we can write $f(x,y,u)=f(x,y,0)+f_u(x,y,\theta(x,y)u)u$, where $0<\theta(x,y)<1$. Now, we may consider (1.2) as a linear problem with the smooth coefficient f_u and use the bounds of the exact solution and its derivatives obtained in [58] for linear problems.

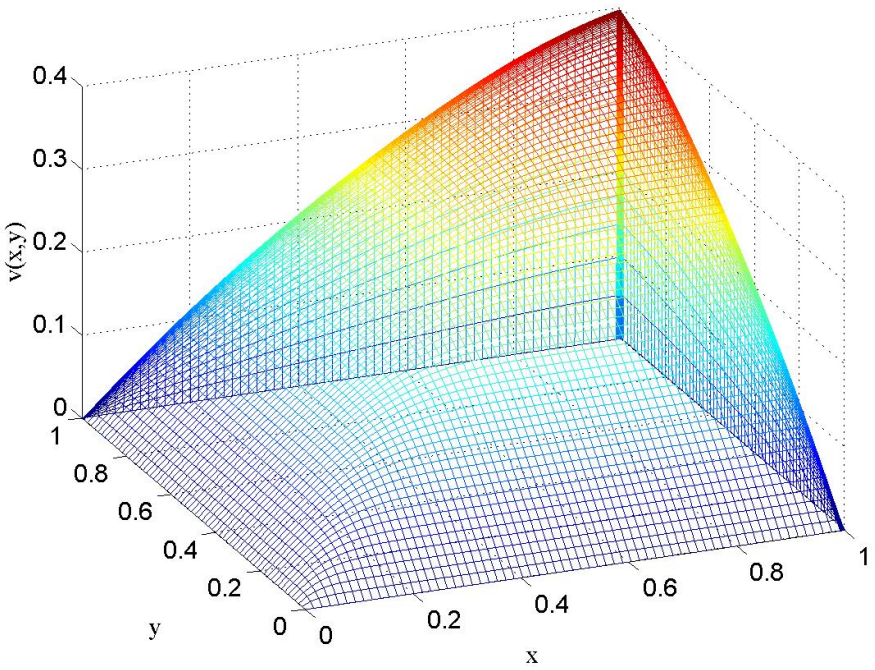

Figure 1.3: Solution of the convection-diffusion example with elliptic boundary layers.

Proposition 1.

$$\left|\frac{\partial u}{\partial x}(x,y)\right| \le C\left(1+\frac{1}{\varepsilon}\exp\left(-\beta_1\frac{(1-x)}{\varepsilon}\right)\right),$$
$$\left|\frac{\partial u}{\partial y}(x,y)\right| \le C\left(1+\frac{1}{\varepsilon}\exp\left(-\beta_2\frac{(1-y)}{\varepsilon}\right)\right),$$

where C is a constant independent of ε and $N = \max(N_x, N_y)$.

This convection-diffusion problem with elliptic boundary layers is characterised by boundary layers of width $\mathcal{O}(\varepsilon|\ln\varepsilon|)$ close to $x = 1$ and $y = 1$. The solution of (1.2) with $b_1 = b_2 = 1$, $f(x, y, u) = \exp(-1) - \exp(-u)$, $g(x, y) = 0$, and $\varepsilon = 0.001$ is given in Figure 1.3.

We introduce a nonuniform mesh $\overline{\Omega}^h = \overline{\Omega}^{hx} \times \overline{\Omega}^{hy}$:

$$\overline{\Omega}^{hx} = \{x_i,\ 0 \le i \le N_x;\ x_0 = 0,\ x_{N_x} = 1;\ h_{xi} = x_{i+1} - x_i\}, \tag{1.3}$$
$$\overline{\Omega}^{hy} = \{y_j,\ 0 \le j \le N_y;\ y_0 = 0,\ y_{N_y} = 1;\ h_{yj} = y_{j+1} - y_j\},$$

and on this mesh approximate (1.2) by the upwind difference scheme

$$\mathcal{L}v(p) + f(p, v) = 0, \quad p \in \Omega^h, \quad v(p) = g(p), \quad p \in \partial\Omega^h, \tag{1.4}$$

where the difference operator $\mathcal{L}$ is defined by

$$\mathcal{L}v = -\varepsilon(D_+^x D_-^x + D_+^y D_-^y)v + b_1 D_-^x v + b_2 D_-^y v.$$

$(D_+^x D_-^x)v(p)$ and $(D_+^y D_-^y)v(p)$ are the central difference approximations to the second derivatives and $D_-^x v$ and $D_-^y v$ are the backward difference approximations to the first derivatives with respect to x and y, respectively,

$$\begin{aligned}
&(D_+^x D_-^x)v_{ij} = (\hbar_{xi})^{-1}[(v_{i+1,j} - v_{ij})(h_{xi})^{-1} - (v_{ij} - v_{i-1,j})(h_{xi-1})^{(-1)}], \qquad (1.5)\\
&(D_+^y D_-^y)v_{ij} = (\hbar_{yj})^{-1}[(v_{i,j+1} - v_{ij})(h_{yj})^{-1} - (v_{ij} - v_{i,j-1})(h_{yj-1})^{(-1)}],\\
&D_-^x v_{ij} = (h_{xi-1})^{-1}(v_{ij} - v_{i-1,j}), \quad D_-^y v_{ij} = (h_{yj-1})^{-1}(v_{ij} - v_{i,j-1}),\\
&\hbar_{xi} = 2^{-1}(h_{xi} + h_{xi-1}), \quad \hbar_{yj} = 2^{-1}(h_{yj} + h_{yj-1}).
\end{aligned}$$

where $v_{ij} = v(x_i, y_j)$. We write the difference scheme at an interior mesh point $(x_i, y_j) \in \Omega^h$ in the form

$$d_{ij}v_{ij} - w_{ij}v_{i-1,j} - e_{ij}v_{i+1,j} - s_{ij}v_{i,j-1} - n_{ij}v_{i,j+1} + f(x_i, y_j, v_{ij}) + g_{ij}^* = 0, \qquad (1.6)$$

where g_{ij}^* is associated with the boundary function $g(p)$, and

$$\begin{aligned}
&w_{ij} = \varepsilon/(\hbar_{xi}h_{x,i-1}) + b_{1ij}/h_{x,i-1}\,, \quad e_{ij} = \varepsilon/(\hbar_{xi}h_{xi}) + b_{2ij}/h_{y,j-1}\,,\\
&s_{ij} = \varepsilon/(\hbar_{yj}h_{y,j-1})\,, \quad n_{ij} = \varepsilon/(\hbar_{yj}h_{yj})\,, \quad d_{ij} = w_{ij} + e_{ij} + s_{ij} + n_{ij}\,.
\end{aligned}$$

The coefficients of the difference scheme satisfy the inequalities

$$\begin{aligned}
&d_{ij} > 0\,, \quad w_{ij}\,, e_{ij}\,, s_{ij}\,, n_{ij} \geq 0\,, \qquad (1.7)\\
&d_{ij} - (w_{ij} + e_{ij} + s_{ij} + n_{ij}) \geq 0\,, \quad i = 1, \ldots, N_x - 1\,, \quad j = 1, \ldots, N_y - 1\,.
\end{aligned}$$

We use the piecewise uniform Shishkin mesh [32] in the x- and y-directions. As this problem has a boundary layer close to $x = 1$, we divide the x-interval $[0, 1]$ into two intervals $[0, 1 - \sigma_x]$ and $[1 - \sigma_x, 1]$. We locate half of the mesh points evenly in the boundary layer $[1 - \sigma_x, 1]$, and the remaining half are distributed evenly outside of the boundary layer $[0, 1 - \sigma_x]$. We distribute the mesh points in the y-direction in a similar way. We divide the y-interval $[0, 1]$ into two intervals $[0, 1 - \sigma_y]$ and $[1 - \sigma_y, 1]$ and locate half of the mesh points evenly in the boundary layer $[1 - \sigma_y, 1]$. The remaining half are distributed evenly outside of the boundary layer $[0, 1 - \sigma_y]$. This results in the course

mesh in the intervals $[0, 1-\sigma_x]$ and $[0, 1-\sigma_y]$ and the fine mesh in the intervals $[1-\sigma_x, 1]$ and $[1-\sigma_y, 1]$. We define h_{xi} and h_{yj} as follows

$$h_{xi} = \begin{cases} \frac{2(1-\sigma_x)}{N_x}, & 1 \le i \le \frac{N_x}{2}, \\ \frac{2\sigma_x}{N_x}, & \frac{N_x}{2}+1 \le i \le N_x, \end{cases} \quad h_{yj} = \begin{cases} \frac{2(1-\sigma_y)}{N_y}, & 1 \le j \le \frac{N_y}{2}, \\ \frac{2\sigma_y}{N_y}, & \frac{N_y}{2}+1 \le j \le N_y, \end{cases} \tag{1.8}$$

The transition point $1-\sigma_x$ and $1-\sigma_y$ are given by [50]

$$\sigma_x = \min\left\{\frac{1}{2}, \frac{2\varepsilon \ln(N_x)}{\beta_1}\right\}, \quad \sigma_y = \min\left\{\frac{1}{2}, \frac{2\varepsilon \ln(N_y)}{\beta_2}\right\}.$$

We conjecture that the upwind difference scheme (1.4) on piecewise uniform mesh (1.8) converges ε-uniformly to the solution of (1.2), that is,

$$\max_{p \in \overline{\Omega}^h} |U(p) - u(p)| \le Cd(N^{-1}), \quad N = \min(N_x, N_y),$$

where d is near order 1 in N^{-1}, and constant C is independent of ε and N. Our numerical experiments on some test problems confirm this conjecture.

1.4.2 Convection-diffusion problem with parabolic layers

$$-\varepsilon(u_{xx} + u_{yy}) + b_1(x,y)u_x + f(x,y,u) = 0, \quad (x,y) \in \Omega = [0,1] \times [0,1], \tag{1.9}$$

$$b_1 \ge \beta = \text{const.} > 0, \quad u(x,y) = g(x,y), \quad (x,y) \in \partial\Omega,$$

$$f_u \ge c_* = \text{const.} > 0, \quad (x,y,u) \in \overline{\Omega} \times (-\infty, \infty), \quad (f_u \equiv \partial f/\partial u).$$

In [75], the following bounds on the first derivatives are proven.

Proposition 2.

$$\left|\frac{\partial u}{\partial x}(x,y)\right| \le C\left(1 + \frac{1}{\varepsilon}\exp\left(-\beta\frac{(1-x)}{\varepsilon}\right)\right),$$

$$\left|\frac{\partial u}{\partial y}(x,y)\right| \le C\left(1 + \frac{1}{\sqrt{\varepsilon}}\left(\exp\left(\frac{-\sqrt{c_*}y}{\sqrt{\varepsilon}}\right) + \exp\left(\frac{-\sqrt{c_*}(1-y)}{\sqrt{\varepsilon}}\right)\right)\right),$$

where C is a constant independent of ε and $N = \max(N_x, N_y)$.

This convection-diffusion problem with parabolic layers is characterised by an elliptic boundary layer of width $\mathcal{O}(\varepsilon|\ln\varepsilon|)$ close to $x = 1$ and by two parabolic boundary layers of width $\mathcal{O}(\sqrt{\varepsilon}|\ln\varepsilon|)$ close to $y = 0$ and $y = 1$. The solution of (1.9) with $b_1 = 1$, $f(x,y,u) = \exp(-1) - \exp(-v)$, $g(x,y) = 0$, and $\varepsilon = 0.001$, is given in Figure 1.4.

We approximate (1.9) by the upwind difference scheme

$$\mathcal{L}v(p) + f(p,v) = 0, \quad p \in \Omega^h, \quad v(p) = g(p), \quad p \in \partial\Omega^h, \tag{1.10}$$

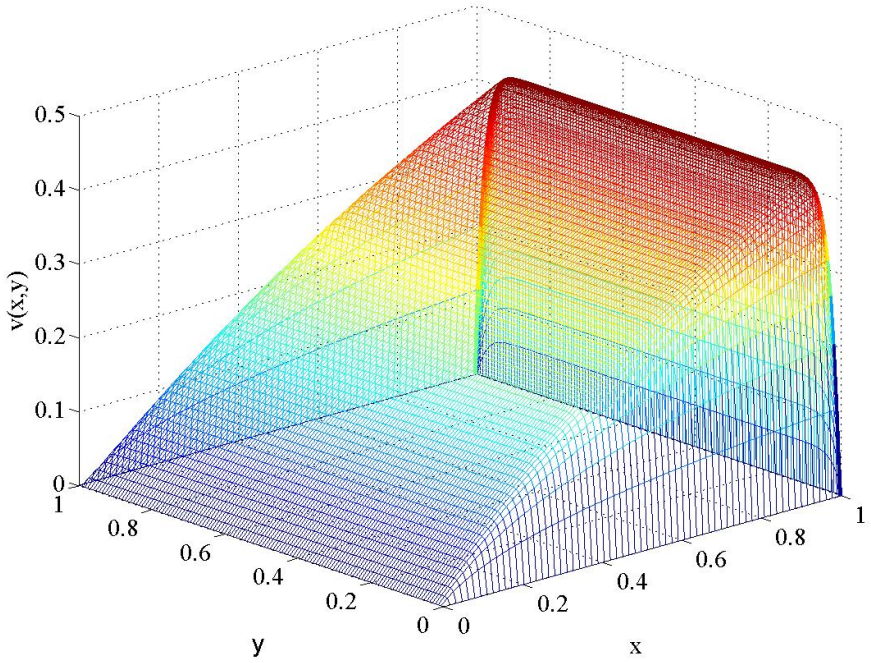

Figure 1.4: Solution of the convection-diffusion example with parabolic boundary layers.

where Ω^h from (1.3), and the difference operator $\mathcal{L}$ is defined by

$$\mathcal{L}v = -\varepsilon(D_+^x D_-^x + D_+^y D_-^y)v + b_1 D_-^x v.$$

$(D_+^x D_-^x)v(p)$ and $(D_+^y D_-^y)v(p)$ are the central difference approximations to the second derivatives and $D_-^x v$ is the backward difference approximation to the first derivative with respect to x, as defined in (1.5). We write the difference scheme at an interior mesh point $(x_i, y_j) \in \Omega^h$ in the form (1.6), where

$$w_{ij} = \varepsilon/(\hbar_{xi} h_{x,i-1}) + b_{ij}/h_{x,i-1}\,, \quad e_{ij} = \varepsilon/(\hbar_{xi} h_{xi})\,,$$
$$s_{ij} = \varepsilon/(\hbar_{yj} h_{y,j-1})\,, \quad n_{ij} = \varepsilon/(\hbar_{yj} h_{yj})\,, \quad d_{ij} = w_{ij} + e_{ij} + s_{ij} + n_{ij}\,.$$

The coefficients of the difference scheme satisfy the inequalities (1.7).

We use the piecewise uniform Shishkin mesh [32] in the x- and y-directions. As this problem has an elliptic boundary layer close to $x = 1$, we use the piecewise uniform mesh (1.8). As this problem has two parabolic layers close to $y = 0$ and $y = 1$, we divide the y-interval $[0, 1]$ into three intervals $[0, \sigma_y]$, $[\sigma_y, 1 - \sigma_y]$, and $[1 - \sigma_y, 1]$, and locate a quarter of the mesh points evenly throughout the boundary layers $[0, \sigma_y]$ and $[1 - \sigma_y, 1]$. The other half of the mesh points are distributed evenly in the interval outside of the

boundary layers $[\sigma_y, 1-\sigma_y]$. This results in the fine mesh in the boundary layers $[0, \sigma_y]$ and $[1-\sigma_y, 1]$ and the course mesh in the interval $[\sigma_y, 1-\sigma_y]$. We define h_{xi} and h_{yj} as follows

$$h_{xi} = \begin{cases} \frac{2(1-\sigma_x)}{N_x}, & 1 \le i \le \frac{N_x}{2}, \\ \frac{2\sigma_x}{N_x}, & \frac{N_x}{2}+1 \le i \le N_x, \end{cases} \quad h_{yj} = \begin{cases} \frac{4\sigma_y}{N_y}, & 1 \le j \le \frac{N_y}{4}, \\ \frac{2(1-2\sigma_y)}{N_y}, & \frac{N_y}{4}+1 \le j \le \frac{3N_y}{4}, \\ \frac{4\sigma_y}{N_y}, & \frac{3N_y}{4}+1 \le j \le N_y. \end{cases} \tag{1.11}$$

The transition points $1-\sigma_x$, σ_y and $1-\sigma_y$ are given by [32]

$$\sigma_x = \min\left\{\frac{1}{2}, \frac{2\varepsilon \ln(N_x)}{\beta}\right\}, \quad \sigma_y = \min\left\{\frac{1}{4}, \frac{\sqrt{\varepsilon}\ln(N_y)}{\sqrt{c_*}}\right\}.$$

We conjecture that the upwind difference scheme (1.10) on piecewise uniform mesh (1.11) converges ε-uniformly to the solution of (1.9), that is,

$$\max_{p \in \overline{\Omega}^h} |U(p) - u(p)| \le C d(N^{-1}), \quad N = \min(N_x, N_y),$$

where d is near order 1 in N^{-1}, and constant C is independent of ε and N. Our numerical experiments on some test problems confirm this conjecture.

1.4.3 Anisotropic convection-diffusion problem

$$-\varepsilon u_{xx} - u_{yy} + b_1(x,y)u_x + f(x,y,u) = 0, \quad (x,y) \in \Omega = [0,1]\times[0,1], \tag{1.12}$$

$$b_1 \ge \beta = \text{const.} > 0, \quad u(x,y) = g(x,y), \quad (x,y) \in \partial\Omega,$$

$$f_u \ge c_* = \text{const.} > 0, \quad (x,y,u) \in \overline{\Omega} \times (-\infty, \infty), \quad (f_u \equiv \partial f/\partial u).$$

Similar to the proof of the bounds on the first derivatives for the anisotropic convection-diffusion problem of parabolic-type from [14], one can prove the following bounds on the first derivatives of problem (1.12).

Proposition 3.

$$\left|\frac{\partial u}{\partial x}(x,y)\right| \le C\left(1 + \frac{1}{\varepsilon}\exp\left(-\beta\frac{(1-x)}{\varepsilon}\right)\right),$$

$$\left|\frac{\partial u}{\partial y}(x,y)\right| \le C,$$

where C is a constant independent of ε and $N = \max(N_x, N_y)$.

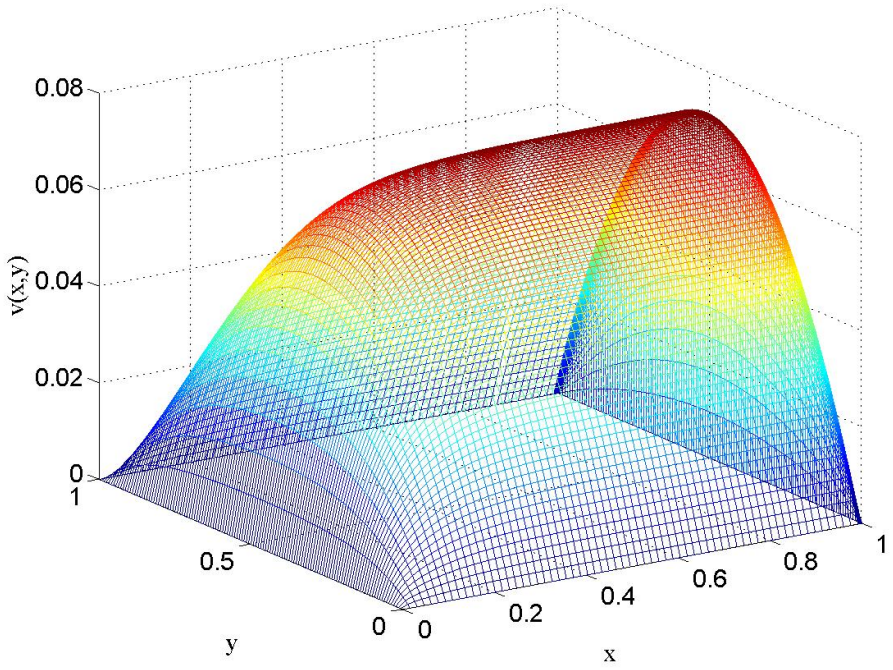

Figure 1.5: Solution of the anisotropic convection-diffusion example.

This anisotropic convection-diffusion problem is characterised by an elliptic boundary layer of width $\mathcal{O}(\varepsilon|\ln\varepsilon|)$ close to $x = 1$. The solution of (1.12) with $b_1 = 1$, $f(x, y, u) = \exp(-1) - \exp(-u)$, $g(x, y) = 0$, and $\varepsilon = 0.001$ is given in Figure 1.5.

For a mesh function $v(p)$, $p \in \Omega^h$, an approximation of (1.12) using the upwind difference scheme is given by

$$\mathcal{L}v(p) + f(p, v) = 0, \quad p \in \Omega^h, \quad v(p) = g(p), \quad p \in \partial\Omega^h, \tag{1.13}$$

where Ω^h is the mesh (1.3), and the difference operator $\mathcal{L}$ is defined by

$$\mathcal{L}v = -\varepsilon(D_+^x D_-^x)v - (D_+^y D_-^y)v + b_1 D_-^x v.$$

$(D_+^x D_-^x)v(p)$ and $(D_+^y D_-^y)v(p)$ are the central difference approximations to the second derivatives and $D_-^x v$ is the backward difference approximations to the first derivative with respect to x from (1.5). We write the difference scheme at an interior mesh point $(x_i, y_j) \in \Omega^h$ in the form (1.6), where

$$w_{ij} = \varepsilon/(\hbar_{xi}h_{xi-1}) + b_{ij}/h_{xi-1}\,, \quad e_{ij} = \varepsilon/(\hbar_{xi}h_{xi})\,,$$
$$s_{ij} = 1/(\hbar_{yj}h_{yj-1})\,, \quad n_{ij} = 1/(\hbar_{yj}h_{yj})\,, \quad d_{ij} = w_{ij} + e_{ij} + s_{ij} + n_{ij}\,.$$

The coefficients of the difference scheme satisfy the inequalities (1.7).

We use the piecewise uniform Shishkin mesh [32] in the x-direction and a uniform mesh in the y-direction. As this problem has an elliptic boundary layer close to $x = 1$, we use the piecewise uniform mesh (1.8) in the x-direction. The mesh points in the y-direction are distributed evenly over the whole interval $[0, 1]$.

We conjecture that the upwind difference scheme (1.13) on piecewise uniform mesh (1.8) in the x-direction and uniform mesh in the y-direction, converges ε-uniformly to the solution of (1.12), that is,

$$\max_{p\in\overline{\Omega}^h} |U(p) - u(p)| \le Cd(N^{-1}), \quad N = \min(N_x, N_y),$$

where d is near order 1 in N^{-1}, and constant C is independent of ε and N. Our numerical experiments on some test problems confirm this conjecture.

1.5 Nonlinear difference scheme

We consider the nonlinear convection-diffusion problem in $\mathcal{R}^k$, $k = 1, 2, \ldots$,

$$Lu + f(x, u) = 0, \quad x = (x_1, x_2, \ldots, x_k) \in \Omega, \tag{1.14}$$

$$u(x) = g(x), \quad x \in \partial\Omega,$$

where Ω is a connected bounded domain with boundary $\partial\Omega$. The differential operator L is given by

$$Lu = -\varepsilon \sum_{\alpha=1}^{k} \frac{\partial^2 u}{\partial x_\alpha^2} + \sum_{\alpha=1}^{k} b_\alpha(x) \frac{\partial u}{\partial x_\alpha},$$

where ε is a small positive parameter, and the coefficient $b_\alpha(x)$, $\alpha = 1, 2, \ldots, k$, are smooth functions.

For approximation of (1.14), we use the nonlinear difference scheme

$$\mathcal{L}v(p) + f(p, v) = 0, \quad p \in \Omega^h, \quad v(p) = g(p), \quad p \in \partial\Omega^h, \tag{1.15}$$

$$\mathcal{L}v(p) = d(p)v(p) - \sum_{p'\in\sigma'(p)} e(p, p')v(p'),$$

where $\sigma(p)$ is the stencil of the scheme at an interior point p, $\sigma'(p) = \sigma(p)\backslash\{p\}$, $\overline{\Omega}^h$ is a mesh and $\partial\Omega^h$ is the boundary of $\overline{\Omega}^h$. We assume that the mesh $\overline{\Omega}^h$ is connected, that

is, for any two interior mesh points in the domain, there exists a path through adjacent mesh points from within the domain, connecting the points. We make the following assumptions on the coefficients of the difference operator $\mathcal{L}$:

$$d(p) > 0, \quad e(p,p') \geq 0, \quad d(p) - \sum_{p' \in \sigma'(p)} e(p,p') > 0, \quad p \in \Omega^h. \tag{1.16}$$

We introduce a linear version of (1.15)

$$(\mathcal{L} + c)w(p) = f_0(p), \quad p \in \Omega^h, \tag{1.17}$$

$$w(p) = g(p), \quad p \in \partial\Omega^h, \quad c(p) \geq c_0 = \text{const} > 0, \quad p \in \overline{\Omega}^h.$$

In the following lemma, we prove the maximum principle for the difference operator $\mathcal{L} + c$ and estimate the solution of (1.17).

Lemma 1. *Let the difference operator $\mathcal{L}$ satisfy (1.16) and the mesh $\overline{\Omega}^h$ be connected.*

(i) If a mesh function $w(p)$ satisfies the conditions

$$(\mathcal{L} + c(p))w(p) \geq 0\,(\leq 0), \quad p \in \Omega^h, \quad w(p) \geq 0\,(\leq 0), \quad p \in \partial\Omega^h,$$

then $w(p) \geq 0$ (≤ 0), $p \in \overline{\Omega}^h$.

(ii) The following estimate of the solution to (1.17) holds true

$$\|w\|_{\overline{\Omega}^h} \leq \max\left[\|w\|_{\partial\Omega^h}, \|f_0\|_{\overline{\Omega}^h}/c_0\right], \tag{1.18}$$

where

$$\|w\|_{\overline{\Omega}^h} \equiv \max_{p \in \overline{\Omega}^h} |w(p)|, \quad \|w\|_{\partial\Omega^h} \equiv \max_{p \in \partial\Omega^h} |w(p)|, \quad \|f_0\|_{\overline{\Omega}^h} \equiv \max_{p \in \overline{\Omega}^h} |f_0(p)|.$$

Proof. We prove the first part of the lemma by the method of contradiction. Assume that $(\mathcal{L} + c)w(p) \geq 0$, $p \in \Omega^h$ and $w(p) \geq 0$, $p \in \partial\Omega^h$. Suppose that there exists $p \in \Omega^h$ such that $w(p) < 0$. Let W denote the set of mesh points where $w(p) = \min_{p \in \overline{\Omega}^h} w(p)$. Thus, W is the set of all the mesh points where $w(p)$ is minimal. By our assumptions, $w(p)$ is not constant, therefore, there exists $p_* \in W$ such that for some $p' \in \sigma'(p_*)$, $w(p') > w(p_*)$.

From here, (1.16) and $w(p_*) < 0$, we have

$$\begin{aligned}(\mathcal{L}+c)w(p_*) &= (d(p_*)+c(p_*))w(p_*) - \sum_{p'\in\sigma'(p_*)} e(p_*,p')w(p') \\ &= (d(p_*)+c(p_*))w(p_*) + \sum_{p'\in\sigma'(p_*)} e(p_*,p')(w(p_*)-w(p')) \\ &\quad - \sum_{p'\in\sigma'(p_*)} e(p_*,p')w(p_*) \\ &= (d(p_*) - \sum_{p'\in\sigma'(p_*)} e(p_*,p'))w(p_*) + c(p_*)w(p_*) \\ &\quad + \sum_{p'\in\sigma'(p_*)} e(p_*,p')(w(p_*)-w(p')) < 0.\end{aligned}$$

This contradicts our assumption, and we prove the result. The case of the reversed inequalities, $(\mathcal{L}+c)w(p) \le 0$, $p \in \Omega^h$ and $w(p) \le 0$, $p \in \partial\Omega^h$, may be proved in a similar manner. Thus, we prove part (i).

We now prove part (ii) of the lemma. Suppose $w(p)$ attains its maximum on the boundary $\partial\Omega^h$,

$$\|w\|_{\overline{\Omega}^h} = \|w\|_{\partial\Omega^h}.$$

Part (ii) of the lemma is satisfied. Suppose $w(p)$ attains its maximum at an interior mesh point $p^* \in \Omega^h$, where $|w(p^*)| \ge |w(p)|$, for all $p \in \overline{\Omega}^h$. From here, (1.16) and (1.17), we have

$$\begin{aligned}|(\mathcal{L}+c)w(p^*)| &= |(d(p)+c)w(p^*) - \sum_{p'\in\sigma'(p^*)} e(p,p')w(p')| \\ &\ge |(d(p^*)+c)w(p^*)| - |\sum_{p'\in\sigma'(p^*)} e(p,p')w(p')| \\ &\ge (d(p^*)+c)|w(p^*)| - \sum_{p'\in\sigma'(p^*)} e(p,p')|w(p')| \\ &\ge (d(p^*) - \sum_{p'\in\sigma'(p^*)} e(p,p'))|w(p^*)| + c|w(p^*)| \\ &\ge c|w(p^*)|.\end{aligned}$$

From here, it follows that

$$c|w(p^*)| \le \|f_0\|.$$

Taking into account that $\|w\|_{\overline{\Omega}^h} = w(p^*)$, and $c \ge c_0$, we prove part (ii) of the lemma. □

We say that $\overline{v}(p)$ is an upper solution of (1.15) if it satisfies the inequalities

$$\mathcal{L}\overline{v}(p) + f(p,\overline{v}) \ge 0, \quad p \in \Omega^h, \quad \overline{v} \ge g \quad \text{on } \partial\Omega^h.$$

Similarly, $\underline{v}(p)$ is called a lower solution if it satisfies all the reversed inequalities.

Lemma 2. *Upper and lower solutions of (1.15) satisfy the inequality*

$$\underline{v}(p) \le \overline{v}(p), \quad p \in \overline{\Omega}^h. \tag{1.19}$$

Proof. Let $\delta v = \overline{v} - \underline{v}$. From the definition of lower and upper solutions and the mean value theorem, we have

$$\begin{aligned} \mathcal{L}\overline{v}(p) + f(p,\overline{v}) - \Big(\mathcal{L}\underline{v}(p) + f(p,\underline{v})\Big) &\ge 0 \\ \mathcal{L}(\overline{v}(p) - \underline{v}(p)) + f(p,\overline{v}) - f(p,\underline{v}) &\ge 0 \\ \mathcal{L}\delta v(p) + f_v(p)\delta v(p) &\ge 0, \quad p \in \Omega^h, \end{aligned}$$

where $f_v(p) \equiv f_v[p, \widehat{v}(p)]$, $\widehat{v}(p)$ lies between $\underline{v}(p)$ and $\overline{v}(p)$. Therefore $\delta v(p)$ satisfies the conditions of the maximum principle in Lemma 1. Applying the maximum principle, we have

$$\delta v(p) = \overline{v}(p) - \underline{v}(p) \ge 0, \quad p \in \overline{\Omega}^h,$$

and prove the lemma. □

1.6 Monotone iterative method

We assume that the mesh function $f(p, v)$ satisfies the two-sided constraint

$$0 < c_* \le f_u \le c^*, \quad c_*, c^* = \text{const}. \tag{1.20}$$

We select an initial mesh function $v^{(0)}$ on $\overline{\Omega}^h$ such that it satisfies the boundary condition $g(p)$ on $\partial\Omega^h$. The monotone iterative method is given by the following recurrence formulae:

$$(\mathcal{L} + c^*)z^{(n)}(p) = -\mathcal{R}(p, v^{(n-1)}), \quad p \in \Omega^h, \tag{1.21}$$

$$z^{(n)}(p) = 0, \quad p \in \partial\Omega^h, \quad v^{(n)}(p) = z^{(n)}(p) + v^{(n-1)}(p), \quad p \in \overline{\Omega}^h,$$

$$\mathcal{R}(p, v^{(n-1)}) = \mathcal{L}v^{(n-1)}(p) + f(v^{(n-1)}, p).$$

Remark 1. For simplicity, we require our initial solution satisfies the boundary condition $g(p)$ on $\partial\Omega^h$, however this is not essential. Instead, we can only require that

$$\overline{v}^{(0)}(p) \ge g(p), \quad \underline{v}^{(0)}(p) \le g(p), \quad p \in \partial\Omega^h.$$

In this case, $z^{(n)}(p)$ is now defined by

$$z^{(1)}(p) = g(p) - \overline{v}^{(0)}(p), \quad \text{or } z^{(1)}(p) = g(p) - \underline{v}^{(0)}(p), \quad p \in \partial\Omega^h,$$

$$z^{(n)}(p) = 0, \quad n \geq 2, \quad p \in \partial\Omega^h.$$

Using this definition, we see that

$$v^{(1)}(p) = v^{(0)}(p) + z^{(1)}(p) = v^{(0)}(p) + (g(p) - v^{(0)}(p)) = g(p), \quad p \in \partial\Omega^h.$$

Theorem 1. *Let $\overline{v}^{(0)}, \underline{v}^{(0)}$ be upper and lower solutions of (1.15). Assume that the difference operator $\mathcal{L}$ and function f satisfy (1.16) and (1.20), respectively, and the mesh $\overline{\Omega}^h$ is connected. Then the upper and lower sequences $\{\overline{v}^{(n)}\}$ and $\{\underline{v}^{(n)}\}$ generated by the iterative method (1.21) converge monotonically from, respectively, above and below, to the unique solution v of (1.15):*

$$\underline{v}^{(n-1)}(p) \leq \underline{v}^{(n)}(p) \leq v(p) \leq \overline{v}^{(n)}(p) \leq \overline{v}^{(n-1)}(p), \quad p \in \overline{\Omega}^h, \quad n \geq 1.$$

Proof. We only consider the case of upper solutions. The case of lower solutions may be shown in a similar manner.

Assume that $\overline{v}^{(0)}(p)$ is an upper solution which satisfies the boundary condition. Then from (1.21) with $n = 1$, we have

$$(\mathcal{L} + c^*)z^{(1)}(p) = -\mathcal{R}(p, \overline{v}^{(0)}) \leq 0, \quad p \in \Omega^h, \quad z^{(1)}(p) = 0, \quad p \in \partial\Omega^h.$$

By the maximum principle from Lemma 1, we conclude that $z^{(1)}(p) \leq 0$, $p \in \overline{\Omega}^h$. Using the recurrence formula (1.21) and the mean value theorem, we have

$$\begin{aligned}
\mathcal{R}(p, v^{(1)}) &= \mathcal{R}(p, \overline{v}^{(0)} + z^{(1)}) \\
&= \mathcal{L}\overline{v}^{(0)}(p) + \mathcal{L}z^{(1)}(p) + f(p, \overline{v}^{(0)} + z^{(1)}) \\
&= \mathcal{L}\overline{v}^{(0)}(p) + f(p, \overline{v}^{(0)}) + \mathcal{L}z^{(1)}(p) + f(p, \overline{v}^{(0)} + z^{(1)}) - f(p, \overline{v}^{(0)}) \\
&= \mathcal{R}(p, \overline{v}^{(0)}) + \mathcal{L}z^{(1)}(p) + f_v^{(1)}(p)z^{(1)}(p) \\
&= -(\mathcal{L} + c^*)z^{(1)}(p) + \mathcal{L}z^{(1)}(p) + f_v^{(1)}(p)z^{(1)}(p) \\
&= -(c^* - f_v^{(1)}(p))z^{(1)}(p),
\end{aligned}$$

where $f_v^{(1)}(p) = f_v[p, \overline{v}^{(0)}(p) + \Theta^{(1)}(p)z^{(1)}(p)]$, $0 < \Theta^{(1)}(p) < 1$. From here and (1.20), we obtain

$$\mathcal{R}(p, v^{(1)}) = -(c^* - f_v^{(1)}(p))z^{(1)}(p) \geq 0,$$

and conclude that $v^{(1)}(p)$ is an upper solution. By induction on n, we can prove that $z^{(n)}(p) \leq 0$, $p \in \overline{\Omega}^h$, and $\overline{v}^{(n)}(p)$ is an upper solution for all $n \geq 0$. Thus, $\{\overline{v}^{(n)}\}$ is a monotonically decreasing sequence of upper solutions. This sequence is bounded below by $\underline{v}$, where $\underline{v}$ is any lower solution (1.19). We conclude that the mesh function defined by

$$v(p) = \lim_{n\to\infty} \overline{v}^{(n)}(p), \quad p \in \overline{\Omega}^h,$$

exists.

We now prove that $v(p)$ is the exact solution to (1.15). From (1.21), it follows that

$$\lim_{n\to\infty} (\mathcal{L} + c^*) z^{(n)}(p) = \lim_{n\to\infty} -\mathcal{R}(p, v^{(n-1)}).$$

As $\lim_{n\to\infty} z^{(n)}(p) \to 0$, $p \in \overline{\Omega}^h$,

$$\lim_{n\to\infty} -\mathcal{R}(p, v^{(n-1)}) = -\mathcal{R}(p, v) = 0.$$

Thus, $\mathcal{L}v(p) + f(p, v) = 0$, $p \in \Omega^h$, and taking into account that $v(p) = g(p)$, $p \in \partial\Omega^h$, we prove that $v(p)$ is the exact solution of (1.15).

We now prove that $v(p)$ is the unique solutions of (1.15). Suppose there exists another solution $\widehat{v}(p)$ of (1.15). Let $\varsigma(p) = v(p) - \widehat{v}(p)$, $p \in \overline{\Omega}^h$. By the mean value theorem,

$$\begin{aligned} \mathcal{R}(p, v(p)) - \mathcal{R}(p, \widehat{v}(p)) &= \mathcal{L}v(p) + f(p, v(p)) - \mathcal{L}\widehat{v}(p) - f(p, \widehat{v}(p)) \\ &= \mathcal{L}(v(p) - \widehat{v}(p)) + f(p, v(p)) - f(p, \widehat{v}(p)) \\ &= \mathcal{L}\varsigma(p) + f_v^*\varsigma(p) = 0, \quad p \in \Omega^h, \end{aligned}$$

where $f_v^* = f_v(p, v^*(p))$, $v^*(p)$ lies between $v(p)$ and $\widehat{v}(p)$. Since $\varsigma(p) = 0$, $p \in \partial\Omega^h$, by the maximum principle from Lemma 1, we conclude that

$$\varsigma(p) = v(p) - \widehat{v}(p) = 0, \quad p \in \overline{\Omega}^h,$$

and prove the uniqueness of the solution $v(p)$ and the theorem. □

Remark 2. An advantage of the monotone iterative method is that we can provide a simple method of finding initial upper or lower solutions without any prior knowledge of the solution. Let $t(p)$ be an arbitrary mesh function defined on $\overline{\Omega}^h$ which satisfies the boundary condition $t(p) = g(p)$ on $\partial\Omega^h$. We consider the difference problems

$$\mathcal{L}z_\nu(p) + c_* z_\nu(p) = \nu|\mathcal{R}(t, q)|, \quad p \in \Omega^h, \quad z_\nu = 0, \quad p \in \partial\Omega^h, \quad \nu = 1, -1, \tag{1.22}$$

where $\mathcal{R}(t, q) = \mathcal{L}t(p) + f(t, q)$. Then $\overline{v}^{(0)}(p) = t(p) + z_1(p)$, $\underline{v}^{(0)}(p) = t(p) + z_{-1}(p)$, $p \in \overline{\Omega}^h$, are initial upper and lower solutions, respectively.

Proof. We consider only the case for the initial upper solution, as the lower solution case may be shown in a similar manner. By the maximum principle from Lemma 1, we conclude that $z_1(p) \geq 0$, $p \in \overline{\Omega}^h$. From here and (1.22), by the mean value theorem, we have

$$\begin{aligned}
\mathcal{L}\overline{v}^{(0)}(p) + f(p, \overline{v}^{(0)}) &= \mathcal{L}(t(p) + z_1(p)) + f(p, t + z_1) \\
&= \mathcal{L}t(p) + f(p, t + z_1) + \mathcal{L}z_1(p) \pm c_* z_1(p) \\
&= \mathcal{L}t(p) + f(p, t + z_1) \pm f(t, p) \\
&\quad + |\mathcal{L}t(p) + f(t, p)| - c_* z_1(p) \\
&= \mathcal{L}t(p) + f(p, t) + |\mathcal{L}t(p) + f(t, p)| \\
&\quad + f_v^* z_1(p) - c_* z_1(p) \\
&= \mathcal{L}t(p) + f(p, t) + |\mathcal{L}t(p) + f(t, p)| \\
&\quad + (f_v^* - c_*) z_1(p) \geq 0,
\end{aligned}$$

where $f_v^* = f_v(p, v^*(p))$, $v^*(p)$ lies between $t(p)$ and $t(p) + z_1(p)$. From here, we conclude that $\overline{v}^{(0)}(p)$ is an upper solution. □

Chapter 2

One dimensional convection-diffusion problem

This chapter deals with a uniform (in a perturbation parameter) convergent difference scheme for solving a nonlinear singularly perturbed two-point boundary value problem of the convection-diffusion type with discontinuous data. Construction of the difference scheme is based on locally exact schemes or on local Green's functions. Uniform convergence of the proposed difference scheme on arbitrary meshes is proven. A monotone iterative method, which is based on the method of upper and lower solutions, is applied to computing the nonlinear difference scheme. Numerical experiments are presented.

2.1 Introduction

In this chapter, we are interested in the semilinear two-point boundary-value problem with a convective dominated term and discontinuous data

$$-\varepsilon u^{''} + b(x)u^{'} + c(x,u) + f(x) = 0, \quad x \in \Omega = (0,1), \tag{2.1}$$

$$u(0) = 0,\ u(1) = 0, \quad b(x) \geq b_* = \text{const} > 0,\ c_u \geq 0,\ (c_u \equiv \partial c/\partial u),$$

where ε is a small positive parameter. Suppose that the function c is sufficiently smooth and b, f are piecewise smooth functions, i.e.

$$b(x), f(x) \in Q_p^n(\overline{\Omega}),\ n \geq 0.$$

We say that $v(x) \in Q_p^n(\overline{\Omega})$ if it is defined on $\overline{\Omega}$ and has derivatives up to order n, the function itself and its derivatives may only have jump discontinuities at a finite set of points $p = \{p_1, \ldots, p_J\}, 0 < p_j < p_{j+1}, j = 1, \ldots, J-1$, i.e., $Q_p^n(\overline{\Omega}) = C^n(\overline{\Omega} \backslash p)$.

The solution to (2.1) is a function with a continuous first derivative, which satisfies the boundary conditions and the equation everywhere, with the exception of the points in p. The problem (2.1) has a unique solution [70]

$$u(x) \in C^1(\overline{\Omega}) \cap Q_p^{n+2}(\overline{\Omega}).$$

Linear versions of problem (2.1) with discontinuous data are investigated in [19], [33]. The solution of the linear problem possesses a strong boundary layer at $x = 1$ and weak interior layers at the points of discontinuity p. The boundary layer is strong in the sense that the solution is bounded, but the magnitude of its first derivative at $x = 1$, grows unboundedly as $\varepsilon \to 0$. The interior layers at p are weak: the solution and the first derivative are bounded but the magnitude of the second derivative grows unboundedly as $\varepsilon \to 0$. We show (see Lemma 3) that problem (2.1) possesses a strong boundary layer at $x = 1$, and the solution and the first derivative are bounded at the points of discontinuity p.

Our goal is to construct an ε-uniform numerical method (1.1) for solving problem (2.1). In [19], [33], for solving the linear version of problem (2.1), the uniform numerical methods are constructed by using the integral-difference method (or the method of locally exact schemes) on arbitrary nonuniform meshes [19], and by using the standard upwind finite difference method on the piecewise uniform mesh, which is fitted to boundary and interior layers [33].

In Section 2.2, we establish some *a priori* estimates of the solution and its first derivative. In Section 2.3, we construct a numerical method by applying the integral-difference approach. Note that in the constructed numerical method, a difference operator corresponding to the linear differential operator $-\varepsilon d^2/dx^2 + bd/dx$ is equivalent to the upwind finite volume method from [44], [79]. In Section 2.4, we prove uniform convergence of the numerical method on arbitrary nonuniform meshes by extending in a natural way the proof of the main theoretical result from [13] (the difference scheme in the case of problem (2.1) with smooth data converges ε-uniformly). In Section 2.5, we construct a monotone iterative method for solving the nonlinear difference scheme and prove that the iterates

converge ε-uniformly to the solution of problem (2.1). In Section 2.6, numerical results are presented, which are in agreement with the theoretical results.

Most of the content of this chapter is published in [25].

2.2 Properties of the continuous problem

The following lemma contains *a priori* estimates of the solution to problem (2.1).

Lemma 3. *If $b(x), f(x) \in Q_p^n(\overline{\Omega})$, $n \geq 0$, then a unique solution to (2.1) exists and $u(x) \in C^1(\overline{\Omega}) \cap Q_p^{n+2}(\overline{\Omega})$. The solution $u(x)$ satisfies the following estimates:*

$$\left| \frac{d^k u(x)}{dx^k} \right| \leq C \left[1 + \varepsilon^{-k} \exp\left(-\frac{b_*(1-x)}{\varepsilon} \right) \right], \ x \in \overline{\Omega}, \quad k = 0, 1,$$

here and throughout the chapter, C denotes a generic positive constant independent of ε.

Proof. The result that problem (2.1) with the piecewise smooth functions b and f has a unique solution can be found in [70].

Firstly, we estimate the solution $u(x)$ to (2.1). The transformation $u(x) = e^{\gamma x} w(x)$ with a positive constant γ yields the equation and the boundary conditions

$$-\varepsilon w'' + \tilde{b}(x) w' + \tilde{c}(x, w) + e^{-\gamma x} f = 0, \quad w(0) = w(1) = 0,$$

$$\tilde{b} = b - 2\varepsilon\gamma, \quad \tilde{c}(x, w) = e^{-\gamma x} c(x, e^{\gamma x} w) + (b\gamma - \varepsilon\gamma^2) w.$$

If we choose $\gamma = b_*/4$ and assume that $\varepsilon \leq 1$, then

$$\tilde{b}(x) \geq \tilde{b}_* = b_*/2, \quad \tilde{c}_w \geq \tilde{c}_* = (3/16) b_*^2.$$

If $w(x)$ is the exact solution of the above problem, then by the mean-value theorem, we can represent $\tilde{c}(x, w)$ in the form

$$\tilde{c}(x, w) = \tilde{c}(x, 0) + \tilde{c}_w(x) w(x),$$

where $\tilde{c}_w(x) = \tilde{c}_w\left(x, \theta(x) w(x)\right)$, $0 < \theta(x) < 1$. Assuming that $\tilde{c}_w(x)$ is given as a function of x, then the solution $w(x)$ may be considered as a solution of the linear problem

$$\tilde{L}_\varepsilon w \equiv -\varepsilon w'' + \tilde{b}(x) w' + \tilde{c}_w(x) w = -\tilde{f}(x), \quad x \in \Omega = (0, 1), \tag{2.2}$$

$$w(0) = w(1) = 0, \quad \tilde{f} = \tilde{c}(x, 0) + e^{-\gamma x} f, \quad \tilde{b} \geq \tilde{b}_* > 0, \quad \tilde{c}_w \geq \tilde{c}_* > 0.$$

We now prove that the maximum principle for the differential operator $\tilde{L}_\varepsilon$ with the piecewise smooth coefficients holds true: if $w(x) \in C^1(\overline{\Omega}) \cap Q_p^{n+2}(\overline{\Omega})$ and satisfies $\tilde{L}_\varepsilon w(x) \geq 0,\ x \in \Omega,\ w(0), w(1) \geq 0$, then $w(x) \geq 0,\ x \in \overline{\Omega}$. Suppose to the contrary that there is a point x where $w(x) < 0$. Let W denote the set of mesh points where $w(x) = \min_{x\in\overline{\Omega}} w(x)$. Thus, W is the set of all the mesh points where $w(x)$ is minimal. By our assumption, there exists $x_* \in W$ such that $w(x_*) < 0$. If $x_* \notin p$, where p is the set of the points of discontinuity, then from $w^{'}(x_*) = 0$ and $w^{''}(x_*) \geq 0$, it follows that $\tilde{L}_\varepsilon w(x_*) < 0$, so we get the contradiction with our assumption. Now suppose that $x_* \in p$. Since $w^{'}(x)$ is a continuous function then $w^{'}(x_*) = 0$ and $w^{'}(x) \leq 0$ in some small vicinity $[x_* - \delta, x_*]$, $\delta > 0$. In general, $w^{''}(x)$ has a jump point at x_*, but on the interval $[x_* - \delta, x_*)$, it is a continuous function. Now, if δ is small enough, then $\tilde{b}$, $\tilde{c}_w$ and $\tilde{f}$ are continuous functions and $w^{''}(x)$ does not change a sign in this interval. Representing $w^{'}(x)$ in the form $w^{'}(x) = -\int_x^{x_*-0} w^{''}(s)ds$, we conclude that $w''(x) \leq 0,\ x \in [x_* - \delta, x_*)$. Hence, $\tilde{L}_\varepsilon w(x) < 0,\ x \in [x_* - \delta, x_*)$, that contradicts our assumption. The uniform estimate on the solution $w(x)$ of problem (2.2) is derived by applying the maximum principle to the functions $-\max_{x\in\Omega} |\tilde{f}(x)|/\tilde{c}_* \pm w(x)$. Taking into account that $u(x) = \exp(\gamma x)w(x)$, we prove the uniform estimate on $u(x)$.

We now prove the estimate on $u^{'}(x)$. Representing the differential equation from (2.1) in the linear form (2.2)

$$-\varepsilon u^{''} + b(x)u^{'} + c_u(x)u + \hat{f}(x) = 0, \quad \hat{f}(x) = c(x,0) + f(x),$$

we can prove the estimate on $u^{'}(x)$ in the same way as in [19]. □

Consider the problem

$$-\varepsilon v^{''}(x) + \bar{b}(x)v^{'}(x) + c(x,v) + \bar{f}(x) = 0,\ x \in \Omega = (0,1), \tag{2.3}$$

$$v(0) = 0, \quad v(1) = 0, \quad \bar{b}(x) \geq b_*,$$

where c and b_* are defined in (2.1).

Lemma 4. *In (2.1), (2.3), let $b(x), \bar{b}(x), f(x), \bar{f}(x) \in Q_p^n(\overline{\Omega})$, $n \geq 0$. Then for $z(x) = u(x) - v(x)$ the following estimate holds:*

$$\max_{x\in\overline{\Omega}} |z(x)| \leq C \left(\sup_{x\in\overline{\Omega}} |b(x) - \bar{b}(x)| + \sup_{x\in\overline{\Omega}} |f(x) - \bar{f}(x)| \right),$$

where $u(x)$, $v(x)$ are the solutions to (2.1) and (2.3), respectively, a constant C is independent of ε.

Proof. Introduce Green's function of the differential operator $L_\varepsilon^b = -\varepsilon d^2/dx^2 + bd/dx$:

$$G(x,s) = \frac{1}{-\varepsilon w(s)} \begin{cases} \varphi^{II}(x)\varphi^I(s), & 0 \le x \le s \le 1, \\ \varphi^{II}(s)\varphi^I(x), & 0 \le s \le x \le 1, \end{cases}$$

$$\varphi^I(x) = l(x)/l(0), \quad \varphi^{II}(x) = 1 - \varphi^I(x), \quad l(x) = \int_x^1 e(s)ds,$$

$$e(s) = \exp\left(-\varepsilon^{-1}\int_s^1 b(\tau)d\tau\right), \quad w(s) = -e(s)/l(0).$$

The functions $\varphi^I(x), \varphi^{II}(x)$ are the solutions of the problems

$$L_\varepsilon^b \varphi^{I,II} = 0,\ x \in \Omega,\ \varphi^I(0) = \varphi^{II}(1) = 1,\ \varphi^I(1) = \varphi^{II}(0) = 0.$$

From the definition of $G(x,s)$, one can conclude that $G(x,s) \ge 0$.

Now we prove the uniform in the small parameter estimate

$$\max_{x\in\overline{\Omega}} \int_0^1 G(x,s)ds \le C. \tag{2.4}$$

Using the explicit formula for $G(x,s)$, we get

$$\begin{aligned} \int_0^1 G(x,s)ds &= \frac{1}{\varepsilon}\int_0^x \frac{l(x)(l(0)-l(s))}{l(0)e(s)}ds + \\ &\quad \frac{1}{\varepsilon}\int_x^1 \frac{(l(0)-l(x))l(s)}{l(0)e(s)}ds. \end{aligned}$$

From here, it follows that

$$\int_0^1 G(x,s)ds \le \frac{2}{\varepsilon}\int_0^1 \frac{l(0)-l(s)}{e(s)}ds = \frac{2}{\varepsilon}\int_0^1 g(s)ds,$$

$$g(s) = \exp\left(\varepsilon^{-1}\int_s^1 b(t)dt\right)\int_0^s \exp\left(-\varepsilon^{-1}\int_y^1 b(t)dt\right)dy.$$

The function $g(s)$ is the solution of the initial value problem

$$g'(s) = -\frac{b(s)}{\varepsilon}g(s) + 1, \quad g(0) = 0.$$

From the maximum principle for this initial value problem, we obtain the estimate

$$\max_{s\in\overline{\Omega}} |g(s)| \le \varepsilon/b_*.$$

From here, we conclude (2.4) with $C = 2/b_*$.

From (2.1), (2.3) and using the mean-value theorem, it follows that $z(x) = u(x) - v(x)$ is the solution of the following problem

$$L_\varepsilon z(x) \equiv -\varepsilon z^{''}(x) + b z^{'}(x) + c_u z(x) = -(b - \bar{b}) v^{'}(x) - (f - \bar{f}), \quad x \in \Omega,$$

$$z(0) = 0, \quad z(1) = 0.$$

Let $z^*(x)$ be the solution of the problem

$$L_\varepsilon z^*(x) = |(b - \bar{b}) v^{'}(x)| + |f(x) - \bar{f}(x)|, \quad x \in \Omega, \quad z^*(0) = z^*(1) = 0.$$

From the maximum principle, the following inequality holds

$$|z(x)| \le z^*(x), \quad x \in \overline{\Omega}.$$

Now using Green's function $G(x,s)$ of the differential operator L_ε^b, we write down $z^*(x)$ in the form

$$\begin{aligned} z^*(x) &= -\int_0^1 G(x,s) c_u(s) z^*(s) ds + \\ &\quad \int_0^1 G(x,s) \left(|(b - \bar{b}) v^{'}(s)| + |f(s) - \bar{f}(s)| \right) ds. \end{aligned}$$

Since $G(x,s) \ge 0$, $z^*(x) \ge 0$ and $c_u(x) > 0$, it follows that

$$z^*(x) \le \int_0^1 G(x,s) \left(|(b - \bar{b}) v^{'}(s)| + |f(s) - \bar{f}(s)| \right) ds.$$

From here, (2.4), Lemma 3 applied to (2.3), and taking into account that

$$\varepsilon^{-1} \int_0^1 \exp\left(-\varepsilon^{-1} b_* (1 - x)\right) dx = b_*^{-1} (1 - \exp\left(\varepsilon^{-1} b_*\right) \le 2 b_*^{-1},$$

we prove Lemma 4. □

2.3 Construction of the difference scheme

On $\overline{\Omega}$, introduce a nonuniform mesh

$$\overline{\Omega}^h = \{0 = x_0 < x_1 < \ldots < x_{N-1} < x_N = 1,\ h_i = x_{i+1} - x_i\}, \quad p \subset \Omega^h,$$

where we assume that the points of discontinuity of the functions $b(x)$ and $f(x)$ belong to Ω^h.

On $\overline{\Omega}$, introduce the piecewise-constant functions

$$\bar{b}(x) = b(x_i + 0), \quad \bar{f}(x) = f(x_i + 0), \quad x_i \le x \le x_{i+1},$$

$$x_i \in \overline{\Omega}^h, \ i = 0, \ldots, N-1, \quad f(x_i \pm 0) = \lim_{x \to x_i \pm 0} f(x).$$

We now apply the integral-difference method from [19] to the problem

$$-\varepsilon v''(x) + \bar{b}(x) v'(x) + c(x, v) + \bar{f}(x) = 0, \quad x \in \Omega, \tag{2.5}$$

$$v(0) = 0, \quad v(1) = 0,$$

where the functions $\bar{b}$ and $\bar{f}$ are defined above and the function c from (2.1). Let G_i be Green's function of the differential operator $-\varepsilon d^2/dx^2 + \bar{b}(x_i) d/dx$ on $[x_i, x_{i+1}]$. We represent the exact solution of the problem (2.5) on $[x_i, x_{i+1}]$ in the form

$$v_i(x) = v(x_i)\, \varphi_i^I(x) + v(x_{i+1})\, \varphi_i^{II}(x) + \int_{x_i}^{x_{i+1}} G_i(x, s) \psi_i(s)\, ds,$$

$$\psi_i(x) \equiv -c(x, v) - \bar{f}(x), \quad x \in [x_i, x_{i+1}],$$

where the local Green function G_i is given by

$$G_i(x, s) = \frac{1}{-\varepsilon w_i(s)} \begin{cases} \varphi_i^I(s) \varphi_i^{II}(x), & x \le s; \\ \varphi_i^I(x) \varphi_i^{II}(s), & x \ge s, \end{cases}$$

$$w_i(s) = \varphi_i^{II}(s) \left[\varphi_i^I(x)\right]'_{x=s} - \varphi_i^I(s) \left[\varphi_i^{II}(x)\right]'_{x=s},$$

and $\varphi_i^I(x), \ \varphi_i^{II}(x)$ are defined by

$$\varphi_i^I(x) = \frac{1 - \exp\left(-b_i(x_{i+1} - x)/\varepsilon\right)}{1 - \exp\left(-b_i h_i/\varepsilon\right)}, \quad \varphi_i^{II}(x) = 1 - \varphi_i^I(x), \quad x_i \le x \le x_{i+1},$$

where $b_i = b(x_i + 0)$. Equating the derivatives $dv_{i-1}(x_i - 0)/dx$ and $dv_i(x_i + 0)/dx$ calculated on the intervals $[x_{i-1}, x_i]$ and $[x_i, x_{i+1}]$, respectively, we get the following integral-difference scheme

$$B_i(v_{i+1} - v_i) - A_i(v_i - v_{i-1}) = \Psi_i\left[\psi_{i-1}, \psi_i\right], \quad i = 1, \ldots, N-1, \tag{2.6}$$

$$v_0 = v_N = 0,$$

$$A_i = -\frac{b_{i-1}}{1 - \exp\left(-b_{i-1} h_{i-1}/\varepsilon\right)}, \quad B_i = \exp\left(-b_i h_i/\varepsilon\right) A_{i+1},$$

$$\Psi_i\left[\psi_{i-1},\psi_i\right] = -\frac{A_i}{b_{i-1}}\int_{x_{i-1}}^{x_i}\left[1-\exp\left(-\frac{b_{i-1}(s-x_{i-1})}{\varepsilon}\right)\right]\psi_{i-1}(s)ds - \frac{B_i}{b_i}\int_{x_i}^{x_{i+1}}\left[\exp\left(\frac{b_i(x_{i+1}-s)}{\varepsilon}\right)-1\right]\psi_i(s)ds.$$

Now we approximate $c(s,v)$ on $[x_{i-1},x_{i+1}]$ by the value at x_i. Doing this, we obtain the following nonlinear difference scheme

$$B_i(V_{i+1}-V_i)-A_i(V_i-V_{i-1}) = -D_i c\left(x_i,V_i\right)-\left(D_i^{(l)}f_{i-1}+D_i^{(r)}f_i\right), \qquad (2.7)$$

$$i=1,\ldots,N-1, \quad V_0=V_N=0, \quad D_i=\Psi_i[1,1], \quad f_i=f(x_i+0),$$

$$D_i=D_i^{(l)}+D_i^{(r)}, \quad D_i^{(l)}=\frac{|A_i|h_{i-1}}{b_{i-1}}-\frac{\varepsilon}{b_{i-1}}>0, \quad D_i^{(r)}=\frac{\varepsilon}{b_i}-\frac{|B_i|h_i}{b_i}>0.$$

We mention here that the coefficients of the difference scheme satisfy the inequalities $A_i<0$, $B_i<0$ and $D_i>0$.

With the above assumptions on the data of the problem (2.5), the nonlinear problem (2.7) has a unique solution [52]. Under the assumption $c^*\geq c_u\geq c_*>0$, where c_*, c^* are constants, this result will be proved in Section 2.5.

Remark 3. The linear part of the nonlinear difference operator from (2.7) corresponding to the linear differential operator $-\varepsilon d^2/dx^2+\bar{b}d/dx$ can be represented in the equivalent form

$$B_i(V_{i+1}-V_i)-A_i(V_i-V_{i-1}) = -\varepsilon\left(\frac{V_{i+1}-V_i}{h_i}-\frac{V_i-V_{i-1}}{h_{i-1}}\right)+\rho\left(\frac{b_ih_i}{\varepsilon}\right)b_i\left(V_{i+1}-V_i\right)+\rho\left(-\frac{b_{i-1}h_{i-1}}{\varepsilon}\right)b_{i-1}\left(V_i-V_{i-1}\right),$$

where

$$\rho(\zeta)=\frac{1}{\zeta}\left(1-\frac{\zeta}{\exp(\zeta)-1}\right).$$

This difference operator is equivalent to the upwind finite volume method from [44], [79], where the weighting function $\rho(\zeta)$ is denoted by $\rho_I(\zeta)$. We mention here that the difference operator satisfies the maximum principle.

2.4 Uniform convergence of the difference scheme

In the following lemma, we estimate a solution of the linear difference problem

$$B_i\left(W_{i+1}-W_i\right)-A_i\left(W_i-W_{i-1}\right)=T_i, \quad 1\leq i\leq N-1, \quad W_0=W_N=0, \qquad (2.8)$$

where A_i and B_i from (2.7).

Lemma 5. *For the linear difference problem (2.8), the following estimate holds true*

$$|W_i| \le \frac{1}{b_*} \sum_{j=1}^{N-1} |T_j|, \quad 1 \le i \le N-1, \tag{2.9}$$

where b_* defined in (2.1).

Proof. First of all, we transform the difference problem (2.8) to a self-adjoint form [61]. We multiply each equation in (2.8) by $P_i \neq 0$ and require that $A_i P_i = a_i$ and $B_i P_i = a_{i+1}$. From here, one can conclude that $A_{i+1} P_{i+1} = B_i P_i = a_{i+1}$, i.e.,

$$P_{i+1} = \frac{B_i}{A_{i+1}} P_i = P_1 \prod_{k=1}^{i} \frac{B_k}{A_{k+1}}.$$

Using the relation $B_k = \exp\left(-b_k h_k/\varepsilon\right) A_{k+1}$ from (2.6) and choosing $P_1 = 1$, we get

$$P_{i+1} = \prod_{k=1}^{i} \exp\left(-\frac{b_k h_k}{\varepsilon}\right).$$

From here and taking into account that $a_{i+1} = A_{i+1} P_{i+1}$, the difference problem (2.8) is transformed to the self-adjoint form

$$a_{i+1}(W_{i+1} - W_i) - a_i(W_i - W_{i-1}) = T_i P_i, \quad 1 \le i \le N-1, \tag{2.10}$$

$$W_0 = W_N = 0, \quad a_{i+1} = \left(-\frac{b_i}{1 - \exp(-b_i h_i/\varepsilon)}\right) \prod_{k=1}^{i} \exp\left(-\frac{b_k h_k}{\varepsilon}\right),$$

where $a_1 = A_1$. Representing the right hand side in the form

$$T_i P_i = R_i - R_{i+1}, \quad R_i = \sum_{k=i}^{N-1} T_k P_k, \quad 1 \le i \le N-1, \quad R_N = 0,$$

we obtain

$$a_{j+1}(W_{j+1} - W_j) + R_{j+1} = a_j(W_j - W_{j-1}) + R_j = K, \quad 1 \le j \le N-1,$$

where K is a constant which will be determined below. Thus,

$$W_j = W_{j-1} + \frac{K - R_j}{a_j}.$$

Summing these expressions from $j = 1$ to $j = i$, we get

$$W_i = W_0 + K \sum_{j=1}^{i} \frac{1}{a_j} - \sum_{j=1}^{i} \frac{R_j}{a_j}.$$

From here and taking into account that $W_0 = W_N = 0$, one can conclude that

$$K = \left(\sum_{j=1}^{N} \frac{R_j}{a_j} \right) \left(\sum_{j=1}^{N} \frac{1}{a_j} \right)^{-1}.$$

We can represent W_i in the form

$$W_i = -(1-\theta_i) \sum_{j=1}^{i} \frac{R_j}{a_j} + \theta_i \sum_{j=i+1}^{N} \frac{R_j}{a_j}, \quad 1 \le i \le N-1,$$

where

$$\theta_i = \left(\sum_{j=1}^{i} \frac{1}{a_j} \right) \left(\sum_{j=1}^{N} \frac{1}{a_j} \right)^{-1}, \quad 0 < \theta_i < 1.$$

Thus,

$$|W_i| \le (1-\theta_i) \sum_{j=1}^{N} \frac{|R_j|}{|a_j|} + \theta_i \sum_{j=i+1}^{N} \frac{|R_j|}{|a_j|} \le \sum_{j=1}^{N-1} \frac{|R_j|}{|a_j|},$$

where we take into account that $R_N = 0$. Now we estimate

$$Q = \sum_{j=1}^{N-1} \frac{|R_j|}{|a_j|} = \sum_{j=1}^{N-1} \frac{1}{|a_j|} \left(\sum_{k=j}^{N-1} |T_k| \, P_k \right).$$

Changing the order of summation, write down Q as

$$Q = \sum_{j=1}^{N-1} |T_j| \, P_j \left(\sum_{k=1}^{j} \frac{1}{|a_k|} \right).$$

Representing $|a_{i+1}|$ in the form

$$|a_{i+1}| = \frac{b_i}{P_{i+1}^{-1} - P_i^{-1}}, \quad 1 \le i \le N-1, \quad |a_1| = -A_1,$$

and using the assumption on b from (2.1), we estimate

$$\sum_{k=1}^{j} \frac{1}{|a_k|} = \frac{1}{|a_1|} + \sum_{k=2}^{j-1} \frac{P_{k+1}^{-1} - P_k^{-1}}{b_k} \le \frac{1}{b_* P_j}.$$

Thus,

$$Q \le \frac{1}{b_*} \sum_{j=1}^{N-1} |T_j|,$$

and we prove the lemma. □

Theorem 2. *The nonlinear difference scheme (2.7) on arbitrary meshes converges ε-uniformly to the solution of problem (2.1) :*

$$\max_{0 \le i \le N} |u(x_i) - V_i| \le Ch, \quad h = \max_{0 \le i \le N-1} h_i.$$

Proof. From Lemma 4 and the construction of the functions $\bar{b}$ and $\bar{f}$ in (2.5), we conclude the estimate

$$\max_{x\in\overline{\Omega}} |u(x) - v(x)| \le Ch, \tag{2.11}$$

where u and v are the solutions to (2.1) and (2.5), respectively.

We now estimate the error $\{Z_i = v(x_i) - V_i,\ 0 \le i \le N\}$ in the approximation of the continuous solution of the problem (2.5) by the nonlinear difference scheme (2.7). From (2.5) and (2.7) by the mean-value theorem, we conclude that the error function $\{Z_i,\ 0 \le i \le N\}$ solves the following difference problem

$$B_i\,(Z_{i+1} - Z_i) - A_i\,(Z_i - Z_{i-1}) + D_i c_u Z_i = \Psi_i[\delta_{i-1}, \delta_i], \quad 1 \le i \le N-1, \tag{2.12}$$

$$Z_0 = Z_N = 0,$$

where

$$\delta_{i-1}(s) = \int_s^{x_i} \frac{dc}{dx} dx, \quad s \in [x_{i-1}, x_i], \quad \delta_i(s) = -\int_{x_i}^{s} \frac{dc}{dx} dx, \quad s \in [x_i, x_{i+1}],$$

and the functional Ψ_i is defined in (2.6).

Let $w_i(s) \ge 0,\ s \in [x_i, x_{i+1}],\ 0 \le i \le N-1$. From (2.6), we can represent $\Psi_i[w_{i-1}, w_i]$ in the form

$$\begin{aligned}\Psi_i \;=\; & \frac{1}{1 - \exp\left(-b_{i-1}h_{i-1}/\varepsilon\right)} \int_{x_{i-1}}^{x_i} \left[1 - \exp\left(-b_{i-1}(s - x_{i-1})/\varepsilon\right)\right] w_{i-1}(s)ds + \\ & \frac{1}{\exp\left(b_i h_i/\varepsilon\right) - 1} \int_{x_i}^{x_{i+1}} \left[\exp\left(b_i(x_{i+1} - s)/\varepsilon\right) - 1\right] w_i(s)ds.\end{aligned}$$

Taking into account the inequalities

$$1 - \exp\left(-b_{i-1}(s - x_{i-1})/\varepsilon\right) \le 1 - \exp\left(-b_{i-1}h_{i-1}/\varepsilon\right), \quad s \in [x_{i-1}, x_i],$$

$$\exp\left(b_i(x_{i+1} - s)/\varepsilon\right) - 1 \le \exp\left(b_i h_i/\varepsilon\right) - 1, \quad s \in [x_i, x_{i+1}],$$

we prove that

$$0 \le \Psi_i[w_{i-1}, w_i] \le \int_{x_{i-1}}^{x_i} w_{i-1}(s)ds + \int_{x_i}^{x_{i+1}} w_i(s)ds, \quad 1 \le i \le N-1.$$

From here, Lemma 3 applied to (2.5), we obtain the estimate of the right-hand side from (2.12) in the form

$$|\Psi_i[\delta_{i-1}, \delta_{i+1}]| \le T_i, \quad 1 \le i \le N-1, \tag{2.13}$$

$$T_i = C \int_{x_{i-1}}^{x_{i+1}} h \left[1 + \frac{1}{\varepsilon} \exp\left(-\frac{b_*(1-s)}{\varepsilon} \right) \right] ds.$$

Using the maximum principle for the difference operator in (2.8) and Lemma 5, the solution of problem (2.8) with the above right-hand side is estimated as

$$0 \le W_i \le \frac{1}{b_*} \sum_{j=1}^{N-1} T_k \le C \sum_{j=1}^{N-1} \int_{x_{i-1}}^{x_{i+1}} h \left[1 + \frac{1}{\varepsilon} \exp\left(-\frac{b_*(1-s)}{\varepsilon} \right) \right] ds.$$

Thus,

$$0 \le W_i \le Ch, \quad 1 \le i \le N-1.$$

We now show that

$$|Z_i| \le W_i \le Ch, \quad 0 \le i \le N, \tag{2.14}$$

where $\{Z_i,\ 0 \le i \le N\}$ is the solution to (2.12) and $\{W_i,\ 0 \le i \le N\}$ is the solution to (2.8) with the right-hand side from (2.13). $\{W_i,\ 0 \le i \le N\}$ satisfies the difference problem

$$B_i \left(W_{i+1} - W_i \right) - A_i \left(W_i - W_{i-1} \right) + D_i c_u W_i = T_i + D_i c_u W_i, \quad 1 \le i \le N-1,$$

$$W_0 = W_N = 0,$$

where c_u from (2.12). Taking into account that $D_i > 0$, $c_u \ge 0$, $W_i \ge 0$ and $|\Psi_i[\delta_{i-1}, \delta_i]| \le T_i,\ 1 \le i \le N-1$, by the maximum principle for the difference operator in (2.12) we conclude (2.14). The theorem now follows from (2.11) and (2.14). □

2.5 Monotone iterative method

In this section, we construct an iterative method for solving the nonlinear difference scheme (2.7) which possesses monotone convergence. This method is based on the monotone approach from [10].

Additionally, we assume that $c(x,u)$ from (2.1) satisfies the two-sided constraint

$$0 < c_* \le c_u \le c^*, \quad c_*, c^* = \text{const}. \tag{2.15}$$

We mention that the assumption $c_u \ge c_*$ can always be achieved by a transformation $u = \tilde{u} \exp(\gamma x)$, with γ chosen appropriately.

Introduce the linear version of (2.7)

$$\left(\mathcal{L} + c_i \right) W_i = -F_i, \quad 1 \le i \le N-1, \quad W_0 = w_0, \quad W_N = w_1, \tag{2.16}$$

$$c_i \geq c_0 = \text{const} > 0,\ 1 \leq i \leq N, \quad F_i = \left(D_i^{(l)} f_{i-1} + D_i^{(r)} f_i\right)/D_i,$$

$$\mathcal{L}W_i \equiv \left[B_i(W_{i+1} - W_i) - A_i(W_i - W_{i-1})\right]/D_i.$$

Now we formulate the maximum principle for the difference operator $\mathcal{L} + c$ and give an estimate of the solution to (2.16). We use the notation $V = (V_0, \ldots, V_N)^T$.

Lemma 6. *(i) If W satisfies the conditions*

$$(\mathcal{L} + c_i)\, W_i \geq 0\, (\leq 0),\ 1 \leq i \leq N-1, \quad W_0, W_N \geq 0\, (\leq 0),$$

then $W_i \geq 0$ (≤ 0), $0 \leq i \leq N$.

(ii) The following estimate of the solution to (2.16) holds true

$$\|W\|_{\overline{\Omega}^h} \leq \max\left[|W_0|, |W_N|, \|f_i\|_{\Omega^h}/c_0\right], \tag{2.17}$$

where

$$\|W\|_{\overline{\Omega}^h} \equiv \max_{0 \leq i \leq N} |W_i|, \quad \|f\|_{\Omega^h} \equiv \max_{1 \leq i \leq N-1} |f_i|.$$

Proof. Taking into account that $D_i = D_i^{(l)} + D_i^{(r)}$, we conclude

$$\|F\|_{\Omega^h} \leq \|f\|_{\Omega^h}. \tag{2.18}$$

Now, the proof of the lemma can be found in Lemma 1. □

The iterative method is constructed in the following way. Choose an initial mesh function $V^{(0)}$ satisfying the boundary conditions $V_0^{(0)} = V_N^{(0)} = 0$. The iterative sequence $\{V^{(n)}\}$, $n \geq 1$, is defined by the recurrence formulae

$$(\mathcal{L} + c^*)\, Z_i^{(n)} = -\mathcal{R}_i\left(V^{(n-1)}\right), \quad 1 \leq i \leq N-1, \tag{2.19}$$

$$Z_0^{(n)} = Z_N^{(n)} = 0,$$

$$V_i^{(n)} = V_i^{(n-1)} + Z_i^{(n)}, 0 \leq i \leq N.$$

$$\mathcal{R}_i\left(V^{(n-1)}\right) \equiv \mathcal{L}V_i^{(n-1)} + c\left(V_i^{(n-1)}\right) + F_i,$$

where $\mathcal{L}$ and F are defined in (2.16) and $\mathcal{R}_i\left(V^{(n-1)}\right)$ is the residual of the difference scheme (2.7) on $V^{(n-1)}$.

We say that $\overline{V}$ is an upper solution of (2.7) if it satisfies the inequalities

$$\mathcal{L}\overline{V}_i + c_i(\overline{V}_i) + F_i \geq 0,\ 1 \leq i \leq N-1, \quad \overline{V}_0, \overline{V}_N \geq 0.$$

Similarly, $\underline{V}$ is called a lower solution if it satisfies the reversed inequalities. Upper and lower solutions satisfy the inequality

$$\underline{V}_i \le \overline{V}_i, \quad 0 \le i \le N.$$

This inequality is proved in Lemma 2.

We have the following theorem which gives the monotone property of the iterative method (2.19).

Theorem 3. *Let $\overline{V}^{(0)}, \underline{V}^{(0)}$ be upper and lower solutions of (2.7), and let $c(x,u)$ satisfy (2.15). Then the upper sequence $\{\overline{V}^{(n)}\}$ generated by (2.19) converges monotonically from above to the unique solution V of (2.7), the lower sequence $\{\underline{V}^{(n)}\}$ generated by (2.19) converges monotonically from below to V:*

$$\underline{V}_i^{(0)} \le \underline{V}_i^{(n)} \le \underline{V}_i^{(n+1)} \le V_i \le \overline{V}_i^{(n+1)} \le \overline{V}_i^{(n)} \le \overline{V}_i^{(0)}, \quad 0 \le i \le N,$$

and the sequences converge at the linear rate $q = 1 - c_*/c^*$.

Proof. We consider only the case of the upper sequence. If $\overline{V}^{(0)}$ is an upper solution, then from (2.19) we conclude that

$$(\mathcal{L} + c^*)\, Z_i^{(1)} \le 0, \quad 1 \le i \le N-1, \quad Z_0^{(1)} = Z_N^{(1)} = 0.$$

From Lemma 6, by the maximum principle for the difference operator $\mathcal{L} + c^*$, it follows that $Z_i^{(1)} \le 0$, $0 \le i \le N$. Using the mean-value theorem and the equation for $Z_i^{(1)}$, we represent $\mathcal{R}_i\left(V^{(1)}\right)$ in the form

$$\mathcal{R}_i\left(V^{(1)}\right) = -(c^* - c_{u,i}^{(1)}) Z_i^{(1)}, \quad 1 \le i \le N-1, \tag{2.20}$$

where $c_{u,i}^{(1)} \equiv c_u[x_i, \overline{V}_i^{(0)} + \vartheta_i^{(1)} Z_i^{(1)}], 0 < \vartheta_i^{(1)} < 1$. Since the mesh function $Z^{(1)}$ is non-positive on Ω^h and taking into account (2.15), we conclude that $\overline{V}^{(1)}$ is an upper solution. By induction we obtain that $Z_i^{(n)} \le 0$, $0 \le i \le N$, $n = 1, 2, \ldots$, and prove that $\{\overline{V}^{(n)}\}$ is a monotonically decreasing sequence of upper solutions.

We now prove that the monotone sequence $\{\overline{V}^{(n)}\}$ converges to the solution of (2.7). Similar to (2.20), we obtain

$$\mathcal{R}_i(\overline{V}^{(n)}) = -(c^* - c_{u,i}^{(n)}) Z_i^{(n)}, \quad 1 \le i \le N-1,$$

and from (2.19), it follows that $Z_i^{(n+1)}$ satisfies the difference equation

$$(\mathcal{L} + c^*)\, Z_i^{(n+1)} = (c^* - c_{u,i}^{(n)}) Z_i^{(n)}, \quad 1 \le i \le N-1.$$

Using (2.15) and (2.17), we have

$$\|Z^{(n+1)}\|_{\overline{\Omega}^h} \le q \|Z^{(n)}\|_{\overline{\Omega}^h}.$$

From here, and by induction on n it follows that

$$\|Z^{(n+1)}\|_{\overline{\Omega}^h} \le q^n \|Z^{(1)}\|_{\overline{\Omega}^h}. \tag{2.21}$$

This proves convergence of the upper sequence at the linear rate q. Now by linearity of the operator $\mathcal{L}$ and the continuity of c, we have also from (2.19) that the mesh function V defined by

$$V = \lim_{n\to\infty} \overline{V}^{(n)}, \quad 0 \le i \le N,$$

is an exact solution to (2.7). The uniqueness of the solution to (2.7) follows from estimate (2.17). Indeed, if by contradiction, we assume that there exist two solutions V_1 and V_2 to (2.7), then by the mean-value theorem, the difference $\delta V = V_1 - V_2$ satisfies the difference problem

$$\mathcal{L}\delta V_i + c_{u,i}\delta V_i = 0, \quad 1 \le i \le N-1, \quad \delta V_0 = \delta V_N = 0.$$

By (2.17), $\delta V = 0$ which leads to the uniqueness of the solution to (2.7). This proves the theorem. □

An initial upper or lower solution for the monotone iterative method may be constructed using the method given in Remark 2.

Remark 4. Since the initial iteration in method (2.19) is either an upper or lower solution, which can be constructed directly from the difference equation without any knowledge of the solution, this algorithm eliminates the search for the initial iteration as is often needed in Newton's method. This gives a practical advantage in the computation of numerical solutions.

Let the initial function $V_i^{(0)}$, $0 \le i \le N$ be chosen in the form of (1.22) with $t(x) = 0$, that is, $V^{(0)}$ is the solution of the difference problem

$$(\mathcal{L} + c_*)\, V_i^{(0)} = \nu \left| c(x_i, 0) + F_i \right|, \quad 1 \le i \le N-1, \tag{2.22}$$

$$V_0^{(0)} = V_N^{(0)} = 0, \quad \nu = 1, -1.$$

Then the functions $\overline{V}_i^{(0)}$, $\underline{V}_i^{(0)}$ $0 \le i \le N$, corresponding to $\nu = 1$ and $\nu = -1$ are upper and lower solutions, respectively.

Theorem 4. *If the initial upper or lower solution $V^{(0)}$ is chosen in the form of (2.22), then the monotone iterative method (2.19) converges ε-uniformly to the solution of problem (2.1):*

$$\left\|V^{(n)} - u\right\|_{\overline{\Omega}^h} \le Ch + \frac{c_0\,(q)^n}{(1-q)} \left(\|c(x_i,0)\|_{\overline{\Omega}^h} + \|f_i\|_{\overline{\Omega}^h}\right),$$

$$q = 1 - \frac{c_*}{c^*} < 1, \quad c_0 = \frac{3c_* + c^*}{c_* c^*},$$

where constant C is independent of ε and h.

Proof. Using (2.21), we have

$$\begin{aligned}\left\|V^{(n+k)} - V^{(n)}\right\|_{\overline{\Omega}^h} &\le \sum_{j=n}^{n+k-1} \left\|V^{(j+1)} - V^{(j)}\right\|_{\overline{\Omega}^h} = \sum_{j=n}^{n+k-1} \left\|Z^{(j+1)}\right\|_{\overline{\Omega}^h} \\ &\le \frac{q}{1-q} \left\|Z^{(n)}\right\|_{\overline{\Omega}^h} \le \frac{(q)^n}{1-q} \left\|Z^{(1)}\right\|_{\overline{\Omega}^h}.\end{aligned}$$

Taking into account that $\lim V^{(n+k)} = V$ as $k \to \infty$, where V is the solution to (2.7), we conclude the estimate

$$\left\|V^{(n)} - V\right\|_{\overline{\Omega}^h} \le \frac{(q)^n}{1-q} \left\|Z^{(1)}\right\|_{\overline{\Omega}^h}. \tag{2.23}$$

From (2.18), (2.19), (2.22), the definition of $\mathcal{R}_i$ in (2.19) and the mean-value theorem

$$\begin{aligned}\left\|Z^{(1)}\right\|_{\overline{\Omega}^h} &\le \frac{1}{c^*} \left\|\mathcal{L}V^{(0)}\right\|_{\Omega^h} + \frac{1}{c^*} \left\|c(x_i, V^{(0)})\right\|_{\overline{\Omega}^h} + \frac{1}{c^*} \|f\|_{\overline{\Omega}^h} \\ &\le \frac{1}{c^*} \left(c_* \left\|V^{(0)}\right\|_{\overline{\Omega}^h} + \|c(x_i,0)\|_{\overline{\Omega}^h} + \|f\|_{\overline{\Omega}^h}\right) + \\ &\quad \frac{1}{c^*} \|c(x_i,0)\|_{\overline{\Omega}^h} + \left\|V^{(0)}\right\|_{\overline{\Omega}^h} + \frac{1}{c^*} \|f\|_{\overline{\Omega}^h}.\end{aligned}$$

From here and estimating $V^{(0)}$ from (2.22) by (2.17) and (2.18),

$$\left\|V^{(0)}\right\|_{\overline{\Omega}^h} \le \frac{1}{c_*} \|c(x_i,0)\|_{\overline{\Omega}^h} + \frac{1}{c_*} \|f\|_{\overline{\Omega}^h},$$

we conclude the estimate on $Z^{(1)}$ in the form

$$\left\|Z^{(1)}\right\|_{\overline{\Omega}^h} \le c_0 \left(\|c(x_i,0)\|_{\overline{\Omega}^h} + \|f\|_{\overline{\Omega}^h}\right),$$

where c_0 is defined in the theorem. Thus, from here, (2.23) and Theorem 2, we prove the theorem. □

2.6 Numerical experiments

We solve the nonlinear difference scheme (2.7) on uniform meshes by the monotone iterative method (2.19). The stopping criterion is

$$\max_{0\le i\le N} |V_i^{(n)} - V_i^{(n-1)}| \le \sigma,$$

where σ is the required accuracy. If at step $n = n_*$ we satisfy the stopping criterion, then $V = V^{(n_*)}$, where V is the corresponding numerical solution.

In the absence of an exact solution for test problems, for fixed value of ε, the nonlinear difference scheme (2.7) with $N = 8192$ is solved by the monotone iterative method (2.19) with the stopping criterion $\sigma = 10^{-5}$. This generates a reference solution V_{ref}.

The basic feature of monotone convergence of the upper and lower sequences is observed in all the numerical experiments. In fact, the monotone property of the sequences holds at every mesh point in the domain. Of course, this is expected from the analytical considerations.

Test problem 1. Consider the following test problem:

$$-\varepsilon u'' + b(x)u' + c(x,u) + f(x) = 0, \quad u(0) = 1, \quad u(1) = 1,$$

$$c(x,u) = 1 - \exp(-u),$$

$$b(x) = 1, \quad f(x) = \begin{cases} 1, & x \le 0.5, \\ -0.5, & x > 0.5. \end{cases}$$

It is easily seen that

$$\overline{V}_i^{(0)} = 1, \quad \underline{V}_i^{(0)} = 0, \quad 0 \le i \le N, \tag{2.24}$$

are upper and lower solutions to (2.7). From Theorem 3, we conclude that

$$c_* = \min_{0\le u\le 1} c_u = e^{-1}, \quad c^* = \max_{0\le u\le 1} c_u = 1, \tag{2.25}$$

where c_* and c^* are defined in (2.15). In our numerical experiments, the upper solution $\overline{V}_i^{(0)} = 1$, $0 \le i \le N$ is used as an initial iteration.

In Table 2.1 for various values of ε and N, we present the maximal approximate error

$$\overline{E}_{N,\varepsilon} = \max_{0\le i\le N} E_{N,\varepsilon}(x_i), \quad E_{N,\varepsilon} \equiv |V_{N,\varepsilon} - V_{ref,\varepsilon}|,$$

where $V_{N,\varepsilon}$ is the numerical solution of the nonlinear difference scheme (2.7) by the monotone iterative method (2.19). For $\varepsilon \le 10^{-4}$, the error is independent of ε and decreases

$N\backslash\varepsilon$	0.1	0.01	0.001	≤ 0.0001
32	2.52×10^{-3}	5.04×10^{-3}	5.32×10^{-3}	5.33×10^{-3}
64	1.23×10^{-3}	2.49×10^{-3}	2.64×10^{-3}	2.65×10^{-3}
128	6.07×10^{-4}	1.23×10^{-3}	1.31×10^{-3}	1.31×10^{-3}
256	2.97×10^{-4}	6.03×10^{-4}	6.42×10^{-4}	6.45×10^{-4}
512	1.44×10^{-4}	2.91×10^{-4}	3.10×10^{-4}	3.12×10^{-4}
1024	6.70×10^{-5}	1.36×10^{-4}	1.45×10^{-4}	1.46×10^{-4}
2048	2.87×10^{-5}	5.82×10^{-5}	6.20×10^{-5}	6.24×10^{-5}

Table 2.1: Maximal approximate error $\overline{E}_{N,\varepsilon}$ for the monotone iterative method (2.19) applied to the test problem 1.

with N. This table verifies our convergent results from Theorems 2 and 4, that is the nonlinear difference scheme by the monotone iterative method converges ε-uniformly.

The numerical order of convergence $\overline{\alpha}_{N,\varepsilon}$ and the uniform numerical order of convergence $\overline{\alpha}^*_N$ are calculated as in [32],

$$\overline{R}_{N,\varepsilon} = \max_{0\leq i\leq N} |V_{N,\varepsilon} - V_{2N,\varepsilon}|, \quad \overline{R}^*_N = \max_{\varepsilon} \overline{R}_{N,\varepsilon},$$

$$\overline{\alpha}_{N,\varepsilon} = \log_2 \left(\frac{\overline{R}_{N,\varepsilon}}{\overline{R}_{2N,\varepsilon}} \right), \quad \overline{\alpha}^*_N = \log_2 \left(\frac{\overline{R}^*_N}{\overline{R}^*_{2N}} \right),$$

and are close to one. This confirms the theoretical result from Theorem 2.

The iteration counts are presented in Table 2.2. For $\varepsilon \leq 10^{-3}$, the convergence iteration count is 6. This result confirms the theoretical result from Theorems 3 and 4, that the convergence factor q of the monotone iterative method (2.19) is independent of ε.

The approximate error $E_{N,\varepsilon}$ with $N = 128$ and $\varepsilon = 10^{-2}, 10^{-3}$ is depicted in Figure 2.1. The maximum of the approximate error ia attained in the vicinity of the point of discontinuity $x = 0.5$ of $f(x)$.

Test problem 2. The second test problem is defined by

$$-\varepsilon u'' + b(x)u' + c(x,u) + f(x) = 0, \quad u(0) = 1, \quad u(1) = 1,$$

$$c(x,u) = 1 - \exp(-u),$$

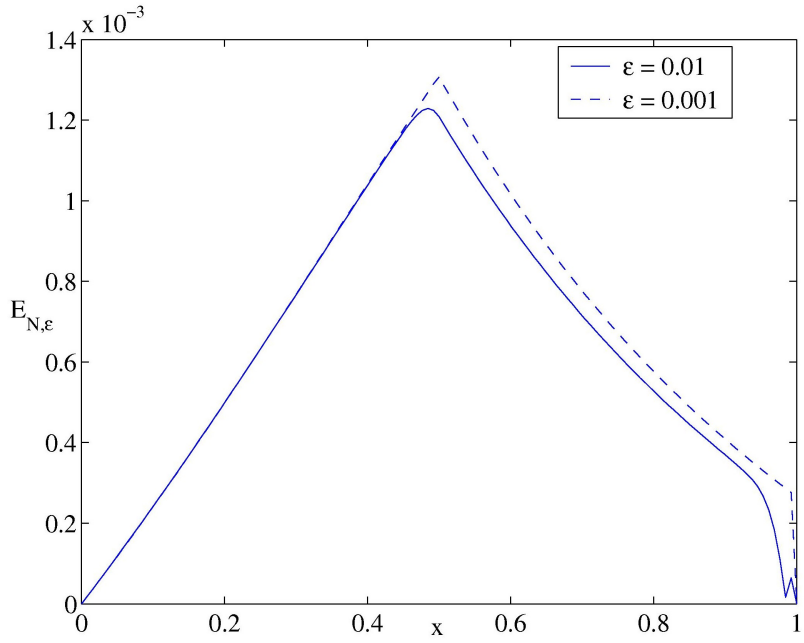

Figure 2.1: $E_{N,\varepsilon}(x)$ with N=128 and $\varepsilon = 10^{-2}, 10^{-3}$ for the test problem 1.

$N\backslash\varepsilon$	1	0.1	0.01	≤ 0.001
32	5	7	6	6
64	5	7	6	6
128	5	7	6	6
256	5	7	6	6
≥ 512	5	7	7	6

Table 2.2: Iteration counts for the monotone iterative method (2.19) applied to the test problem 1.

$$b(x) = \begin{cases} 2, & x \leq 0.5, \\ 1, & x > 0.5, \end{cases}, \quad f(x) = -0.5.$$

Similar to the test problem 1, the functions from (2.24) are upper and lower solutions to (2.7) and c_*, c^* are defined by (2.25). The upper solution $\overline{V}_i^{(0)} = 1$, $0 \leq i \leq N$ is used as an initial iteration.

In Table 2.3, the maximal approximate error is presented for various values of ε and N. For $\varepsilon \leq 10^{-4}$, the error is independent of ε and decreases with N. This table verifies our convergent results from Theorems 2 and 4.

$N\backslash\varepsilon$	0.1	0.01	0.001	≤ 0.0001
32	3.97×10^{-4}	8.25×10^{-4}	8.69×10^{-4}	8.69×10^{-4}
64	1.95×10^{-4}	4.08×10^{-4}	4.40×10^{-4}	4.40×10^{-4}
128	9.64×10^{-5}	2.02×10^{-4}	2.19×10^{-4}	2.20×10^{-4}
256	4.76×10^{-5}	9.92×10^{-5}	1.08×10^{-4}	1.09×10^{-4}
512	2.33×10^{-5}	4.80×10^{-5}	5.22×10^{-5}	5.27×10^{-5}
1024	1.12×10^{-5}	2.24×10^{-5}	2.44×10^{-5}	2.46×10^{-5}
2048	5.23×10^{-6}	9.63×10^{-6}	1.05×10^{-5}	1.06×10^{-5}

Table 2.3: Maximal approximate error $\overline{E}_{N,\varepsilon}$ for the monotone iterative method (2.19) applied to the test problem 2.

The numerical order of convergence $\overline{\alpha}_{N,\varepsilon}$ and the uniform numerical order of convergence $\overline{\alpha}^*_N$ are close to one, this confirms the theoretical result from Theorem 2.

The iteration counts are the same as for the test problem 1 presented in Table 2.2. For $\varepsilon \leq 10^{-3}$, the convergence iteration count is 6. This result confirms the theoretical result from Theorems 3 and 4 that the convergence factor q of the monotone iterative method (2.19) is independent of ε.

The approximate error $E_{N,\varepsilon}$ with $N = 128$ and $\varepsilon = 10^{-2}, 10^{-3}$ is depicted in Figure 2.2. The maximum of the approximate error is attained in the boundary layer at $x = 1$.

2.6.1 Numerical observations

The numerical experiments confirm the theoretical result in Theorem 2 on uniform convergence on the nonlinear difference scheme (2.7). The numerical experiments also confirm the theoretical results in Theorems 3 and 4, that the iterative method (2.19) converges monotonically and ε-uniformly.

2.7 Conclusions

In this chapter, we construct the nonlinear difference scheme (2.7) for solving the semilinear two-point boundary-value problem with the convective dominated term and discontinuous data (2.1). The construction of the nonlinear difference scheme is based on locally exact schemes or on local Green's functions. In Lemma 3, *a priori* estimates of the exact

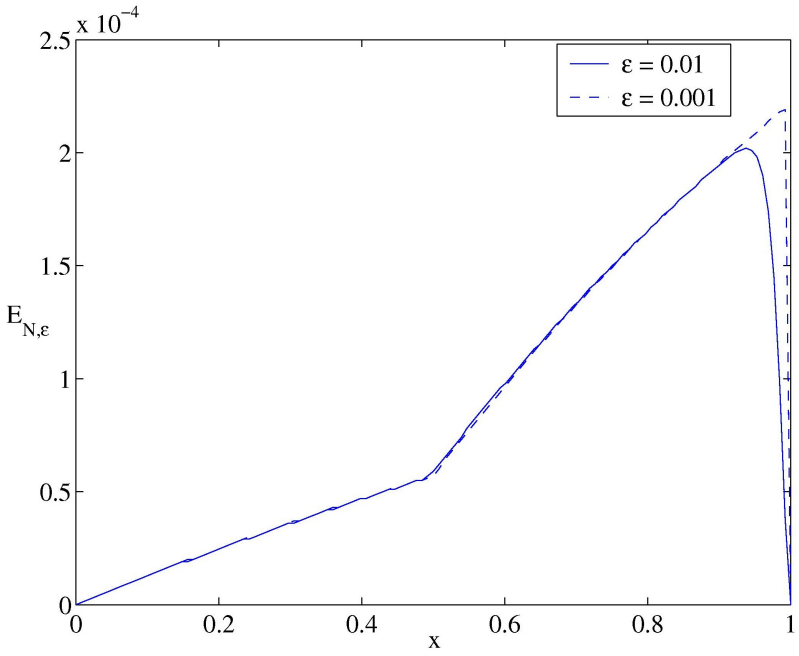

Figure 2.2: $E_{N,\varepsilon}$ with N=128 and $\varepsilon = 10^{-2}, 10^{-3}$ for the test problem 2.

solution are proved. In Theorem 2, we prove that the nonlinear difference scheme (2.7) on arbitrary meshes converges ε-uniformly to the solution of the continuous problem (2.1) by extending in a natural way the proof of the main theoretical results from [13]. The monotone iterative method (2.19) based on the upper and lower solutions is constructed to compute the nonlinear difference scheme (2.7). In Theorem 3, we prove that the upper and lower solutions generated by (2.19) converge monotonically to the exact solution of (2.7) with a linear rate. By choosing the special initial functions (2.22), in Theorem 4 we prove ε-uniform convergence of the iterative sequences to the exact solution of the continuous problem. The numerical experiments are implemented for the test problems and confirm the theoretical results.

Chapter 3

Monotone domain decomposition algorithms

This chapter describes iterative domain decomposition algorithms for solving two dimensional nonlinear singularly perturbed convection-diffusion problems with the convective dominated term. We consider one-level domain decomposition algorithm which consists of outer iterates and two-level domain decomposition algorithm which consists of outer and inner iterates. One outer iterative step represents computing nonlinear difference subproblems on overlapping subdomains in serial according to upwind error propagation (the multiplicative Schwarz method). The one-level domain decomposition algorithm solves linear subproblems, whereas the two-level domain decomposition algorithm solves nonlinear subproblems. At the level of the inner iterations, each nonlinear subproblem is solved by the monotone additive Schwarz algorithm. The advantage of the algorithm is that the algorithm solves only linear discrete systems at each iterative step. We prove that the algorithms converge monotonically to the exact solution of the system. The advantage of the two-level algorithm is that it has a potential for parallel computing. Results of numerical experiments are presented.

3.1 Introduction

In this chapter, we consider the one- and two-level Schwarz domain decomposition algorithms from [37]. The one-level domain decomposition algorithm consists of outer iterations, whereas the two-level domain decomposition algorithm consists of the two iterative

processes: outer and inner iterations. One outer iterative step represents computing M subproblems on overlapping vertical subdomains (strips) $\overline{\Omega}_m$, $m = 1, \ldots, M$, serially starting from subdomain $\overline{\Omega}_1$ and finishing off on $\overline{\Omega}_M$ (according to upwind error propagation). Thus, the multiplicative Schwarz method is the outer part of the algorithm. At the level of the inner iterations, each vertical strip $\overline{\Omega}_m$ is split into nonoverlapping boxes (horizontal strips) with interface γ. Small interfacial subdomains are introduced near the interface γ, and approximate boundary values computed on γ are used for solving problems on the nonoverlapping box-subdomains. Thus, the additive Schwarz method is the inner part of the algorithm.

The proposed algorithm combines the Schwarz domain decomposition algorithm with the method of upper and lower solutions. The method of upper and lower solutions is a monotone iterative method which also provides a method of constructing initial solutions without prior knowledge of the actual solution, as is often required in Newton's method. The monotonicity condition guarantees that systems of algebraic equations based on such methods are well posed. The advantages of the algorithm are that the algorithm solves only linear discrete systems at each iterative step, converges monotonically to the exact solution of the system and is parallelisable.

In Section 3.2, we investigate the one-level monotone domain decomposition algorithm for solving the nonlinear difference scheme (1.15). Monotone convergence of upper and lower solutions generated by this algorithm is proved. Numerical experiments are implemented for the convection-diffusion problem with parabolic layers and the anisotropic convection-diffusion problem. Section 3.3 deals with the two-level monotone domain decomposition algorithm for solving (1.15). We prove monotone convergence of the inner iterates. Numerical experiments are implemented for the convection-diffusion problem with parabolic layers and the anisotropic convection-diffusion problem.

Some of the content of this chapter is published in [22] and [23].

3.2 Monotone domain decomposition algorithm

In this section, we investigate the one-level monotone domain decomposition algorithms for solving the nonlinear difference scheme (1.15) based on the multiplicative Schwarz algorithm. We assume that $f(p, v)$ in (1.15) satisfies the two-sided constraint (1.20), and

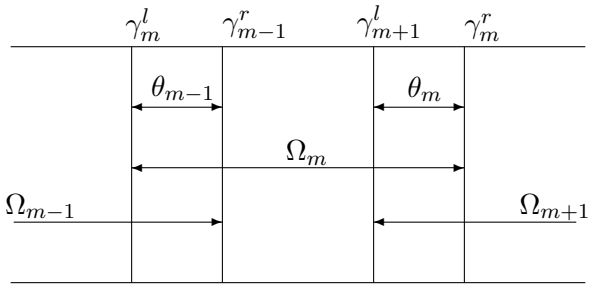

Figure 3.1: Fragment of the domain decomposition with overlapping subdomains Ω_{m-1}, Ω_m, Ω_{m+1} and overlaps θ_{m-1}, θ_m.

the computational domain Ω is two dimensional rectangular domain. The rectangular mesh $\overline{\Omega}^h = \overline{\Omega}^{hx} \times \overline{\Omega}^{hy}$ given in (1.3) is used.

We introduce the set of the overlapping vertical strips $\overline{\Omega}_m$, $m = 1, \ldots, M$, with the boundaries

$$\partial\Omega_m = \gamma_m^l \cup \gamma_m^r \cup \gamma_m^0,$$

where γ_m^l and γ_m^r are the left and right boundaries of $\overline{\Omega}_m$, respectively, and γ_m^0 belongs to the boundary of $\overline{\Omega}$. A fragment of the domain decomposition is illustrated on Figure 3.1. Thus,

$$\overline{\Omega}_m \cap \overline{\Omega}_{m+1} = \overline{\theta}_m, \quad m = 1, \ldots, M-1,$$

where $\overline{\theta}_m$ is the overlap between two subdomains $\overline{\Omega}_m$ and $\overline{\Omega}_{m+1}$. We assume that the left $\gamma_m^{hl} = \gamma_m^l \cap \overline{\Omega}^h$ and right $\gamma_m^{hr} = \gamma_m^r \cap \overline{\Omega}^h$ boundaries of $\overline{\Omega}_m^h = \overline{\Omega}_m \cap \overline{\Omega}^h$, $m = 1, \ldots, M$, are mesh lines.

One complete iterative step includes solving a sequence of M problems on subdomains $\overline{\Omega}_m^h$, $m = 1, \ldots, M$, in serial. For computing the problem on subdomain $\overline{\Omega}_m^h$, $m > 1$, the Dirichlet boundary condition on the left boundary is updated by using the solution of the problem on subdomain $\overline{\Omega}_{m-1}^h$ (previous substep of the outer iterative step). The solution on the right boundary is set equal to the solution found from the previous iteration $v^{(n-1)}(p)$, and the solutions on the top and bottom boundaries are equal to the original boundary condition $g(p)$.

The domain decomposition algorithm is as follows:

1. Initialisation: On the whole mesh $\overline{\Omega}^h$, choose an initial function $v^{(0)}(p)$, $p \in \overline{\Omega}^h$, satisfying the boundary condition $v^{(0)}(p) = g(p)$ on $\partial\Omega^h$.

2. On subdomains $\overline{\Omega}_m^h$, $m = 1, \ldots, M$, compute in serial mesh functions $v_m^{(n)}(p)$, $m = 1, \ldots, M$, satisfying the difference schemes

$$(\mathcal{L} + c^*) z_m^{(n)}(p) = -\mathcal{R}(p, v^{(n-1)}), \quad p \in \Omega_m^h, \tag{3.1}$$

$$z_m^{(n)}(p) = v_m^{(n)}(p) - v^{(n-1)}(p),$$

$$\mathcal{R}(p, v^{(n-1)}) = \mathcal{L} v^{(n-1)}(p) + f(p, v^{(n-1)}),$$

with the boundary conditions

$$z_m^{(n)}(p) = \begin{cases} z_{m-1}^{(n)}(p), & p \in \gamma_m^{hl}, \\ 0, & p \in \gamma_m^{hr} \cup \gamma_m^{h0}, \end{cases}$$

where

$$\gamma_m^{hl} = \gamma_m^l \cap \overline{\Omega}_m^h, \quad \gamma_m^{hr} = \gamma_m^r \cap \overline{\Omega}_m^h, \quad \gamma_m^{h0} = \gamma_m^0 \cap \overline{\Omega}_m^h.$$

3. Compute the solution $v^{(n)}(p)$, $p \in \overline{\Omega}^h$ by piecing together the solutions on the subdomains

$$v^{(n)}(p) = \begin{cases} v_m^{(n)}(p), & p \in \overline{\Omega}_m^h \setminus \theta_m^h, \ m = 1, \ldots, M-1; \\ v_M^{(n)}(p), & p \in \overline{\Omega}_M^h. \end{cases} \tag{3.2}$$

4. Stopping criterion: if

$$\max_{p \in \overline{\Omega}^h} |v^{(n)}(p) - v^{(n-1)}(p)| \le \Delta,$$

where Δ is the required accuracy, then stop; otherwise continue iteration by going to Step 2.

The following theorem gives the convergence property of the monotone domain decomposition algorithm (DD).

Theorem 5. *Let $\overline{v}^{(0)}$, $\underline{v}^{(0)}$ be upper and lower solutions of (1.15). Then the upper and lower sequences $\{\overline{v}^{(n)}\}$ and $\{\underline{v}^{(n)}\}$ generated by the monotone* DD *algorithm (3.1)–*

(3.2), converge monotonically from above and below, respectively, to the unique solution v of (1.15):

$$\underline{v}^{(n-1)}(p) \le \underline{v}^{(n)}(p) \le v(p) \le \overline{v}^{(n)}(p) \le \overline{v}^{(n-1)}(p), \quad p \in \overline{\Omega}^h, \quad n \ge 1.$$

Proof. We consider only the case of upper sequences, as the case of lower sequences is proved in a similar manner.

Let $\overline{v}^{(0)}(p)$, $p \in \overline{\Omega}^h$ be an upper solution of (1.15) satisfying the boundary condition. From (3.1) with $n = 1$, $m = 1$, we have

$$(\mathcal{L} + c^*) z_1^{(n)}(p) = -\mathcal{R}(p, \overline{v}^{(0)}) \le 0, \quad p \in \Omega_1^h,$$

$$z_1^{(1)}(0) = 0 \quad p \in \partial\Omega_1^h.$$

By the maximum principle in Lemma 1, we conclude

$$z_1^{(1)}(p) \le 0, \quad p \in \overline{\Omega}_1^h. \tag{3.3}$$

From here and (3.1) with $n = 1$ and $m = 2$, we have

$$(\mathcal{L} + c^*) z_2^{(n)}(p) = -\mathcal{R}(p, \overline{v}^{(0)}) \le 0, \quad p \in \Omega_2^h,$$

$$z_2^{(1)}(\gamma_2^{hl}) = z_1^{(1)}(\gamma_2^{hl}) \le 0, \quad z_2^{(1)}(\gamma_2^{hr}) = 0, \quad z_2^{(1)}(\gamma_2^{h0}) = 0.$$

By the maximum principle in Lemma 1, we conclude

$$z_2^{(1)}(p) \le 0, \quad p \in \overline{\Omega}_2^h,$$

and by induction on m, $z_m^{(1)}(p) \le 0$, $p \in \overline{\Omega}_m^h$, $m = 2, \ldots, M$. From here and (3.3), it follows that

$$z_m^{(1)}(p) \le 0, \quad p \in \overline{\Omega}_m^h, \quad m = 1, \ldots, M. \tag{3.4}$$

We now prove that $v^{(1)}(p)$ defined in Step 3 is an upper solution, that is,

$$\mathcal{R}(p, v^{(1)}) \ge 0, \quad p \in \Omega^h. \tag{3.5}$$

From (3.1) and the mean value theorem,

$$\begin{aligned}
\mathcal{R}(p, v_m^{(1)}) &= \mathcal{R}(p, \overline{v}^{(0)} + z_m^{(1)}) \\
&= \mathcal{L}\overline{v}^{(0)}(p) + \mathcal{L}z_m^{(1)}(p) + f(p, \overline{v}^{(0)} + z_m^{(1)}) \\
&= \mathcal{L}\overline{v}^{(0)}(p) + f(p, \overline{v}^{(0)}) + \mathcal{L}z_m^{(1)}(p) + f(p, \overline{v}^{(0)} + z_m^{(1)}) - f(p, \overline{v}^{(0)}) \\
&= \mathcal{R}(p, \overline{v}^{(0)}) + \mathcal{L}z_m^{(1)}(p) + f_v^{(1)}(p) z_m^{(1)}(p) \\
&= -(\mathcal{L} + c^*) z_m^{(1)}(p) + \mathcal{L}z_m^{(1)}(p) + f_v^{(1)}(p) z_m^{(1)}(p) \\
&= -(c^* - f_v^{(1)}(p)) z_m^{(1)}(p), \quad p \in \Omega_m^h, \quad m = 1, \ldots, M,
\end{aligned}$$

where $f_v^{(1)}(p) = f_v[p, \overline{v}^{(0)}(p) + \Theta_m^{(1)}(p) z_m^{(1)}(p)]$, $0 < \Theta_m^{(1)}(p) < 1$. From here, (1.20) and (3.4), we have

$$\mathcal{R}(p, v_m^{(1)}) = -(c^* - f_v^{(1)}(p)) z_m^{(1)} \geq 0 \quad p \in \Omega_m^h, \quad m = 1, \ldots, M.$$

From here and (3.2), we have

$$\mathcal{R}(p, v^{(1)}) \geq 0, \quad p \in \Omega^h \setminus \gamma^{hl}, \quad \gamma^{hl} = \bigcup_{m=2}^{M} \gamma_m^{hl}.$$

We therefore only need to verify (3.5) on the boundaries γ_m^{hl}, $m = 2, \ldots, M$. Let

$$\zeta_m^{(n)}(p) = v_m^{(n)}(p) - v_{m+1}^{(n)}(p) = z_m^{(n)}(p) - z_{m+1}^{(n)}(p), \quad p \in \overline{\theta}_m^h.$$

For $m = 1, \ldots, M-1$, from (3.1) we have

$$(\mathcal{L} + c^*) \zeta_m^{(1)}(p) = 0, \quad p \in \theta_m^h,$$

$$\zeta_m^{(1)}(\gamma_m^{hr}) \geq 0, \quad \zeta_m^{(1)}(\gamma_{m+1}^{hl}) = 0, \quad \zeta_m^{(1)}(\gamma_1^{h0} \cap \gamma_2^{h0}) = 0,$$

and by the maximum principle in Lemma 1, conclude $\zeta_m^{(1)} \geq 0$, $p \in \overline{\theta}_m^h$, $m = 1, \ldots, M-1$. From here, it follows that

$$v_m^{(1)}(p) \geq v_{m+1}^{(1)}(p), \quad p \in \overline{\theta}_m^h, \quad m = 1, \ldots, M-1.$$

For $p \in \gamma_{m+1}^{hl}$, $m = 1, \ldots, M-1$, we have

$$\begin{aligned}
\mathcal{R}(p, v^{(1)}) &= (d(p) + c^*)v^{(1)}(p) - \sum_{p' \in \sigma'(p)} e(p, p')v^{(1)}(p') \\
&= (d(p) + c^*)v^{(1)}(p) - \sum_{p' \in \sigma_L(p)} e(p, p')v^{(1)}(p') \\
&\quad - \sum_{p' \in \sigma_U(p)} e(p, p')v^{(1)}(p') \\
&= (d(p) + c^*)v_m^{(1)}(p) - \sum_{p' \in \sigma_L(p)} e(p, p')v_m^{(1)}(p') \\
&\quad - \sum_{p' \in \sigma_U(p)} e(p, p')v_{m+1}^{(1)}(p') \\
&\geq (d(p) + c^*)v_m^{(1)}(p) - \sum_{p' \in \sigma_L(p)} e(p, p')v_m^{(1)}(p') \\
&\quad - \sum_{p' \in \sigma_U(p)} e(p, p')v_m^{(1)}(p') \\
&= \mathcal{R}(p, v_m^{(1)}) \geq 0,
\end{aligned}$$

where $\sigma_U(p)$ and $\sigma_L(p)$ are the set of stencil points corresponding to a strictly upper and lower triangular part of $\sigma(p)$, respectively, such that $\sigma'(p) = \sigma_U(p) \cup \sigma_L(p)$. Thus, $v^{(1)}$ is an upper solution to (1.15). By induction on n, $\{\overline{v}^{(n)}\}$ is a monotonically decreasing sequence of upper solutions bounded below by $\underline{v}$ (1.19), where $\underline{v}$ is any lower solution. Let $v(p)$ be defined by

$$v(p) = \lim_{n \to \infty} \overline{v}^{(n)}(p), \quad p \in \overline{\Omega}^h.$$

We now prove that $v(p)$ is the solution to (1.15). From (3.1),

$$\lim_{n \to \infty} z_m^{(n)} = \lim_{n \to \infty} (v_m^{(n)} - v_m^{(n-1)}) = 0, \quad m = 1, \ldots, M.$$

Using (3.1) and (3.2), we have

$$\begin{aligned}
\mathcal{R}(p, v) &= \lim_{n \to \infty} \mathcal{R}(p, v^{(n)}) = \lim_{n \to \infty} \mathcal{R}(p, v_m^{(n)}) = -\lim_{n \to \infty} (\mathcal{L} + c^*) z_m^{(n)}(p) = 0, \\
&\quad p \in \Omega_m^h \setminus \theta_m^h, \quad m = 1, \ldots, M-1, \quad p \in \Omega_M.
\end{aligned}$$

Thus,

$$\mathcal{R}(p, v) = 0, \quad p \in \Omega^h.$$

Since, $v(p) = g(p)$, $p \in \partial\Omega^h$, we conclude that $v(p)$ is the solution of (1.15). □

3.2.1 Numerical experiments

We apply the monotone domain decomposition algorithm (3.1)–(3.2) to two test problems, a convection-diffusion problem with parabolic layers and an anisotropic convection-diffusion problem. For the following numerical experiments, the required accuracy is $\Delta = 10^{-6}$, and the number of mesh points in the x- and y-directions are set equal to N.

To solve the linear difference problems (3.1), GMRES with restarts is used with a diagonal preconditioner. Within GMRES, the required accuracy is 10^{-6}, the maximum number of iterations is 50 and a maximum of 20 restarts.

GMRES is an iterative method for finding a numerical solution to a nonsymmetric system in the form $Ax = b$ [60]. In the nth iteration, GMRES approximates the solution by using Arnoldi iterations to find a vector in a Krylov subspace K_n with the minimal residual, where $K_n = \text{span}\{b, Ab, A^2b, \ldots, A^{n-1}b\}$. One of the disadvantages of GMRES is the amount of storage required. To overcome this, GMRES with restarts is used. When the solver is restarted all the accumulated data are cleared and the last set of results is used as the initial data. Preconditions are often used within GMRES to accelerate the convergence. Using a preconditioner P, GMRES solves the problem $PAx = Pb$. We use a diagonal preconditioner, where P is the inverse of the diagonal part of A.

For the following experiments, we focus on balanced domain decompositions where M is even and $M/2$ vertical strips are placed within the boundary layer.

To decompose the domain $\overline{\Omega}^h$ into M overlapping subdomains, we begin by splitting the domain into M subdomains $\overline{Q}_m^h$, $m = 1, \ldots, M$,

$$\overline{Q}_m^h = \{(x_i, y_j) : (m-1)N/M \le i \le mN/M, 0 \le j \le N\},$$
$$\overline{Q}_m^h \cap \overline{Q}_{m+1}^h = \gamma_m^h, \quad \gamma_m^h = \{(x_i, y_j) : i = mN/M, 0 \le j \le N\},$$

where γ_m^h is the interfacial boundary between subdomains $\overline{Q}_m^h$ and $\overline{Q}_{m+1}^h$.

We consider two locations of overlaps. In the first one, we choose overlaps on the left of the interfacial boundaries

$$\overline{\theta}_m^h = \{(x_i, y_j) : mN/M - d \le i \le mN/M, 0 \le j \le N\}, \quad 1 \le d \le N/M - 1,$$

where the minimal and maximal overlap sizes are $d = 1$ and $d = N/M - 1$, respectively. Thus, the overlapping subdomains are defined by

$$\overline{\Omega}_m^h = \{(x_i, y_j) : (m-1)N/M - d \le i \le mN/M, 0 \le j \le N\}, \quad m = 2, \ldots, M,$$

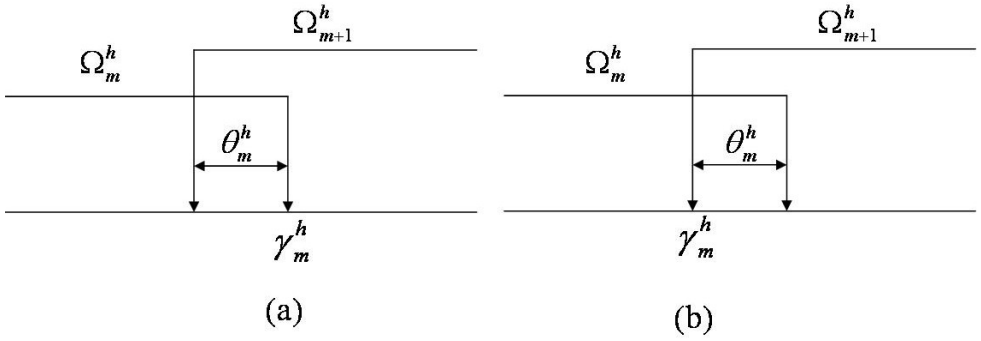

Figure 3.2: (a) Location of the overlap on the left. (b) Location of the overlap on the right.

where $\overline{\Omega}^h_1 = \overline{Q}^h_1$. Figure 3.2(a) illustrates this case. If we choose overlaps on the right of the interfacial boundaries, then in a similar way, we define the overlapping subdomains in the form

$$\overline{\Omega}^h_m = \{(x_i, y_j) : (m-1)N/M \le i \le mN/M + d, 0 \le j \le N\},$$
$$1 \le d \le N/M - 1, \quad m = 1, \ldots, M-1, \quad \overline{\Omega}^h_M = \overline{Q}^h_M.$$

Figure 3.2(b) illustrates this case.

We apply the monotone DD algorithm (3.1)–(3.2) to the test problem (1.12) with $b_1(x, y) = 1$, $f(x, y, u) = 1 - \exp(-u)$ and $g(x, y) = 1$. Table 3.1 displays the iteration counts and execution times of the monotone DD algorithm for the two different locations of the overlaps. Table 3.1 shows that for the minimal overlap size the iteration count is smaller when the overlaps occur on the left of the interfacial boundaries. However, for the maximal overlap size, there is very little difference in the iteration counts of the monotone DD algorithm. For both the minimal and maximal overlap sizes there is little difference in the execution times for the difference locations of the overlaps. Due to the above reasons, the following numerical experiments have the overlaps located on the left of the interfacial boundaries. We define the serial acceleration of the domain decomposition algorithm as the execution time of the undecomposed method ($M = 1$) divided by the minimum execution time of the domain decomposition algorithm. A serial acceleration greater than 1 indicates an advantage in the domain decomposition algorithm.

ε	10^{-1}				10^{-2}				10^{-3}				10^{-4}			
M	1	2	4	8	1	2	4	8	1	2	4	8	1	2	4	8
	Iteration counts															
Left	6	$\frac{19}{6}$	$\frac{21}{7}$	$\frac{29}{12}$	6	$\frac{7}{6}$	$\frac{21}{7}$	$\frac{32}{13}$	6	$\frac{6}{6}$	$\frac{24}{7}$	$\frac{39}{16}$	6	$\frac{6}{6}$	$\frac{25}{7}$	$\frac{40}{16}$
Right	6	$\frac{19}{6}$	$\frac{21}{7}$	$\frac{29}{12}$	6	$\frac{25}{6}$	$\frac{28}{7}$	$\frac{36}{14}$	6	$\frac{32}{6}$	$\frac{36}{7}$	$\frac{48}{17}$	6	$\frac{33}{6}$	$\frac{37}{7}$	$\frac{49}{18}$
	Execution times (seconds)															
Left	5	$\frac{10}{6}$	$\frac{5}{5}$	$\frac{3}{2}$	10	$\frac{5}{12}$	$\frac{6}{6}$	$\frac{8}{7}$	10	$\frac{4}{12}$	$\frac{8}{6}$	$\frac{9}{7}$	10	$\frac{4}{12}$	$\frac{7}{6}$	$\frac{9}{7}$
Right	5	$\frac{11}{8}$	$\frac{4}{5}$	$\frac{3}{2}$	10	$\frac{10}{12}$	$\frac{7}{6}$	$\frac{8}{7}$	10	$\frac{6}{10}$	$\frac{6}{6}$	$\frac{9}{7}$	10	$\frac{4}{10}$	$\frac{5}{6}$	$\frac{9}{8}$

Table 3.1: Iteration counts and execution times of the monotone DD algorithm with the overlaps located to the left and right of the interfacial boundaries using the minimal and maximal overlap sizes above and below the line, respectively.

3.2.2 Convection-diffusion problem with parabolic boundary layers

Consider the test problem

$$-\varepsilon(u_{xx}+u_{yy})+u_x+f(x,y,u)=0, \quad (x,y)\in\Omega=(0,1)\times(0,1), \tag{3.6}$$

$$f(x,y,u)=1-\exp(-u), \quad u=1 \text{ on } \partial\Omega.$$

This problem is the convection-diffusion problem with parabolic boundary layers (1.9), which is characterised by an elliptic boundary layer close to $x=1$ and by parabolic boundary layers close to $y=0$ and $y=1$. To solve this problem numerically, we use the two dimensional piecewise uniform Shishkin mesh (1.11) constructed in Section 1.4.2. We also require the constants c_*, c^* from (1.20). For this test problem, we have $f_u=\exp(-u)$ and the reduced equation

$$u_x+1-\exp(-u)=0, \quad (x,y)\in\Omega.$$

Thus, $u_r=0$ is the solution to the reduced equation. By writing the difference scheme at an interior mesh point as in (1.6), where the coefficients satisfy the inequalities (1.7), we can see that $\overline{u}(p)=1$, $p\in\overline{\Omega}$ and $\underline{u}(p)=0$, $p\in\Omega$, $\underline{u}(p)=1$, $p\in\partial\Omega$ are upper and lower solutions, respectively. Therefore $0\le u(p)\le 1$, $\exp(-1)\le f_u(p)\le 1$, $p\in\Omega^h$, and

$$c_*=\exp(-1), \quad c^*=1. \tag{3.7}$$

We also require an initial solution. For the test problem (3.6), $\overline{u}^{(0)}(p) = 1$, $p \in \overline{\Omega}^h$ is an initial upper solution.

Table 3.2 displays the iteration counts and execution times over varying numbers of subdomains for different values of ε and N for the monotone DD algorithm. The results are displayed for the minimal and maximal overlap size above and below the line, respectively. The iteration counts in Table 3.2 show that the maximal overlap gives the smallest iteration counts. The table also shows that as the number of subdomains increases the iteration counts also increase. The iteration counts for all ε smaller than 10^{-4} are the same, therefore we can conclude that the monotone DD algorithm uniformly converges in its iteration counts with respect to ε.

N	32				64					128					
ε/M	1	2	4	8	1	2	4	8	16	1	2	4	8	16	32
	Iteration counts														
10^{-1}	8	$\frac{20}{8}$	$\frac{27}{11}$	$\frac{42}{30}$	8	$\frac{34}{8}$	$\frac{46}{10}$	$\frac{73}{25}$	$\frac{129}{89}$	8	$\frac{60}{8}$	$\frac{82}{10}$	$\frac{131}{23}$	$\frac{234}{75}$	$\frac{430}{297}$
10^{-2}	9	$\frac{9}{9}$	$\frac{20}{10}$	$\frac{35}{25}$	9	$\frac{9}{9}$	$\frac{29}{9}$	$\frac{46}{17}$	$\frac{84}{59}$	9	$\frac{11}{9}$	$\frac{44}{9}$	$\frac{68}{13}$	$\frac{120}{39}$	$\frac{226}{155}$
10^{-3}	9	$\frac{9}{9}$	$\frac{21}{10}$	$\frac{36}{27}$	9	$\frac{9}{9}$	$\frac{29}{9}$	$\frac{49}{18}$	$\frac{91}{64}$	9	$\frac{9}{9}$	$\frac{44}{9}$	$\frac{70}{14}$	$\frac{128}{43}$	$\frac{244}{170}$
10^{-4}	9	$\frac{9}{9}$	$\frac{21}{10}$	$\frac{37}{27}$	9	$\frac{9}{9}$	$\frac{29}{9}$	$\frac{49}{18}$	$\frac{92}{65}$	9	$\frac{9}{9}$	$\frac{44}{9}$	$\frac{71}{14}$	$\frac{130}{44}$	$\frac{249}{173}$
10^{-5}	9	$\frac{9}{9}$	$\frac{21}{10}$	$\frac{37}{27}$	9	$\frac{9}{9}$	$\frac{29}{9}$	$\frac{49}{18}$	$\frac{92}{65}$	9	$\frac{9}{9}$	$\frac{44}{9}$	$\frac{71}{14}$	$\frac{130}{44}$	$\frac{249}{173}$
10^{-6}	9	$\frac{9}{9}$	$\frac{21}{10}$	$\frac{37}{27}$	9	$\frac{9}{9}$	$\frac{29}{9}$	$\frac{49}{18}$	$\frac{92}{65}$	9	$\frac{9}{9}$	$\frac{44}{9}$	$\frac{71}{14}$	$\frac{130}{44}$	$\frac{249}{173}$
	Execution times (seconds)														
10^{-1}	0.29	$\frac{0.43}{0.35}$	$\frac{0.30}{0.28}$	$\frac{0.32}{0.31}$	5.3	$\frac{5.7}{6.0}$	$\frac{4.6}{2.8}$	$\frac{3.4}{3.3}$	$\frac{3.9}{3.6}$	93	$\frac{443}{137}$	$\frac{122}{115}$	$\frac{57}{45}$	$\frac{44}{40}$	$\frac{68}{59}$
10^{-2}	0.30	$\frac{0.16}{0.33}$	$\frac{0.17}{0.18}$	$\frac{0.20}{0.18}$	7.9	$\frac{1.2}{7.2}$	$\frac{2.1}{2.0}$	$\frac{1.8}{1.6}$	$\frac{2.9}{2.6}$	122	$\frac{71}{165}$	$\frac{54}{98}$	$\frac{26}{24}$	$\frac{30}{22}$	$\frac{51}{44}$
10^{-3}	0.27	$\frac{0.11}{0.28}$	$\frac{0.13}{0.13}$	$\frac{0.16}{0.15}$	8.4	$\frac{0.9}{7.2}$	$\frac{1.5}{1.4}$	$\frac{1.4}{1.1}$	$\frac{2.3}{1.9}$	172	$\frac{54}{191}$	$\frac{37}{81}$	$\frac{16}{15}$	$\frac{18}{13}$	$\frac{38}{30}$
10^{-4}	0.27	$\frac{0.11}{0.29}$	$\frac{0.12}{0.13}$	$\frac{0.17}{0.15}$	8.4	$\frac{0.9}{7.4}$	$\frac{1.5}{1.4}$	$\frac{1.4}{1.0}$	$\frac{2.4}{2.0}$	193	$\frac{59}{208}$	$\frac{37}{85}$	$\frac{17}{15}$	$\frac{20}{13}$	$\frac{40}{32}$
10^{-5}	0.27	$\frac{0.11}{0.29}$	$\frac{0.12}{0.11}$	$\frac{0.15}{0.13}$	8.5	$\frac{1.0}{7.5}$	$\frac{1.4}{1.2}$	$\frac{1.2}{0.9}$	$\frac{2.1}{1.8}$	199	$\frac{58}{208}$	$\frac{34}{85}$	$\frac{16}{13}$	$\frac{17}{11}$	$\frac{37}{29}$
10^{-6}	0.27	$\frac{0.11}{0.29}$	$\frac{0.12}{0.11}$	$\frac{0.15}{0.13}$	8.5	$\frac{1.0}{7.4}$	$\frac{1.4}{1.2}$	$\frac{1.2}{0.9}$	$\frac{2.1}{1.7}$	197	$\frac{58}{207}$	$\frac{36}{84}$	$\frac{15}{13}$	$\frac{17}{11}$	$\frac{37}{29}$

Table 3.2: Iteration counts and execution times of the monotone DD algorithm using the minimal and maximal overlap size above and below the line, respectively, for the test problem (3.6).

Using the execution times in Table 3.2, we display the serial acceleration for the monotone DD algorithm in Figure 3.3. Figure 3.3 shows that for all values of ε and N, the

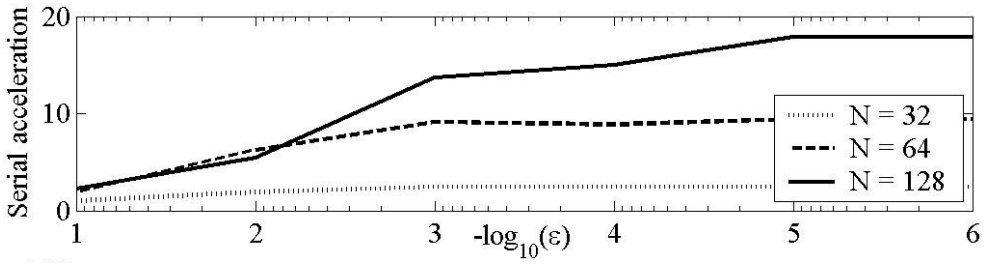

Figure 3.3: Serial acceleration of the monotone DD algorithm for the test problem (3.6).

serial acceleration is greater than 1 indicating an advantage in using the monotone DD algorithm. As ε decreases the serial acceleration from using the monotone DD algorithm increases. For $\varepsilon < 10^{-2}$, as N increases the serial acceleration also increases.

3.2.3 Anisotropic convection-diffusion problem

Consider the test problem

$$-\varepsilon u_{xx} - u_{yy} + u_x + f(x, y, u) = 0, \quad (x, y) \in \Omega = (0, 1) \times (0, 1), \tag{3.8}$$

$$f(x, y, u) = 1 - \exp(-u), \quad u = 1 \text{ on } \partial\Omega.$$

This problem is the anisotropic convection-diffusion problem (1.12) which is characterised by an elliptic boundary layer close to $x = 1$. To solve this problem numerically, we use the two dimensional piecewise uniform Shishkin mesh (1.8) constructed in Section 1.4.3. We also require the constants c_*, c^* from (1.20). For this test problem, we have $f_u = \exp(-u)$ and the reduced equation

$$u_{yy} + u_x + 1 - \exp(-u) = 0, \quad (x, y) \in \Omega.$$

Thus, $u_r = 0$ is the solution to the reduced equation. By writing the difference scheme at an interior mesh point as in (1.6), where the coefficients satisfy the inequalities (1.7), we can see that $\overline{u}(p) = 1$, $p \in \overline{\Omega}$ and $\underline{u}(p) = 0$, $p \in \Omega$, $\underline{u}(p) = 1$, $p \in \partial\Omega$ are upper and lower solutions, respectively. Therefore $0 \leq u(p) \leq 1$, $\exp(-1) \leq f_u(p) \leq 1$, $p \in \Omega^h$ and

$$c_* = \exp(-1), \quad c^* = 1. \tag{3.9}$$

We also require an initial solution. For the test problem (3.8), $\overline{u}^{(0)}(p) = 1$, $p \in \overline{\Omega}^h$ is an initial upper solution.

Table 3.3 displays the iteration counts and execution times over varying numbers of subdomains for different values of ε and N for the monotone DD algorithm. The results are displayed for the minimal and maximal overlap size above and below the line, respectively. These iteration counts show that for the larger overlap size, the iteration counts are less. Table 3.3 also shows that as the number of subdomains increases so does the number of iterations needed for the method to converge. We can also conclude from Table 3.3 that the monotone DD algorithm uniformly converges in its iteration counts with respect to ε.

Using the execution times in Table 3.3, we display the serial acceleration for the monotone DD algorithm in Figure 3.4. Figure 3.4 shows that for all values of ε and N, the serial acceleration is greater than 1 indicating an advantage in using the monotone DD algorithm.

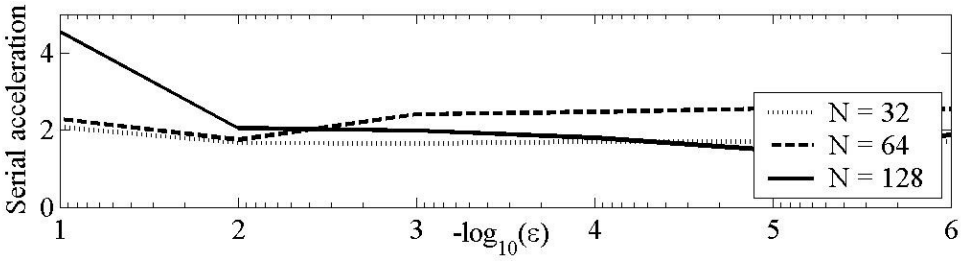

Figure 3.4: Serial acceleration of the monotone DD algorithm for the test problem (3.8).

3.2.4 Numerical observations

Through the numerical experiments, we observe that for both of the test problems the monotone DD algorithm is parameter uniformly convergent in its iteration counts. We can also see that the serial acceleration for both of the test problems is greater than one when compared to the undecomposed method. This indicates an advantage in using the monotone DD algorithm.

3.3 Two-level monotone domain decomposition algorithm

In this section, we investigate the two-level monotone domain decomposition algorithm for solving the nonlinear difference scheme (1.15) based on the multiplicative Schwarz

N	32				64					128					
ε/M	1	2	4	8	1	2	4	8	16	1	2	4	8	16	32
	Iteration counts														
10^{-1}	6	$\frac{12}{6}$	$\frac{13}{7}$	$\frac{17}{13}$	6	$\frac{19}{6}$	$\frac{21}{7}$	$\frac{29}{11}$	$\frac{46}{32}$	6	$\frac{32}{6}$	$\frac{37}{7}$	$\frac{51}{10}$	$\frac{83}{28}$	$\frac{148}{102}$
10^{-2}	6	$\frac{6}{6}$	$\frac{15}{7}$	$\frac{23}{17}$	6	$\frac{7}{6}$	$\frac{21}{7}$	$\frac{32}{12}$	$\frac{55}{38}$	6	$\frac{8}{6}$	$\frac{32}{6}$	$\frac{47}{10}$	$\frac{79}{26}$	$\frac{145}{100}$
10^{-3}	6	$\frac{6}{6}$	$\frac{17}{8}$	$\frac{27}{20}$	6	$\frac{6}{6}$	$\frac{24}{7}$	$\frac{39}{14}$	$\frac{70}{49}$	6	$\frac{9}{6}$	$\frac{37}{7}$	$\frac{58}{11}$	$\frac{103}{34}$	$\frac{192}{134}$
10^{-4}	6	$\frac{6}{6}$	$\frac{17}{8}$	$\frac{28}{20}$	6	$\frac{6}{6}$	$\frac{25}{7}$	$\frac{40}{15}$	$\frac{72}{51}$	6	$\frac{11}{6}$	$\frac{37}{7}$	$\frac{59}{12}$	$\frac{106}{36}$	$\frac{200}{139}$
10^{-5}	6	$\frac{6}{6}$	$\frac{17}{8}$	$\frac{28}{20}$	6	$\frac{6}{6}$	$\frac{25}{7}$	$\frac{40}{15}$	$\frac{73}{51}$	5	$\frac{10}{5}$	$\frac{37}{7}$	$\frac{60}{12}$	$\frac{107}{36}$	$\frac{201}{140}$
10^{-6}	6	$\frac{6}{6}$	$\frac{17}{8}$	$\frac{28}{20}$	6	$\frac{6}{6}$	$\frac{25}{7}$	$\frac{40}{15}$	$\frac{73}{51}$	6	$\frac{7}{7}$	$\frac{37}{7}$	$\frac{60}{12}$	$\frac{107}{36}$	$\frac{201}{140}$
	Execution times (seconds)														
10^{-1}	0.43	$\frac{0.44}{0.51}$	$\frac{0.25}{0.31}$	$\frac{0.21}{0.23}$	5.6	$\frac{10.3}{7.8}$	$\frac{4.7}{4.9}$	$\frac{2.9}{2.5}$	$\frac{2.7}{2.7}$	138	$\frac{538}{190}$	$\frac{182}{168}$	$\frac{64}{62}$	$\frac{38}{30}$	$\frac{39}{39}$
10^{-2}	0.50	$\frac{0.30}{0.67}$	$\frac{0.44}{0.51}$	$\frac{0.50}{0.52}$	9.2	$\frac{5.3}{11.7}$	$\frac{6.5}{7.0}$	$\frac{8.0}{6.7}$	$\frac{9.9}{9.6}$	171	$\frac{146}{242}$	$\frac{143}{174}$	$\frac{88}{85}$	$\frac{117}{84}$	$\frac{158}{163}$
10^{-3}	0.49	$\frac{0.30}{0.68}$	$\frac{0.49}{0.56}$	$\frac{0.60}{0.59}$	9.3	$\frac{3.9}{12.2}$	$\frac{7.6}{6.4}$	$\frac{9.7}{7.3}$	$\frac{14.4}{13.2}$	184	$\frac{214}{268}$	$\frac{153}{225}$	$\frac{118}{92}$	$\frac{160}{110}$	$\frac{250}{230}$
10^{-4}	0.49	$\frac{0.29}{0.68}$	$\frac{0.48}{0.53}$	$\frac{0.61}{0.58}$	9.2	$\frac{3.7}{12.2}$	$\frac{6.7}{6.4}$	$\frac{9.1}{7.5}$	$\frac{14.1}{13.6}$	184	$\frac{257}{269}$	$\frac{134}{224}$	$\frac{117}{102}$	$\frac{167}{113}$	$\frac{259}{240}$
10^{-5}	0.49	$\frac{0.29}{0.68}$	$\frac{0.47}{0.52}$	$\frac{0.59}{0.58}$	9.4	$\frac{3.7}{12.4}$	$\frac{6.9}{6.3}$	$\frac{9.0}{7.4}$	$\frac{13.6}{12.9}$	147	$\frac{233}{215}$	$\frac{129}{221}$	$\frac{110}{99}$	$\frac{154}{107}$	$\frac{258}{230}$
10^{-6}	0.49	$\frac{0.29}{0.68}$	$\frac{0.48}{0.52}$	$\frac{0.56}{0.55}$	9.4	$\frac{3.7}{12.4}$	$\frac{6.8}{6.2}$	$\frac{8.8}{7.4}$	$\frac{13.5}{12.5}$	184	$\frac{168}{322}$	$\frac{127}{219}$	$\frac{100}{98}$	$\frac{153}{103}$	$\frac{260}{227}$

Table 3.3: Iteration counts and execution times of the monotone DD algorithm using the minimal and maximal overlap size above and below the line, respectively, for the test problem (3.8).

algorithm. The domain decomposition algorithm consists of the two iterative processes: outer and inner iterations.

3.3.1 The outer iterates

We assume that the computational domain Ω is two dimensional rectangular domain and use the rectangular mesh $\overline{\Omega}^h = \overline{\Omega}^{hx} \times \overline{\Omega}^{hy}$ given in (1.3).

We introduce the set of the overlapping vertical strips $\overline{\Omega}_m$, $m = 1, \ldots, M$, with the boundaries

$$\partial\Omega_m = \gamma_m^l \cup \gamma_m^r \cup \gamma_m^0,$$

where γ_m^l and γ_m^r are the left and right boundaries of $\overline{\Omega}_m$, respectively, and γ_m^0 belongs to

the boundary of $\overline{\Omega}$. A fragment of the domain decomposition is illustrated on Figure 3.1. Thus,

$$\overline{\Omega}_m \cap \overline{\Omega}_{m+1} = \overline{\theta}_m, \quad m = 1, \ldots, M-1,$$

where $\overline{\theta}_m$ is the overlap between two subdomains $\overline{\Omega}_m$ and $\overline{\Omega}_{m+1}$.

We assume that the left $\gamma_m^{hl} = \gamma_m^l \cap \overline{\Omega}^h$ and right $\gamma_m^{hr} = \gamma_m^r \cap \overline{\Omega}^h$ boundaries of $\overline{\Omega}_m^h = \overline{\Omega}_m \cap \overline{\Omega}^h$, $m = 1, \ldots, M$, are mesh lines.

At the level of the outer iterates of the algorithm from [37], one complete iterative step includes solving a sequence of M problems on subdomains $\overline{\Omega}_m^h$, $m = 1, \ldots, M$, in serial. For computing the problem on subdomain $\overline{\Omega}_m^h$, $m > 1$, the Dirichlet boundary condition on the left boundary is updated by using the solution of the problem on subdomain $\overline{\Omega}_{m-1}^h$ (previous substep of the outer iterative step). The solution on the right boundary is set equal to the solution found from the previous outer iteration $v^{(n-1)}(p)$ and the solutions on the top and bottom boundaries are equal to the original boundary condition $g(p)$.

The algorithm for the outer iterates of the two-level monotone domain decomposition algorithm is as follows:

1. Initialisation: On the whole mesh $\overline{\Omega}^h$, choose an initial function $v^{(0)}(p)$, $p \in \overline{\Omega}^h$, satisfying the boundary condition $v^{(0)}(p) = g(p)$ on $\partial\Omega^h$.

2. On subdomains $\overline{\Omega}_m^h$, $m = 1, \ldots, M$, compute in serial mesh functions $v_m^{(n)}(p)$, $m = 1, \ldots, M$, satisfying the difference schemes

$$\mathcal{L}v_m^{(n)}(p) + f(p, v_m^{(n)}) = 0, \quad p \in \Omega_m^h, \tag{3.10}$$

with the boundary conditions

$$v_m^{(n)}(p) = \begin{cases} v_{m-1}^{(n)}(p), & p \in \gamma_m^{hl}, \\ v^{(n-1)}(p), & p \in \gamma_m^{hr}, \\ g(p), & p \in \gamma_m^{h0}, \end{cases}$$

where

$$\gamma_m^{hl} = \gamma_m^l \cap \overline{\Omega}_m^h, \quad \gamma_m^{hr} = \gamma_m^r \cap \overline{\Omega}_m^h, \quad \gamma_m^{h0} = \gamma_m^0 \cap \overline{\Omega}_m^h.$$

3. Compute the solution $v^{(n)}(p)$, $p \in \overline{\Omega}^h$ by piecing together the solutions on the subdomains

$$v^{(n)}(p) = \begin{cases} v_m^{(n)}(p), & p \in \overline{\Omega}_m^h \setminus \theta_m^h, \quad m = 1, \ldots, M-1; \\ v_M^{(n)}(p), & p \in \overline{\Omega}_M^h. \end{cases} \tag{3.11}$$

On $\overline{\Omega}_m^h \setminus \theta_m^h$, we set $v^{(n)}$ equal to the solution $v_m^{(n)}$, and overlap $\overline{\theta}_m^h$ is included in $\overline{\Omega}_{m+1}^h$.

4. Stopping criterion: if

$$\max_{p \in \overline{\Omega}^h} |v^{(n)}(p) - v^{(n-1)}(p)| \leq \Delta,$$

where Δ is the required accuracy, then stop; otherwise continue iterating by going to Step 2.

3.3.2 The inner iterates

We assume that f from (1.15) satisfies the two-sided constraint (1.20).

For solving the nonlinear problems (3.10), we use the inner iterates based on the nonoverlapping box-subdomains from [18]. We decompose each subdomain $\overline{\Omega}_m$ into S_m nonoverlapping box-subdomains Ω_{ms}, $s = 1, \ldots, S_m$, with the boundaries

$$\partial\Omega_{ms} = \gamma_{ms}^l \cup \gamma_{ms}^r \cup \gamma_m^b \cup \gamma_{ms}^t,$$

where γ_{ms}^l, γ_{ms}^r, γ_m^b and γ_{ms}^t are the left, right, bottom and top boundaries of $\overline{\Omega}_{ms}$. We assume that the bottom $\gamma_{ms}^{hb} = \rho_{ms}^b \cap \overline{\Omega}^h$ and top $\gamma_{ms}^{ht} = \rho_{ms}^t \cap \overline{\Omega}^h$ boundaries of $\overline{\Omega}_{ms}$, $s = 1, \ldots, S_m$, are mesh lines. Additionally, we introduce $(S_m - 1)$ interfacial subdomains ϑ_{ms}, $s = 1, \ldots, S_m - 1$ (horizontal strips), with the boundaries

$$\partial\vartheta_{ms} = \rho_{ms}^l \cup \rho_{ms}^r \cup \rho_{ms}^b \cup \rho_{ms}^t,$$

where ρ_{ms}^l, ρ_{ms}^r, ρ_{ms}^b and ρ_{ms}^t are the left, right, bottom and top boundaries of $\overline{\vartheta}_{ms}$. We assume that the bottom $\rho_{ms}^{hb} = \rho_{ms}^b \cap \overline{\Omega}^h$ and top $\rho_{ms}^{ht} = \rho_{ms}^t \cap \overline{\Omega}^h$ boundaries of $\overline{\vartheta}_{ms}$, $s = 1, \ldots, S_m - 1$, are mesh lines. Figure 3.5 illustrates a fragment of the box-domain decomposition. On each iterative step of the inner iterates, we first solve problems on the

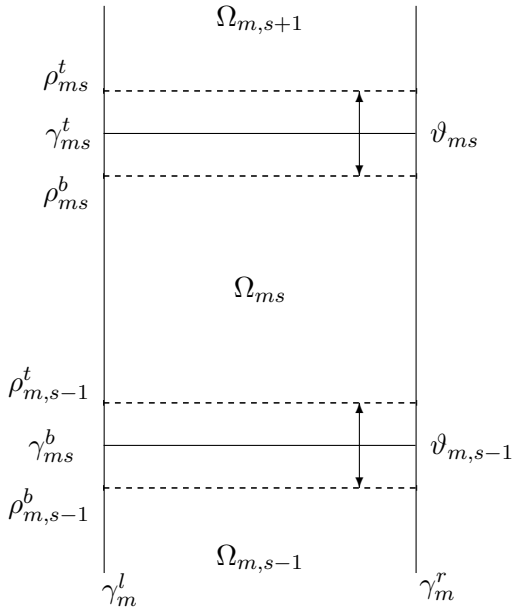

Figure 3.5: Fragment of the box-domain decomposition.

nonoverlapping subdomains $\overline{\Omega}^h_{ms} = \overline{\Omega}_{ms} \cap \overline{\Omega}^h_m$, $s = 1, \ldots, S_m$, with Dirichlet boundary conditions passed from the previous iterate. Then Dirichlet data are passed from these subdomains to the horizontal interfacial subdomains $\overline{\vartheta}^h_{ms} = \overline{\vartheta}_{ms} \cap \overline{\Omega}^h_m$, $s = 1, \ldots, S_m - 1$, and problems on the horizontal interfacial subdomains are computed. Finally, we piece together the solutions on the subdomains.

1. Initialisation: On $\overline{\Omega}^h_m$, choose an initial mesh function $v^{(n,0)}_m(p)$, $p \in \overline{\Omega}^h_m$, satisfying the boundary conditions in (3.10).

2. On the box-subdomains $\overline{\Omega}^h_{ms}$, $s = 1, \ldots, S_m$, compute mesh functions $z^{(n,k)}_{ms}(p)$ (here the index k stands for a number of inner iterative steps) satisfying the difference problems

$$(\mathcal{L} + c^*) z^{(n,k)}_{ms}(p) = -\mathcal{R}(p, v^{(n,k-1)}_m), \quad p \in \Omega^h_{ms}, \tag{3.12}$$

$$z^{(n,k)}_{ms}(p) = 0, \quad p \in \partial\Omega^h_{ms},$$

$$v^{(n,k)}_{ms}(p) = v^{(n,k-1)}_m(p) + z^{(n,k)}_{ms}(p), \quad p \in \overline{\Omega}^h_{ms},$$

where $\mathcal{R}(p, v^{(n,k-1)}_m)$ is the residual of the difference scheme (3.10) on $v^{(n,k-1)}_m$,

that is,

$$\mathcal{R}(p, v_m^{(n,k-1)}) = \mathcal{L}v_m^{(n,k-1)}(p) + f(p, v_m^{(n,k-1)}).$$

3. On the horizontal interfacial subdomains $\overline{\vartheta}_{ms}^h$, $s = 1, \ldots, S_m - 1$, compute the difference problems

$$(\mathcal{L} + c^*)\widetilde{z}_{ms}^{(n,k)}(p) = -\mathcal{R}(p, v_m^{(n,k-1)}), \quad p \in \vartheta_{ms}^h, \tag{3.13}$$

$$\widetilde{v}_{ms}^{(n,k)}(p) = v_m^{(n,k-1)}(p) + \widetilde{z}_{ms}^{(n,k)}(p), \quad p \in \overline{\vartheta}_{ms}^h,$$

$$\widetilde{z}_{ms}^{(n,k)}(p) = \begin{cases} 0, & p \in \rho_{ms}^{hl} \cap \rho_{ms}^{hr}, \\ z_{ms}^{(n,k)}, & p \in \rho_{ms}^{hb}, \\ z_{m,s+1}^{(n,k)}, & p \in \rho_{ms}^{ht}. \end{cases}$$

4. Compute the mesh function $v_m^{(n,k)}(p)$, $p \in \overline{\Omega}_m^h$ by piecing together the solutions on the subdomains

$$v_m^{(n,k)}(p) = \begin{cases} v_{ms}^{(n,k)}(p), & p \in \overline{\Omega}_{ms}^h \setminus (\overline{\vartheta}_{m,s-1}^h \cup \overline{\vartheta}_{ms}^h), \\ & s = 2, \ldots, S_m - 1; \\ v_{m,1}^{(n,k)}(p), & p \in \overline{\Omega}_{m1}^h \setminus \overline{\vartheta}_{m1}^h; \\ v_{m,S_m}^{(n,k)}(p), & p \in \overline{\Omega}_{m,S_m}^h \setminus \overline{\vartheta}_{m,S_m-1}^h; \\ \widetilde{v}_{ms}^{(n,k)}(p), & p \in \overline{\vartheta}_{ms}^h, \ s = 1, \ldots, S_m - 1. \end{cases} \tag{3.14}$$

5. Stopping criterion: if

$$\max_{p \in \overline{\Omega}_m^h} |\mathcal{R}(p, v_m^{(n,k)})| \le \delta, \tag{3.15}$$

where δ is the required accuracy, then stop; otherwise go to Step 2.

Algorithm (3.12)–(3.14) can be carried out by parallel processing. Each of the box-subdomain problems on $\overline{\Omega}_{ms}$, $s = 1, \ldots, S_m$, in Step 2 can be solved on their own processor in parallel. In Step 3, each of the interfacial subdomain problems on $\overline{\vartheta}_{ms}^h$, $s = 1, \ldots, S_m - 1$, can be solved on their own processors in parallel.

Theorem 6. *Let $\overline{v}_m^{(n,0)}$, $\underline{v}_m^{(n,0)}$ be upper and lower solutions of (3.10). Then the sequences $\{\overline{v}_m^{(n,k)}(p)\}$ and $\{\underline{v}_m^{(n,k)}(p)\}$ generated by (3.12)–(3.14) converge monotonically from above*

and below, respectively, to $v_m^{(n)}$,

$$\underline{v}_m^{(n,k-1)}(p) \le \underline{v}_m^{(n,k)}(p) \le v_m^{(n)}(p) \le \overline{v}_m^{(n,k)}(p) \le \overline{v}_m^{(n,k-1)}(p), \quad p \in \overline{\Omega}_m^h,$$

where $v_m^{(n)}$ *is the solution of (3.10).*

Proof. We consider only the case of upper solutions. The case for lower solutions may be proved in a similar way.

Let $\overline{v}_m^{(n,0)}(p)$, $p \in \overline{\Omega}_m^h$ be an upper solution satisfying the boundary conditions in (3.10). From (3.12) we get

$$\begin{aligned}(\mathcal{L}+c^*)z_{ms}^{(n,1)}(p) &= -\mathcal{R}(p,\overline{v}_m^{(n,0)}) \le 0, \quad p \in \Omega_{ms}^h,\\ z_{ms}^{(n,1)}(p) &= 0, \quad p \in \partial\Omega_{ms}^h, \quad s = 1,\dots,S_m.\end{aligned}$$

By the maximum principle in Lemma 1, we conclude that

$$z_{ms}^{(n,1)}(p) \le 0, \quad p \in \overline{\Omega}_{ms}^h, \quad s = 1,\dots,S. \tag{3.16}$$

We now prove that $v_{ms}^{(n,1)}(p)$ is an upper solution on $\overline{\Omega}_{ms}^h$. Using the mean value theorem, from (3.12) we have

$$\begin{aligned}\mathcal{L}v_{ms}^{(n,1)}(p) + f(p,v_{ms}^{(n,1)}) &= \mathcal{L}z_{ms}^{(n,1)}(p) \pm c^* z_{ms}^{(n,1)}(p) + \mathcal{L}\overline{v}_m^{(n,0)}(p) + f(p,v_{ms}^{(n,1)}) \qquad (3.17)\\ &= -\mathcal{R}(p,\overline{v}_m^{(n,0)}) - c^* z_{ms}^{(n,1)}(p) + \mathcal{L}\overline{v}_m^{(n,0)}(p)\\ &\quad + f(p,v_{ms}^{(n,1)})\\ &= -\mathcal{L}\overline{v}_m^{(n,0)}(p) - f(p,\overline{v}^{(n,0)}) - c^* z_{ms}^{(n,1)}(p)\\ &\quad + \mathcal{L}\overline{v}_m^{(n,0)}(p) + f(p,v_{ms}^{(n,1)})\\ &= -(c^* - f_v^{(n,1)})z_{ms}^{(n,1)}(p), \quad p \in \Omega_{ms}^h,\end{aligned}$$

where $f_v^{(n,1)}(p) = f_v[p,\overline{v}^{(n,0)}(p) + \Theta_{ms}^{(n,1)}(p)z_{ms}^{(n,1)}(p)]$, $0 \le \Theta_{ms}^{(n,1)}(p) \le 1$. From (1.20), (3.12) and (3.16), we conclude that

$$\begin{aligned}\mathcal{L}v_{ms}^{(n,1)}(p) + f(p,v_{ms}^{(n,1)}) &\ge 0, \quad p \in \Omega_{ms}^h,\\ v_{ms}^{(n,1)}(p) &= \overline{v}^{(n,0)}(p), \quad p \in \partial\Omega_{ms}^h.\end{aligned} \tag{3.18}$$

Thus, it follows that $v_{ms}^{(n,1)}(p)$, $p \in \overline{\Omega}_{ms}^h$, $s = 1,\dots,S_m$, is an upper solution. From (3.13) and (3.16), we have

$$\begin{aligned}(\mathcal{L}+c^*)\widetilde{z}_{ms}^{(n,1)}(p) &= -\mathcal{R}(p,\overline{v}_m^{(n,0)}) \le 0, \qquad p \in \vartheta_{ms}^h,\\ \widetilde{z}_{ms}^{(n,1)}(p) &\le 0, \qquad p \in \partial\vartheta_{ms}^h.\end{aligned}$$

By the maximum principle from Lemma 1, we conclude that

$$\widetilde{z}_{ms}^{(n,1)}(p) \le 0, \qquad p \in \overline{\vartheta}_{ms}^{h}, \qquad s = 1, \ldots, S_m - 1. \tag{3.19}$$

We now prove that $\widetilde{v}_{ms}^{(n,1)}(p)$ is an upper solution on $\overline{\vartheta}_{ms}^{h}$. By the mean value theorem, from (3.13) we have

$$\begin{aligned}
\mathcal{L}\widetilde{v}_{ms}^{(n,1)}(p) + f(p, \widetilde{v}_{ms}^{(n,1)}) &= \mathcal{L}\widetilde{z}_{ms}^{(n,1)}(p) \pm c^* \widetilde{z}_{ms}^{(n,1)}(p) + \mathcal{L}\overline{v}_{m}^{(n,0)}(p) + f(p, \widetilde{v}_{ms}^{(n,1)}) \\
&= -\mathcal{R}(p, \overline{v}_{m}^{(n,0)}) - c^* \widetilde{z}_{ms}^{(n,1)}(p) + \mathcal{L}\overline{v}_{m}^{(n,0)}(p) \\
&\quad + f(p, \widetilde{v}_{ms}^{(n,1)}) \\
&= -\mathcal{L}\overline{v}_{m}^{(n,0)}(p) - f(p, \overline{v}^{(n,0)}) - c^* \widetilde{z}_{ms}^{(n,1)}(p) \\
&\quad + \mathcal{L}\overline{v}_{m}^{(n,0)}(p) + f(p, \widetilde{v}_{ms}^{(n,1)}) \\
&= -(c^* - \widetilde{f}_{v}^{(n,1)}) \widetilde{z}_{ms}^{(n,1)}(p), \quad p \in \vartheta_{ms}^{h},
\end{aligned}$$

where $\widetilde{f}_{v}^{(n,1)}(p) = f_v[p, \overline{v}^{(n,0)}(p) + \widetilde{\Theta}_{ms}^{(n,1)}(p) \widetilde{z}_{ms}^{(n,1)}(p)]$, $0 \le \widetilde{\Theta}_{ms}^{(n,1)}(p) \le 1$. From (1.20), (3.13) and (3.16), we conclude that

$$\mathcal{L}\widetilde{v}_{ms}^{(n,1)} + f(p, \widetilde{v}_{ms}^{(n,1)}) \ge 0, \quad p \in \vartheta_{ms}^{h}, \tag{3.20}$$

$$\widetilde{v}_{ms}^{(n,1)}(p) = \begin{cases} v_{m}^{(n,0)}(p), & p \in \rho_{ms}^{hl} \cup \rho_{ms}^{hr} \\ v_{ms}^{(n,1)}(p), & p \in \rho_{ms}^{hb} \\ v_{m,s+1}^{(n,1)}(p), & p \in \rho_{ms}^{ht}. \end{cases}$$

We now prove that $v_{m}^{(n,1)}(p)$ defined by (3.14) is an upper solution on $\overline{\Omega}_{m}^{h}$. From (3.14), (3.18) and (3.20), we have

$$\mathcal{L}v_{m}^{(n,1)}(p) + f(p, v_{m}^{(n,1)}) \ge 0, \quad p \in \Omega_{m}^{h} \setminus \sum_{s=1}^{S_m - 1} (\rho_{ms}^{hb} \cup \rho_{ms}^{ht}).$$

This is due to the construction of $v_{m}^{(n,1)}(p)$ from (3.14): $v_{m}^{(n,1)}(p)$ is set equal to the solution of the difference problem (3.12) on $p \in \overline{\Omega}_{ms}^{h} \setminus \left(\overline{\vartheta}_{m,s-1}^{h} \cup \overline{\vartheta}_{ms}^{h} \right)$ and equal to the solution of the difference problem (3.13) on $p \in \overline{\vartheta}_{ms}^{h}$. Therefore we only need to check that $v_{m}^{(n,1)}(p)$ is an upper solution on the interfacial boundaries ρ_{ms}^{hb} and ρ_{ms}^{ht}. Let $w_{ms}^{(n,1)}(p) = v_{ms}^{(n,1)}(p) - \widetilde{v}_{ms}^{(n,1)}(p) = z_{ms}^{(n,1)}(p) - \widetilde{z}_{ms}^{(n,1)}(p)$ and $\xi_{ms}^{hb} = \Omega_{ms}^{h} \cap \vartheta_{ms}^{h}$, then we have

$$\begin{aligned}
\mathcal{L}w_{ms}^{(n,1)}(p) + c^* w_{ms}^{(n,1)}(p) &= \mathcal{L}z_{ms}^{(n,1)}(p) + c^* z_{ms}^{(n,1)}(p) \\
&\quad - (\mathcal{L}\widetilde{z}_{ms}^{(n,1)}(p) + c^* \widetilde{z}_{ms}^{(n,1)}(p)) \\
&= -\mathcal{R}(p, v_{m}^{(n,0)}) + \mathcal{R}(p, v_{m}^{(n,0)}) = 0, \quad p \in \xi_{ms}^{hb},
\end{aligned} \tag{3.21}$$

$$w_{ms}^{(n,1)}(p) = 0, \quad p \in \partial\xi_{ms}^{hb} \setminus \gamma_{ms}^{ht}.$$

By (3.19), we have

$$\begin{aligned} w_{ms}^{(n,1)}(p) &= v_{ms}^{(n,1)}(p) - \widetilde{v}_{ms}^{(n,1)}(p) \\ &= v_{m}^{(n,0)}(p) - \widetilde{v}_{ms}^{(n,1)}(p) \\ &= v_{m}^{(n,0)}(p) - (v_{m}^{(n,0)}(p) + \widetilde{z}_{ms}^{(n,1)}(p)) \geq 0, \quad p \in \gamma_{ms}^{ht}. \end{aligned}$$

From here and (3.21), by the maximum principle from Lemma 1, we have

$$w_{ms}^{(n,1)}(p) = v_{ms}^{(n,1)}(p) - \widetilde{v}_{ms}^{(n,1)}(p) \geq 0, \quad p \in \overline{\xi}_{ms}^{hb}.$$

From here and (3.14), we have for $p \in \rho_{ms}^{hb}$

$$\begin{aligned} \mathcal{L}v_{m}^{(n,1)}(p) + f(p, v_{m}^{(n,1)}) &= d(p)v_{m}^{(n,1)}(p) - \sum_{p' \in \sigma'(p)} e(p,p')v_{m}^{(n,1)}(p') \\ &\quad + f(p, v_{m}^{(n,1)}) \\ &= d(p)v_{ms}^{(n,1)}(p) - \sum_{p' \in \sigma'_L(p)} e(p,p')v_{ms}^{(n,1)}(p') \\ &\quad - \sum_{p' \in \sigma'_U(p)} e(p,p')\widetilde{v}_{ms}^{(n,1)}(p') + f(p, v_{ms}^{(n,1)}) \\ &\geq d(p)v_{ms}^{(n,1)}(p) - \sum_{p' \in \sigma'(p)} e(p,p')v_{ms}^{(n,1)}(p') \\ &\quad + f(p, v_{ms}^{(n,1)}) \\ &\geq \mathcal{L}v_{ms}^{(n,1)}(p) + f(p, v_{ms}^{(n,1)}) \geq 0, \end{aligned}$$

where $\sigma'_U(p)$ is the set of stencil points corresponding to a strictly upper triangular part of $\sigma'(p)$, and $\sigma'_L(p)$ is the set of stencil points corresponding to a strictly lower triangular part of $\sigma'(p)$, such that $\sigma'(p) = \sigma'_U(p) \cup \sigma'_L(p)$.

Let $\widehat{w}_{ms}^{(n,1)}(p) = v_{m,s+1}^{(n,1)}(p) - \widetilde{v}_{ms}^{(n,1)}(p)$ and $\xi_{ms}^{ht} = \Omega_{m,s+1}^{h} \cap \vartheta_{ms}^{h}$, $s = 1, \ldots, S_m - 1$. In a similar way as in (3.21), we can show that

$$\mathcal{L}\widehat{w}_{ms}^{(n,1)}(p) + c^*\widehat{w}_{ms}^{(n,1)}(p) = 0, \quad p \in \xi_{ms}^{ht},$$

$$\widehat{w}_{ms}^{(n,1)}(p) = 0, \quad p \in \partial\xi_{ms}^{ht} \setminus \gamma_{ms}^{ht},$$

$$\widehat{w}_{ms}^{(n,1)}(p) = v_{m,s+1}^{(n,1)}(p) - \widetilde{v}_{ms}^{(n,1)}(p) = v_{m}^{(n,0)}(p) - \widetilde{v}_{ms}^{(n,1)}(p) \geq 0, \quad p \in \gamma_{ms}^{ht}.$$

By the maximum principle from Lemma 1, we have

$$\widehat{w}_{ms}^{(n,1)}(p) = v_{m,s+1}^{(n,1)}(p) - \widetilde{v}_{ms}^{(n,1)}(p) \geq 0, \quad p \in \overline{\xi}_{ms}^{ht}.$$

From here, we have for $p \in \rho_{ms}^{ht}$,

$$\begin{aligned}
\mathcal{L}v_m^{(n,1)}(p) + f(p, v_m^{(n,1)}) &= d(p)v_m^{(n,1)}(p) - \sum_{p' \in \sigma'(p)} e(p,p')v_m^{(n,1)}(p') \\
&\quad + f(p, v_m^{(n,1)}) \\
&= d(p)v_{m,s+1}^{(n,1)}(p) - \sum_{p' \in \sigma'_L(p)} e(p,p')\widetilde{v}_{m,s}^{(n,1)}(p') \\
&\quad - \sum_{p' \in \sigma'_U(p)} e(p,p')v_{m,s+1}^{(n,1)}(p') + f(p, v_{m,s+1}^{(n,1)}) \\
&\geq d(p)v_{m,s+1}^{(n,1)}(p) - \sum_{p' \in \sigma'(p)} e(p,p')v_{m,s+1}^{(n,1)}(p') \\
&\quad + f(p, v_{m,s+1}^{(n,1)}) \\
&\geq \mathcal{L}v_{m,s+1}^{(n,1)} + f(p, v_{m,s+1}^{(n,1)}) \geq 0.
\end{aligned}$$

From here, we conclude that

$$\mathcal{L}v_m^{(n,1)} + f(p, v_m^{(n,1)}) \geq 0, \quad p \in \Omega_m^h,$$

$$v_m^{(n,1)}(p) = v_m^{(n,0)}(p), \quad p \in \partial\Omega_m^h.$$

Thus, $v_m^{(n,1)}$ is an upper solution on $\overline{\Omega}_m^h$. By induction on k, we are able to conclude that $\{v_m^{(n,k)}\}$ is a monotonically decreasing sequence of upper solutions bounded below by $\underline{v}_m$ (1.19), where $\underline{v}_m$ is a lower solution on $\overline{\Omega}_m^h$. Let

$$v_m^{(n)}(p) = \lim_{k \to \infty} v_m^{(n,k)}(p) \qquad p \in \overline{\Omega}_m^h.$$

We now show that $v_m^{(n)}(p)$ is the solution to (3.10). By (3.12)–(3.14),

$$v_m^{(n)}(p) = v_m^{(n,k)}(p) = v_m^{(n,0)}(p), \qquad p \in \partial\Omega_m^h.$$

Hence, $v_m^{(n)}(p)$ satisfies the boundary conditions in (3.10). From (3.14), we have

$$\begin{aligned}
&\lim_{k \to \infty} z_{ms}^{(n,k)}(p) = \lim_{k \to \infty} [v_m^{(n,k)}(p) - v_{ms}^{(n,k-1)}(p)] = 0, \quad p \in \overline{\Omega_{m1}^h \setminus \vartheta_{m1}^h}, \\
&p \in \overline{\Omega_{ms}^h \setminus (\vartheta_{m,s-1}^h \cup \vartheta_{ms}^h)}, \quad s = 2, \ldots, S_m - 1, \quad p \in \overline{\Omega_{m,S_m}^h \setminus \vartheta_{m,S_m-1}^h}, \\
&\lim_{k \to \infty} \widetilde{z}_{ms}^{(n,k)}(p) = \lim_{k \to \infty} [v_m^{(n,k)}(p) - \widetilde{v}_{ms}^{(n,k-1)}(p)] = 0, \\
&p \in \overline{\vartheta}_{ms}^h, \quad s = 1, \ldots, S_m - 1.
\end{aligned}$$

From here, (3.12) and (3.13), we obtain

$$\begin{aligned}
&\mathcal{R}(p, v_m^{(n)}) = \lim_{k\to\infty} \mathcal{R}(p, v_m^{(n,k-1)}) = -\lim_{k\to\infty} (\mathcal{L} + c^*) z_{ms}^{(n,k)}(p) = 0,\\
&p \in \Omega_{m1}^h \setminus \vartheta_{m1}^h, \quad p \in \Omega_{m,S_m}^h \setminus \vartheta_{m,S_m-1}^h,\\
&p \in \Omega_{ms}^h \setminus (\vartheta_{m,s-1}^h \cup \vartheta_{ms}^h), \quad s = 2, \ldots, S_m - 1,\\
&\mathcal{R}(p, v_m^{(n)}) = \lim_{k\to\infty} \mathcal{R}(p, v_m^{(n,k-1)}) = -\lim_{k\to\infty} (\mathcal{L} + c^*) \widetilde{z}_{ms}^{(n,k)}(p) = 0,\\
&p \in \vartheta_{m,s}^h, \quad s = 1, \ldots, S_m - 1.
\end{aligned}$$

Thus, $v_m^{(n)}$ is a solution of (3.10) on $\Omega_m^h \setminus \bigcup_{s=1}^{S_m-1}(\rho_{ms}^{hb} \cup \rho_{ms}^{ht})$. We now verify that $v_m^{(n)}$ satisfies (3.10) on the interfacial boundaries ρ_{ms}^{hb}, ρ_{ms}^{ht}, $s = 1, \ldots, S_m - 1$. Since

$$v_{ms}^{(n,k)}(p) - \widetilde{v}_{ms}^{(n,k)}(p) = v_m^{(n,k-1)}(p) - v_m^{(n,k)}(p), \qquad p \in \gamma_{ms}^{ht}, \quad s = 1, \ldots, S_m - 1,$$

we conclude that

$$\lim_{k\to\infty} v_{ms}^{(n,k)}(p) - \widetilde{v}_{ms}^{(n,k)}(p) = 0, \quad p \in \gamma_{ms}^{ht}, \quad s = 1, \ldots, S_m - 1.$$

From (3.21), we have

$$\mathcal{L} w_{ms}^{(n,k)}(p) + c^* w_{ms}^{(n,k)}(p) = 0, \quad p \in \xi_{ms}^{hb},$$

$$w_{ms}^{(n,k)}(p) = 0, \quad p \in \partial\xi_{ms}^{hb} \setminus \gamma_{ms}^{ht},$$

$$w_{ms}^{(n,k)}(p) = v_{ms}^{(n,k)}(p) - \widetilde{v}_{ms}^{(n,k)}(p), \quad p \in \gamma_{ms}^{ht}.$$

Taking the limit as $k \to \infty$ and applying the maximum principle from Lemma 1, we conclude that

$$\lim_{k\to\infty} w_{ms}^{(n,k)}(p) = \lim_{k\to\infty} v_{ms}^{(n,k)}(p) - \widetilde{v}_{ms}^{(n,k)}(p) = 0, \quad p \in \overline{\xi}_{ms}^{hb}.$$

From here and (3.14), we have

$$\lim_{k\to\infty} v_{ms}^{(n,k)}(p) = \lim_{k\to\infty} \widetilde{v}_{ms}^{(n,k)}(p) = v_m^{(n)}(p), \qquad p \in \overline{\xi}_{ms}^{hb}.$$

From here and (3.17), it follows that for $s = 1, \ldots, S_m - 1$,

$$\begin{aligned}
\mathcal{L} v_m^{(n)}(p) + f(p, v_m^{(n)}) &= \lim_{k\to\infty} (\mathcal{L} v_m^{(n,k)}(p) + f(p, v_m^{(n,k)}))\\
&= \lim_{k\to\infty} (\mathcal{L} v_{ms}^{(n,k)}(p) + f(p, v_{ms}^{(n,k)}))\\
&= \lim_{k\to\infty} (-(c^* - f_v^{(n,k)}(p)) z_{ms}^{(n,k)}(p)) = 0, \quad p \in \rho_{ms}^{hb}.
\end{aligned}$$

Thus, $v_m^{(n)}$ is a solution of (3.10) on $p \in \rho_{ms}^{hb}$, $s = 1, \ldots, S_m - 1$. In a similar way, we can prove that for $s = 1, \ldots, S_m - 1$,

$$\begin{aligned}
\mathcal{L}v_m^{(n)}(p) + f(p, v_m^{(n)}) &= \lim_{k\to\infty} (\mathcal{L}v_m^{(n,k)}(p) + f(p, v_m^{(n,k)})) \\
&= \lim_{k\to\infty} (\mathcal{L}v_{m,s+1}^{(n,k)}(p) + f(p, v_{m,s+1}^{(n,k)})) \\
&= \lim_{k\to\infty} (-(c^* - f_v^{(n,k)}(p))z_{m,s+1}^{(n,k)}(p)) = 0, \quad p \in \rho_{ms}^{ht}.
\end{aligned}$$

Thus, $v_m^{(n)}$ is a solution to (3.10) for $p \in \overline{\Omega}_m^h$, and we prove the theorem. □

3.3.3 Numerical stability of the outer and inner iterates

The solution generated by the two-level domain decomposition algorithm (3.10),(3.11), (3.12)–(3.14) will be denoted by $\widehat{v}^{(n)}(p)$, $p \in \overline{\Omega}^h$.

Theorem 7. *The two-level domain decomposition algorithm (3.10),(3.11), (3.12)–(3.14) with the stopping criterion (3.15) is numerically stable:*

$$\|\widehat{v}^{(n)} - v^{(n)}\|_{\overline{\Omega}^h} \le \frac{1}{c_*}\delta, \quad n \ge 1, \tag{3.22}$$

where $v^{(n)}(p)$ is the solution generated by the outer iterates (3.10),(3.11).

Proof. Introduce the notation

$$w_m^{(n)}(p) = \widehat{v}_m^{(n)}(p) - v_m^{(n)}(p), \quad p \in \overline{\Omega}_m^h, \quad m = 1, \ldots, M,$$

$$w^{(n)}(p) = \widehat{v}^{(n)}(p) - v^{(n)}(p), \quad p \in \overline{\Omega}^h,$$

where $\widehat{v}^{(n)}(p)$ is defined by (3.11) with $\widehat{v}_m^{(n)}(p)$ instead of $v_m^{(n)}(p)$. Since $\widehat{v}^{(0)} = v^{(0)}$, from (3.10), (3.15) and applying the mean-value theorem, we conclude that

$$\mathcal{L}w_1^{(1)}(p) + f_u w_1^{(1)}(p) = \delta_1^{(1)}(p), \quad p \in \Omega_1^h, \quad \|\delta_1^{(1)}\|_{\Omega_1^h} \le \delta,$$

$$w_1^{(1)}(p) = 0, \quad p \in \partial\Omega_1^h,$$

where f_u is calculated at an intermediate point between $\widehat{v}_1^{(1)}(p)$ and $v_1^{(1)}(p)$. From (1.20), by (2.17), we have $\|w_1^{(1)}\|_{\overline{\Omega}_1^h} \le (1/c_*)\delta$. Similarly, for $w_2^{(1)}$, we can get the difference problem

$$\mathcal{L}w_2^{(1)}(p) + f_u w_2^{(1)}(p) = \delta_2^{(1)}(p), \quad p \in \Omega_2^h, \quad \|\delta_2^{(1)}\|_{\Omega_1^h} \le \delta,$$

$$w_2^{(1)}(p) = \begin{cases} 0, & p \in \gamma_2^{h0} \cup \gamma_2^{he}; \\ w_1^{(1)}(p), & p \in \gamma_2^{hb}, \end{cases}$$

From (1.20) and the estimate on $w_1^{(1)}$, by (2.17), we have $\|w_2^{(1)}\|_{\overline{\Omega}_2^h} \le (1/c_*)\delta$. Now by induction on m, we can conclude that

$$\|w_m^{(1)}\|_{\overline{\Omega}_m^h} \le \frac{1}{c_*}\delta, \quad m = 1, \ldots, M,$$

and, hence, we prove (3.22) for $n = 1$.

In the same way as before, we can obtain the difference problems for $w_m^{(2)}$, $m = 1, \ldots, M$,

$$\mathcal{L}w_m^{(2)}(p) + f_u w_m^{(2)}(p) = \delta_m^{(2)}(p), \quad p \in \Omega_m^h, \quad \|\delta_m^{(2)}\|_{\Omega_m^h} \le \delta,$$

$$w_m^{(2)}(p) = \begin{cases} 0, & p \in \gamma_m^{h0}; \\ w_{m-1}^{(2)}(p), & p \in \gamma_m^{hb}; \\ w^{(1)}(p), & p \in \gamma_m^{he}. \end{cases}$$

Since $w_1^{(2)}(p) = 0$, $p \in \gamma_1^{hb}$, from (1.20) and (3.22) with $n = 1$, by (2.17), we have $\|w_1^{(2)}\|_{\overline{\Omega}_1^h} \le (1/c_*)\delta$. Now by induction on m, we can conclude that

$$\|w_m^{(2)}\|_{\overline{\Omega}_m^h} \le \frac{1}{c_*}\delta, \quad m = 1, \ldots, M,$$

and, hence, we prove (3.22) for $n = 2$. Finally, by induction on n, we can prove the required estimate on $w^{(n)} = \widehat{v}^{(n)} - v^{(n)}$. □

3.3.4 Numerical experiments

We apply the two-level monotone domain decomposition algorithm (3.10), (3.11), (3.12)–(3.14) to two test problems: the convection-diffusion problem with parabolic layers (1.9) and the anisotropic convection-diffusion problem (1.12). For these test problems, the required accuracy for the outer and inner iterations are $\Delta = \delta = 10^{-6}$.

The number of mesh points in the x- and y-directions are set equal to N, and the width of the interfacial subdomains is held fixed at $N/(2S)$, where $S = S_m$, $m = 1, \ldots, M$.

To solve the linear difference problems within the inner iterates (3.12), (3.13), GMRES with restarts is used with a diagonal preconditioner. Within GMRES, the required accuracy is 10^{-6}, the maximum number of iterations is 50 and a maximum of 20 restarts.

For the following experiments, we focus on balanced domain decompositions where M is even and $M/2$ vertical strips are placed within the boundary layer. Balanced domain decompositions are more suited to parallel implementation than unbalanced domain decompositions.

The domain is decomposed into M overlapping subdomains as described in Section 3.2.1. The overlap between the vertical strips is chosen so that for the two vertical strips either side of the boundary layer, the overlap occurs outside of the boundary layer, as described in Section 3.2.1.

At each step of outer iteration, we require an initial function. In the following numerical experiments, we use the method given in (1.22) to construct initial upper or lower solutions. By using this method, we are guaranteed monotonic convergence within each of the vertical strips by Theorem 6. The method of obtaining initial upper or lower solutions given in (1.22) requires the use of an arbitrary function which satisfies the boundary conditions. Instead of using an arbitrary function, the solution from the pervious outer iterate is used.

For the tables presented below, results for the minimal and maximal size of the overlaps appear above and below the line, respectively.

Convection-diffusion problem with parabolic boundary layers

Consider the test problem (3.6). This test problem is the convection-diffusion problem with parabolic layers (1.9). It is characterised by an elliptic boundary layer close to $x = 1$ and parabolic boundary layers close to $y = 0$ and $y = 1$ (see Section 1.4.2 for details). To solve this problem numerically, we use the piecewise uniform Shishkin mesh (1.11). The construction of this mesh is described in Section 1.4.2. We require the constants c_* and c^*. From (3.7) we have

$$c_* = \exp(-1), \quad c^* = 1.$$

Table 3.4 shows the outer iteration counts over varying numbers of vertical strips and for different values of ε and N. Through numerical experiments, we observe that the number of horizontal strips does not affect the outer iteration counts. The outer iteration counts in Table 3.4 show that for the larger overlap size the outer iteration counts are less. The number of vertical strips increases so does the number of outer iterations needed for the algorithm to converge. Table 3.4 also shows that the two-level domain decomposition algorithm uniformly converges in its outer iteration counts with respect to ε.

Table 3.5 displays the serial execution times over varying numbers of vertical and horizontal strips, for different values of ε and N. From these results, we observe that the execution times are smaller for the maximal overlap size compared to that of the

ε	10^{-2}			10^{-3}			10^{-4}		
$N\backslash M$	2	4	8	2	4	8	2	4	8
	Iteration counts								
2^5	$\frac{6}{3}$	$\frac{10}{5}$	$\frac{13}{13}$	$\frac{4}{3}$	$\frac{10}{5}$	$\frac{13}{13}$	$\frac{3}{3}$	$\frac{10}{5}$	$\frac{13}{13}$
2^6	$\frac{7}{3}$	$\frac{13}{4}$	$\frac{17}{7}$	$\frac{4}{3}$	$\frac{13}{4}$	$\frac{17}{7}$	$\frac{3}{3}$	$\frac{13}{4}$	$\frac{17}{7}$
2^7	$\frac{10}{3}$	$\frac{19}{4}$	$\frac{25}{5}$	$\frac{4}{3}$	$\frac{19}{4}$	$\frac{25}{5}$	$\frac{3}{3}$	$\frac{19}{4}$	$\frac{25}{6}$

Table 3.4: Outer iteration counts using the minimum and maximum overlap size, above and below the line, respectively, for the test problem (3.6).

minimal overlap size. In Table 3.5, the bold results are the minimal execution times for the maximal and minimal overlap size in each column for $N = 128$. These show that the absolute minimum execution time, for both the minimal and maximal overlap size, for all ε, occurs when the number of vertical strips is 8 and the number of horizontal strips is 1. By using these execution times, the acceleration (execution time of the undecomposed algorithm / minimum execution time of the two-level domain decomposition algorithm) of the two-level domain decomposition algorithm is 9, 25, 75, for $\varepsilon = 10^{-2}$, 10^{-3}, 10^{-4}, respectively.

Table 3.5 shows that for a fixed number of vertical strips the ideal number of horizontal strips is the same for all values of ε. For example, for $M = 2$, 4, 8, the minimum execution times occurs for both the minimal and maximal overlap size when $S = 8$, 4, 1, respectively.

Anisotropic convection-diffusion problem

For the following numerical experiments a Linux cluster from the New Zealand Supercomputing Centre (NZSC) is used. The NZSC supercomputer consists of about 3000 processors. For these experiments a chassis of 28 processors is used. The chassis consists of 14 IBM H20 Dual Blade servers each of which contain 2 x 2.8Ghz Intel Xeon processors, 6GB of RAM and 40GB of local disk. Nodes are interconnected with gigabit ethernet. The code used for the experiments below is programmed in C using MPI (Message Passing Interface) to specify the communication between processors.

Consider the test problem (3.8). This test problem is the anisotropic convection-

	ε	10^{-2}				10^{-3}				10^{-4}			
N	$S\backslash M$	1	2	4	8	1	2	4	8	1	2	4	8
		Execution times (seconds)											
2^5	1	0.4	$\frac{0.4}{0.2}$	$\frac{0.4}{0.2}$	$\frac{0.4}{0.4}$	0.3	$\frac{0.2}{0.2}$	$\frac{0.3}{0.1}$	$\frac{0.3}{0.3}$	0.3	$\frac{0.2}{0.1}$	$\frac{0.2}{0.1}$	$\frac{0.2}{0.2}$
	2	0.4	$\frac{0.6}{0.2}$	$\frac{0.6}{0.3}$	$\frac{0.6}{0.6}$	0.4	$\frac{0.3}{0.2}$	$\frac{0.4}{0.2}$	$\frac{0.4}{0.4}$	0.3	$\frac{0.2}{0.2}$	$\frac{0.3}{0.1}$	$\frac{0.3}{0.3}$
	4	0.4	$\frac{0.6}{0.3}$	$\frac{0.7}{0.4}$	$\frac{0.9}{0.9}$	0.4	$\frac{0.4}{0.2}$	$\frac{0.5}{0.2}$	$\frac{0.6}{0.6}$	0.4	$\frac{0.3}{0.2}$	$\frac{0.5}{0.2}$	$\frac{0.5}{0.5}$
	8	0.4	$\frac{0.7}{0.4}$	$\frac{1.1}{0.7}$	$\frac{1.9}{1.9}$	0.3	$\frac{0.4}{0.2}$	$\frac{0.7}{0.3}$	$\frac{1.0}{1.0}$	0.3	$\frac{0.3}{0.2}$	$\frac{0.6}{0.2}$	$\frac{0.9}{0.9}$
2^6	1	13.8	$\frac{6.2}{3.4}$	$\frac{4.4}{1.9}$	$\frac{3.5}{2.2}$	11.2	$\frac{3.2}{3.3}$	$\frac{2.7}{1.1}$	$\frac{2.2}{1.0}$	16.5	$\frac{2.9}{2.7}$	$\frac{2.6}{0.8}$	$\frac{1.9}{0.7}$
	2	8.5	$\frac{7.9}{3.2}$	$\frac{6.1}{2.5}$	$\frac{5.4}{3.1}$	9.9	$\frac{4.2}{2.7}$	$\frac{4.1}{1.5}$	$\frac{3.3}{1.5}$	12.8	$\frac{3.5}{2.5}$	$\frac{3.4}{1.0}$	$\frac{2.8}{0.9}$
	4	5.9	$\frac{7.5}{3.1}$	$\frac{6.3}{2.5}$	$\frac{6.4}{3.5}$	10.2	$\frac{4.5}{2.8}$	$\frac{5.0}{1.7}$	$\frac{4.6}{1.9}$	13.1	$\frac{3.7}{2.7}$	$\frac{4.2}{1.2}$	$\frac{4.0}{1.3}$
	8	4.5	$\frac{7.7}{3.9}$	$\frac{8.1}{3.5}$	$\frac{10.6}{5.5}$	8.1	$\frac{4.6}{2.9}$	$\frac{5.7}{1.9}$	$\frac{6.4}{2.6}$	12.7	$\frac{4.0}{2.7}$	$\frac{5.1}{1.4}$	$\frac{6.1}{1.9}$
2^7	1	226.4	$\frac{619.4}{134.0}$	$\frac{114.6}{52.4}$	$\frac{\mathbf{49.8}}{24.5}$	289.2	$\frac{239.8}{150.0}$	$\frac{56.2}{36.8}$	$\frac{\mathbf{26.9}}{\mathbf{11.4}}$	627.2	$\frac{372.9}{163.7}$	$\frac{61.1}{33.4}$	$\frac{\mathbf{25.8}}{\mathbf{8.4}}$
	2	203.1	$\frac{415.1}{105.3}$	$\frac{\mathbf{94.2}}{37.8}$	$\frac{68.7}{\mathbf{24.1}}$	326.9	$\frac{176.0}{135.9}$	$\frac{\mathbf{49.6}}{27.9}$	$\frac{39.5}{12.4}$	662.8	$\frac{245.8}{160.8}$	$\frac{\mathbf{48.1}}{24.4}$	$\frac{37.1}{8.8}$
	4	106.8	$\frac{250.1}{66.1}$	$\frac{95.6}{\mathbf{32.1}}$	$\frac{73.9}{24.9}$	202.9	$\frac{\mathbf{131.7}}{97.1}$	$\frac{53.5}{\mathbf{23.8}}$	$\frac{49.7}{14.2}$	387.9	$\frac{170.0}{107.0}$	$\frac{51.3}{\mathbf{20.9}}$	$\frac{47.2}{10.3}$
	8	**63.8**	$\frac{\mathbf{186.7}}{\mathbf{62.3}}$	$\frac{106.6}{40.7}$	$\frac{96.2}{31.0}$	**140.0**	$\frac{132.5}{\mathbf{83.6}}$	$\frac{58.5}{24.0}$	$\frac{63.8}{16.3}$	**265.8**	$\frac{\mathbf{164.7}}{\mathbf{88.2}}$	$\frac{56.0}{21.4}$	$\frac{62.1}{12.4}$

Table 3.5: Execution times using the minimum and maximum overlap size, above and below the line, respectively, for the test problem (3.6).

diffusion problem (1.12). It is characterised by an elliptic boundary layer close to $x = 1$ (see Section 1.4.3 for details). To solve this problem numerically, we use the piecewise uniform Shishkin mesh (1.8) in the x-direction and uniform mesh in the y-direction . The construction of this mesh is described in Section 1.4.3. We require the constants c_* and c^*. From (3.9), we have

$$c_* = \exp(-1), \quad c^* = 1.$$

Table 3.6 shows the outer iteration counts over varying numbers of horizontal strips for different values of ε and N. Through numerical experiments, it is observed that the number of horizontal strips does not effect the outer iteration counts. From these outer iteration counts, it is apparent that for the larger overlap size the outer iteration counts are less. It can be seen that as the number of vertical strips increases so does the number

ε	10^{-2}			10^{-3}			10^{-4}		
$N\backslash M$	2	4	8	2	4	8	2	4	8
	Iteration counts								
2^5	$\frac{4}{3}$	$\frac{7}{4}$	$\frac{9}{9}$	$\frac{3}{3}$	$\frac{8}{5}$	$\frac{11}{11}$	$\frac{3}{3}$	$\frac{9}{5}$	$\frac{11}{11}$
2^6	$\frac{6}{3}$	$\frac{10}{4}$	$\frac{12}{5}$	$\frac{3}{3}$	$\frac{11}{4}$	$\frac{15}{6}$	$\frac{3}{3}$	$\frac{11}{4}$	$\frac{15}{7}$
2^7	$\frac{8}{3}$	$\frac{14}{3}$	$\frac{17}{4}$	$\frac{4}{3}$	$\frac{16}{3}$	$\frac{21}{5}$	$\frac{3}{3}$	$\frac{16}{3}$	$\frac{22}{5}$

Table 3.6: Outer iteration counts using the minimum and maximum overlap size, above and below the line, respectively, for the test problem (3.8).

of outer iterations needed for the algorithm to converge. We can also conclude from Table 3.6 that the two-level domain decomposition algorithm uniformly converges in its outer iteration counts with respect to ε.

	ε	10^{-2}				10^{-3}				10^{-4}			
N	$S\backslash M$	1	2	4	8	1	2	4	8	1	2	4	8
		Execution times (seconds)											
2^5	1	4.9	$\frac{5.8}{1.1}$	$\frac{2.0}{1.4}$	$\frac{1.6}{1.6}$	31.0	$\frac{8.6}{0.7}$	$\frac{6.7}{0.7}$	$\frac{5.7}{5.7}$	56.3	$\frac{10.3}{0.6}$	$\frac{10.9}{0.6}$	$\frac{8.8}{8.8}$
	2	1.3	$\frac{2.5}{1.5}$	$\frac{2.1}{1.6}$	$\frac{2.0}{2.0}$	6.7	$\frac{4.8}{1.3}$	$\frac{4.0}{1.2}$	$\frac{3.3}{3.3}$	11.5	$\frac{5.4}{1.1}$	$\frac{5.6}{1.1}$	$\frac{4.5}{4.5}$
	4	0.8	$\frac{1.2}{1.1}$	$\frac{1.4}{1.2}$	$\frac{1.8}{1.8}$	1.7	$\frac{1.8}{1.0}$	$\frac{1.8}{1.0}$	$\frac{2.1}{2.1}$	2.4	$\frac{2.2}{1.0}$	$\frac{2.7}{0.9}$	$\frac{2.6}{2.6}$
	8	0.7	$\frac{1.5}{1.2}$	$\frac{2.5}{1.9}$	$\frac{4.2}{4.2}$	1.1	$\frac{1.2}{1.3}$	$\frac{2.2}{1.6}$	$\frac{3.5}{3.5}$	1.2	$\frac{1.2}{1.3}$	$\frac{2.3}{1.6}$	$\frac{3.3}{3.4}$
2^6	1	57.5	$\frac{79.1}{12.2}$	$\frac{44.5}{12.0}$	$\frac{17.6}{16.5}$	628.7	$\frac{1436.9}{21.2}$	$\frac{80.7}{8.1}$	$\frac{72.4}{13.8}$	1336.2	$\frac{175.6}{8.2}$	$\frac{149.8}{7.0}$	$\frac{114.4}{8.3}$
	2	12.8	$\frac{34.1}{13.5}$	$\frac{29.8}{15.8}$	$\frac{22.2}{16.3}$	116.6	$\frac{125.6}{14.4}$	$\frac{65.1}{11.4}$	$\frac{52.5}{12.3}$	248.5	$\frac{245.4}{13.6}$	$\frac{104.7}{10.0}$	$\frac{85.3}{10.5}$
	4	7.8	$\frac{13.3}{9.4}$	$\frac{16.0}{10.6}$	$\frac{13.4}{10.0}$	27.0	$\frac{41.4}{13.1}$	$\frac{26.2}{8.0}$	$\frac{20.6}{7.8}$	56.1	$\frac{59.5}{10.1}$	$\frac{38.2}{7.0}$	$\frac{30.3}{7.6}$
	8	5.1	$\frac{9.4}{7.1}$	$\frac{12.1}{8.2}$	$\frac{12.7}{9.1}$	12.0	$\frac{14.7}{13.0}$	$\frac{12.2}{6.8}$	$\frac{12.6}{7.0}$	19.5	$\frac{19.5}{9.8}$	$\frac{16.9}{6.1}$	$\frac{16.2}{6.9}$

Table 3.7: Parallel execution times using the minimum and maximum overlap size, above and below the line, respectively, for the test problem (3.8).

Table 3.7 displays the execution times over varying numbers of vertical and horizontal strips, for different values of ε and N. In these numerical experiments, the number of

processors P is set equal to the number of horizontal strips S. The numerical experiments indicate that for serial processing, there is no advantage in increasing the number of horizontal strips, since the minimal executions times occurring when the number of horizontal strips is minimal at $S = 1$. From these results we can observe that the execution times are smaller for the maximal overlap size compared to that of the minimal overlap size.

Figure 3.6 displays the serial acceleration (execution time of the undecomposed algorithm $S = M = 1$ / minimal execution time of the two-level domain decomposition algorithm) of the two-level domain decomposition algorithm. From Figure 3.6, it is evident that there is a significant advantage of the two-level domain decomposition algorithm for serial processing. It can also be seen that as the number of mesh points N increases and/or ε decreases, the acceleration of the two-level domain decomposition algorithm increases.

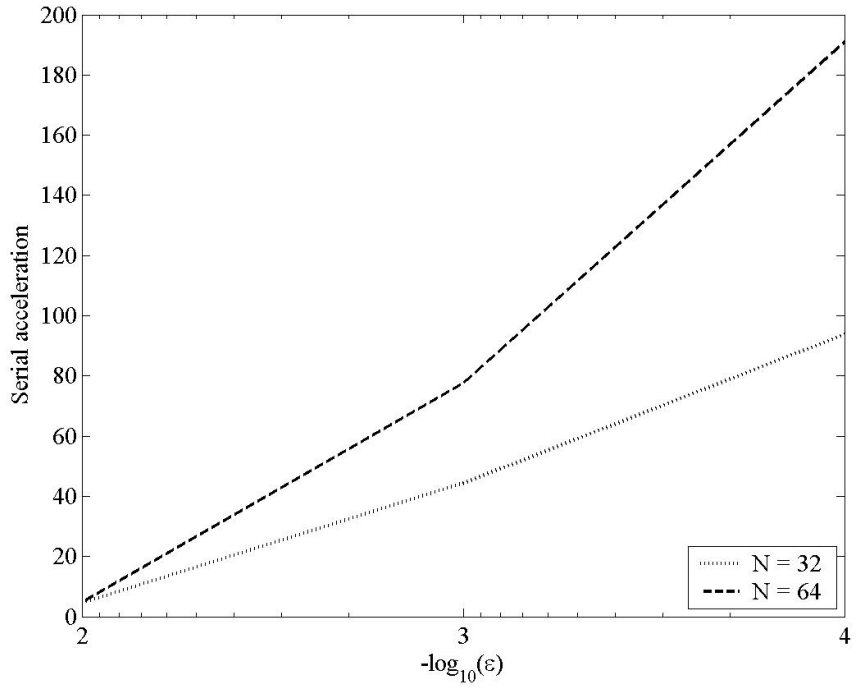

Figure 3.6: Serial acceleration of the two-level domain decomposition algorithm for the test problem (3.8).

If we let $T(N, P)$ represent the minimal execution time over all the two-level domain decompositions for a fixed N and number of processors P, then the parallel speedup is defined as $T(N, 1)/T(N, P)$. Using this definition, the parallel speedup of the two-level

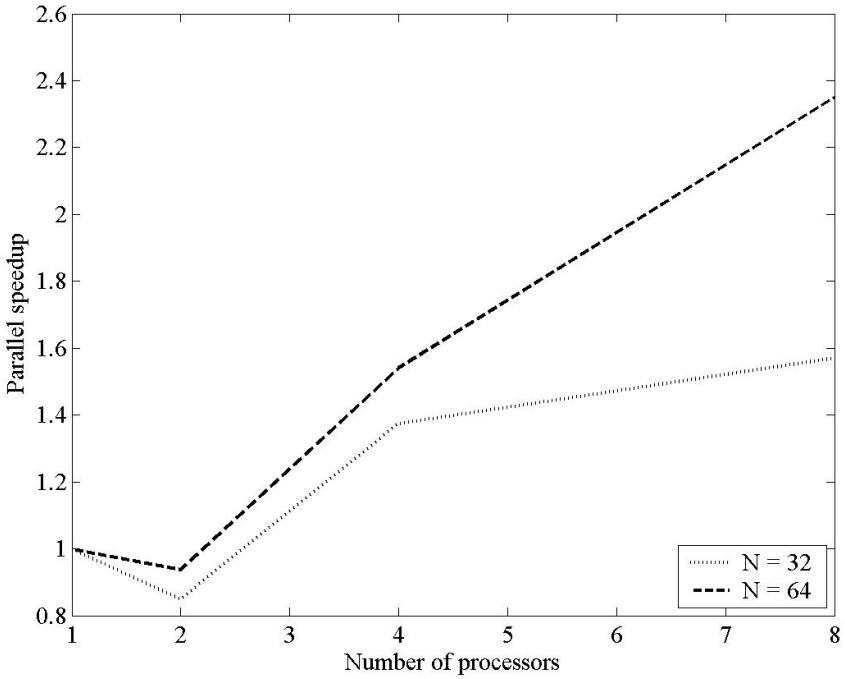

Figure 3.7: Parallel speedup of the two-level domain decomposition algorithm with $\varepsilon = 10^{-2}$ for the test problem (3.8).

domain decomposition is shown in Figure 3.7 for different values N with ε held fixed at 10^{-2}. From Figure 3.7, it is apparent that if the number of processors is greater than 4 the parallel speedup is greater than 1 for all values of N, indicating an advantage in parallel computing. It is also clear from Figure 3.7, that as the number of mesh points increases the parallel speedup also increases. Similarly, as the number of processors increase the parallel speedup also increases. Similar observations were made for different values of ε.

Numerical observations

From these numerical experiments, we observe that the two-level domain decomposition algorithm is uniformly convergent in ε in its outer iteration counts. It is also observed that the execution time of the two-level domain decomposition algorithm decreases as the overlap size increases. The serial execution times for the two-level domain decomposition algorithm show a considerable acceleration compared to the undecomposed method for certain combinations of vertical strips. It is also observed that for large values of N and/or small values of ε, parallel computing using the two-level domain decomposition

algorithm results in a moderate speedup.

3.4 Conclusions

In this chapter, we investigate the one-level monotone domain decomposition algorithm (3.1)–(3.2) based on the multiplicative Schwarz method. In Theorem 5, we prove monotone convergence of this algorithm. We apply this algorithm to the two test problems: the convection-diffusion problem with parabolic layers and the anisotropic convection-diffusion problem. Through the numerical experiments, we observe that for both of the test problems the one-level monotone domain decomposition algorithm is parameter uniformly convergent in its iteration counts, and the serial acceleration for both of the test problems indicates an advantage in using the one-level monotone domain decomposition algorithm.

We construct the two-level monotone domain decomposition algorithm (3.10)–(3.14) based on the combination of outer and inner iterates (the additive Schwarz method). In Theorem 6, we prove monotone convergence of the inner iterates. The numerical stability of the algorithm is proved. Numerical experiments are implemented for the convection-diffusion problem with parabolic layers and the anisotropic convection-diffusion problem. The anisotropic convection-diffusion problem is solved on serial and parallel computers. From these numerical experiments, we observe that the two-level domain decomposition algorithm is parameter uniformly convergent in its outer iteration counts. It is also observed that the execution time of the two-level monotone domain decomposition algorithm decreases as the overlap size increases. The serial execution times for the two-level monotone domain decomposition algorithm show a considerable acceleration compared to the undecomposed method for certain combinations of vertical strips. It is also observed that for large number of mesh points and/or small values of ε, parallel computing using the two-level monotone domain decomposition algorithm results in a moderate speedup.

Chapter 4

Monotone relaxation methods

This chapter deals with monotone relaxation methods based on the ω-Jacobi and successive underrelaxation methods. Point monotone ω-Jacobi and successive underrelaxation methods are constructed, and their monotone convergence are proven. Block monotone ω-Jacobi and successive underrelaxation methods are constructed, and their monotone convergence are proven. We compare the convergence of the block monotone iterative methods. A comparison of the point monotone and block monotone iterative methods is given. Results of numerical experiments are presented.

4.1 Introduction

In this chapter, we construct point monotone and block monotone relaxation methods based on the ω-Jacobi and successive underrelaxation methods, where ω is the relaxation parameter. The proposed methods combine the relaxation methods with the methods of upper and lower solutions. In [53], the monotone Jacobi and Gauss-Seidel methods with the relaxation parameter $\omega = 1$ are constructed and investigated.

We define the ω-Jacobi and successive underrelaxation methods for solving the nonlinear difference scheme (1.15), by modifying the ω-Jacobi and successive underrelaxation methods for solving the linear difference scheme [69] by introducing the diagonal preconditioner ωc^*, where c^* is from (1.20).

In the case of the block ω-Jacobi and successive underrelaxation methods, we modify the corresponding methods for linear difference schemes by introducing the diagonal preconditioner $\omega c^* I$, where I is the block identity matrix.

In Section 4.2, we construct the point monotone ω-Jacobi and successive underrelaxation methods and prove monotone convergence of the methods. In Section 4.3, the block ω-Jacobi and successive underrelaxation methods are constructed, and monotone convergence of these methods is proved. In Section 4.4, we compare the convergence properties of the point monotone and block monotone relaxation methods. Section 4.5 presents numerical experiments for the convection-diffusion problem with parabolic boundary layers and the anisotropic convection-diffusion problem.

4.2 Point monotone relaxation methods

For solving the nonlinear difference scheme (1.15), we now introduce two point monotone iterative methods, the ω-Jacobi (ωJAC) and successive underrelaxation (SUR) methods.

The ω-Jacobi iterative method for solving the linear problem

$$\mathcal{L}v(p) + f(p) = 0, \quad p \in \Omega^h, \quad v(p) = g(p), \quad p \in \partial\Omega^h, \tag{4.1}$$

is given by the recurrence formulae

$$\begin{gathered} d(p)z^{(n)}(p) = -\omega\mathcal{R}(p, v^{(n-1)}), \quad p \in \Omega^h, \quad z^{(n)}(p) = 0, \quad p \in \partial\Omega^h, \\ v^{(n)}(p) = v^{(n-1)}(p) + z^{(n)}(p), \quad p \in \overline{\Omega}^h, \quad \omega \in (0, 1], \\ \mathcal{R}(p, v^{(n-1)}) = \mathcal{L}v^{(n-1)}(p) + f(p), \end{gathered}$$

where ω is the relaxation parameter. We define the ω-Jacobi iterative method for solving the nonlinear difference scheme (1.15) in a similar way with the diagonal preconditioner ωc^*,

$$\begin{gathered} \mathcal{L}_{\omega\text{JAC}} z^{(n)}(p) = -\omega\mathcal{R}(p, v^{(n-1)}), \quad p \in \Omega^h, \quad z^{(n)}(p) = 0, \quad p \in \partial\Omega^h, \\ v^{(n)}(p) = v^{(n-1)}(p) + z^{(n)}(p), \quad p \in \overline{\Omega}^h, \quad \omega \in (0, 1], \\ \mathcal{L}_{\omega\text{JAC}} = d(p) + \omega c^*, \quad \mathcal{R}(p, v^{(n-1)}) = \mathcal{L}v^{(n-1)}(p) + f(p, v^{(n-1)}), \end{gathered} \tag{4.2}$$

where c^* is defined in (1.20).

Remark 5. A special case of the ω-Jacobi method is when $\omega = 1$. If we substitute $\omega = 1$ in (4.2), we get the resulting equation

$$(d(p) + c^*)v^{(n)}(p) = -f(p, v^{(n-1)}), \quad p \in \Omega^h.$$

This is the monotone Jacobi method for solving the nonlinear difference scheme.

The successive underrelaxation iterative method for solving the linear problem (4.1) is given by the recurrence formulae

$$d(p)z^{(n)}(p) - \omega \sum_{p' \in \sigma_L(p)} e(p,p')z^{(n)}(p') = -\omega \mathcal{R}(p, v^{(n-1)}), \quad p \in \Omega^h,$$

$$z^{(n)}(p) = 0, \quad p \in \partial\Omega^h, \quad v^{(n)}(p) = v^{(n-1)}(p) + z^{(n)}(p), \quad p \in \overline{\Omega}^h, \quad \omega \in (0,1],$$

$$\mathcal{R}(p, v^{(n-1)}) = \mathcal{L}v^{(n-1)}(p) + f(p),$$

$\sigma_L(p)$ is a set of stencil points corresponding to a strictly lower triangle part of $\sigma(p)$. We define the successive underrelaxation iterative (SUR) method for solving the nonlinear difference scheme (1.15) in a similar way with the diagonal preconditioner ωc^*. The SUR method is defined by the recurrence formulae

$$\mathcal{L}_{\text{SUR}} z^{(n)}(p) = -\omega \mathcal{R}(p, v^{(n-1)}), \quad p \in \Omega^h, \quad z^{(n)}(p) = 0, \quad p \in \partial\Omega^h, \tag{4.3}$$

$$v^{(n)}(p) = v^{(n-1)}(p) + z^{(n)}(p), \quad p \in \overline{\Omega}^h, \quad \omega \in (0,1],$$

$$\mathcal{L}_{\text{SUR}} z^{(n)}(p) = [d(p) + \omega c^*] z^{(n)}(p) - \omega \sum_{p' \in \sigma_L(p)} e(p,p')z^{(n)}(p'),$$

$$\mathcal{R}(p, v^{(n-1)}) = \mathcal{L}v^{(n-1)}(p) + f(p, v^{(n-1)}).$$

Remark 6. A special case of the SUR method is when $\omega = 1$. If we substitute $\omega = 1$ in (4.3), we get the resulting equation

$$\begin{aligned}(d(p) + c^*)v^{(n)}(p) - \sum_{p' \in \sigma_L(p)} e(p,p')v^{(n)}(p') \quad &= \quad c^* v^{(n-1)}(p) + \sum_{p' \in \sigma_U(p)} e(p,p')v^{(n-1)}(p') \\ &\quad -f(p, v^{(n-1)}), \quad p \in \Omega^h,\end{aligned}$$

where $\sigma_U(p)$ is the set of stencil points corresponding to a strictly upper triangular part of $\sigma(p)$ such that $\sigma'(p) = \sigma_U(p) \cup \sigma_L(p)$. The above method is the monotone Gauss-Seidel method for solving the nonlinear difference scheme.

Lemma 7. *If the coefficients of the difference operator $\mathcal{L}$ satisfy (1.16), then for the difference operators $\mathcal{L}_{\omega\text{JAC}}$ and $\mathcal{L}_{\text{SUR}}$ the maximum principle in Lemma 1 holds true.*

Proof. To prove the lemma, we show that the difference operators $\mathcal{L}_{\omega\text{JAC}}$ and $\mathcal{L}_{\text{SUR}}$ satisfy the conditions from (1.16).

For $\mathcal{L}_{\omega\text{JAC}} = d(p) + \omega c^*$, the diagonal entries are $d(p) + \omega c^* > 0$ and the off diagonal entries are $e(p,p') = 0$, $p' \in \sigma'(p)$. Therefore $\mathcal{L}_{\omega\text{JAC}}$ satisfies the conditions in (1.16).

For $\mathcal{L}_{\rm SUR} z^{(n)}(p) = [d(p)+\omega c^*]z^{(n)}(p) - \omega \sum_{p'\in\sigma_L(p)} e(p,p')z^{(n)}(p')$, the diagonal entries are $d(p) + \omega c^* \geq 0$ and the off diagonal entries are $e(p,p') \geq 0$, $p' \in \sigma'_L(p)$. Using the conditions $\omega \in (0,1]$, $c^* > 0$ and (1.16), we have

$$d(p) + \omega c^* - \omega \sum_{p'\in\sigma_L(p)} e(p,p') \geq d(p) - \sum_{p'\in\sigma(p)} e(p,p') > 0.$$

By Lemma 1, we conclude that the maximum principle holds for the difference operators $\mathcal{L}_{\omega\rm JAC}$ and $\mathcal{L}_{\rm SUR}$. □

Theorem 8. *Let $\overline{v}^{(0)}, \underline{v}^{(0)}$ be upper and lower solutions of (1.15). Then the upper and lower sequences $\{\overline{v}^{(n)}\}$ and $\{\underline{v}^{(n)}\}$ generated by the ω-Jacobi method (4.2) converge monotonically from above and below, respectively, to the unique solution v of (1.15):*

$$\underline{v}^{(n-1)}(p) \leq \underline{v}^{(n)}(p) \leq v(p) \leq \overline{v}^{(n)}(p) \leq \overline{v}^{(n-1)}(p), \quad p \in \overline{\Omega}^h, \quad n \geq 1. \tag{4.4}$$

Proof. We consider only the case of the upper sequence. The case for lower solutions may be proved in a similar way.

If $\overline{v}^{(0)}$ is an upper solution, then from (4.2) it follows that

$$\mathcal{L}_{\rm JAC} z^{(1)}(p) \leq 0, \quad p \in \Omega^h, \quad z^{(1)}(p) = 0, \quad p \in \partial\Omega^h.$$

By the maximum principle in Lemma 7, we conclude

$$z^{(1)}(p) = \overline{v}^{(1)}(p) - \overline{v}^{(0)}(p) \leq 0, \quad p \in \overline{\Omega}^h. \tag{4.5}$$

Using (4.2), we represent $\omega\mathcal{R}(p, v^{(1)})$ in the following form:

$$\begin{aligned}
\omega\mathcal{R}(p,\overline{v}^{(1)}) &= \omega[\mathcal{L}\overline{v}^{(1)}(p) + f(p,\overline{v}^{(1)})] \qquad (4.6)\\
&= \omega[d(p)\overline{v}^{(1)}(p) - \sum_{p'\in\sigma'(p)} e(p,p')\overline{v}^{(1)}(p') + f(p,\overline{v}^{(1)})]\\
&= \omega[\mathcal{R}(p,\overline{v}^{(0)}) + d(p)z^{(1)}(p) - \sum_{p'\in\sigma'(p)} e(p,p')z^{(1)}(p')\\
&\quad -(f(p,\overline{v}^{(0)}) - f(p,\overline{v}^{(1)}))]\\
&= -(d(p)+\omega c^*)z^{(1)}(p) + \omega d(p)z^{(1)}(p)\\
&\quad -\omega \sum_{p'\in\sigma'(p)} e(p,p')z^{(1)}(p') + \omega f_v^{(1)}(p)z^{(1)}(p)\\
&= -(1-\omega)d(p)z^{(1)}(p) - \omega \sum_{p'\in\sigma(p)} e(p,p')z^{(1)}(p')\\
&\quad -\omega(c^* - f_v^{(1)}(p))z^{(1)}(p), \quad p \in \Omega^h,
\end{aligned}$$

where the mean value theorem is used in the form

$$f(p,\overline{v}^{(0)}) - f(p,\overline{v}^{(1)}) = f(p,\overline{v}^{(0)}) - f(p,\overline{v}^{(0)} + z^{(1)}) = -f_v^{(1)}(p)z^{(1)},$$

$$f_v^{(1)}(p) = f_v[p,\overline{v}^{(0)}(p) + \theta^{(1)}(p)z^{(1)}(p)], \quad 0 < \theta^{(1)}(p) < 1.$$

Taking into account (1.16), (1.20) and (4.5), we conclude that $\mathcal{R}(p,\overline{v}^{(1)}) \geq 0$, $p \in \Omega^h$. By construction, $\overline{v}^{(1)}(p) = \overline{v}^{(0)}(p) = g(p)$, $p \in \partial\Omega^h$, therefore $\overline{v}^{(1)}(p)$ is an upper solution to the difference scheme (1.15).

By induction on n, we conclude that

$$z^{(n)}(p) \leq 0, \quad p \in \overline{\Omega}^h, \quad n \geq 1. \tag{4.7}$$

Thus, $\{\overline{v}^{(n)}\}$ is a monotonically decreasing sequence of upper solutions. This sequence is bounded below by $\underline{v}$, where $\underline{v}$ is any lower solution (1.19). This means that the function $v(p)$ defined as

$$v(p) = \lim_{n\to\infty} \overline{v}^{(n)}(p), \quad p \in \overline{\Omega}^h, \tag{4.8}$$

exists. We now prove that $v(p)$ is a solution to (1.15). From (4.2), we have

$$\lim_{n\to\infty} z^{(n)}(p) = \lim_{n\to\infty} (\overline{v}^{(n)}(p) - \overline{v}^{(n-1)}(p)) = 0, \quad p \in \overline{\Omega}^h.$$

Similar to (4.6), using the mean value theorem, we can show

$$\begin{aligned} \omega\mathcal{R}(p,\overline{v}^{(n)}) &= -(1-\omega)d(p)z^{(n)} - \omega \sum_{p'\in\sigma(p)} e(p,p')z^{(n)} \\ &\quad -\omega(c^* - f_v^{(n)}(p))z^{(n)}, \quad p \in \Omega^h, \end{aligned}$$

where $f_v^{(n)}(p) = f_v[p,\overline{v}^{(n-1)}(p) + \theta^{(n)}(p)z^{(n)}(p)]$, $0 < \theta^{(n)}(p) < 1$. Since $\lim z^{(n)} = 0$ as $n \to \infty$, we have

$$\lim_{n\to\infty} \omega\mathcal{R}(p,\overline{v}^{(n)}) = 0.$$

From here and (4.8), it follows that $v(p)$ satisfies

$$\mathcal{L}v + f(p,v) = 0, \quad p \in \Omega^h,$$

$$v(p) = \lim_{n\to\infty} \overline{v}^{(n)}(p) = g(p) \quad p \in \partial\Omega^h.$$

Therefore $v(p)$ is a solution of (1.15), and we prove the theorem. □

Theorem 9. *Let $\overline{v}^{(0)}, \underline{v}^{(0)}$ be upper and lower solutions of (1.15). Then the upper and lower sequences $\{\overline{v}^{(n)}\}$ and $\{\underline{v}^{(n)}\}$ generated by the* SUR *method (4.3) converge monotonically from above and below, respectively, to the unique solution v of (1.15):*

$$\underline{v}^{(n-1)}(p) \le \underline{v}^{(n)}(p) \le v(p) \le \overline{v}^{(n)}(p) \le \overline{v}^{(n-1)}(p), \quad p \in \overline{\Omega}^h, \quad n \ge 1. \tag{4.9}$$

Proof. This theorem is proved in a similar manner as Theorem 8. The full proof can be found in Appendix A. □

The following theorem gives a comparison result on convergence of the monotone sequences obtained by the ω-Jacobi method (4.2) and the SUR method (4.3).

Theorem 10. *Let $\{\overline{v}^{(n)}_{\omega\mathrm{JAC}}\}$, $\{\underline{v}^{(n)}_{\omega\mathrm{JAC}}\}$ and $\{\overline{v}^{(n)}_{\mathrm{SUR}}\}$, $\{\underline{v}^{(n)}_{\mathrm{SUR}}\}$ be the sequences of upper and lower solutions generated by the ω-Jacobi method (4.2) and the* SUR *method (4.3), respectively, where $\overline{v}^{(0)}_{\omega\mathrm{JAC}} = \overline{v}^{(0)}_{\mathrm{SUR}} = \overline{v}^{(0)}$ and $\underline{v}^{(0)}_{\omega\mathrm{JAC}} = \underline{v}^{(0)}_{\mathrm{SUR}} = \underline{v}^{(0)}$. Assume that the relaxation parameter ω used in each method is the same. Then*

$$\overline{v}^{(n)}_{\mathrm{SUR}}(p) \le \overline{v}^{(n)}_{\omega\mathrm{JAC}}(p), \quad \underline{v}^{(n)}_{\mathrm{SUR}}(p) \ge \underline{v}^{(n)}_{\omega\mathrm{JAC}}(p), \quad p \in \overline{\Omega}^h, \quad n \ge 1.$$

Proof. We prove this theorem only for the case of upper solutions. The case for lower solutions may by shown in a similar way. We first introduce the following notation:

$$z^{(n)}_{\omega\mathrm{JAC}} = \overline{v}^{(n)}_{\omega\mathrm{JAC}} - \overline{v}^{(n-1)}_{\omega\mathrm{JAC}}, \quad z^{(n)}_{\mathrm{SUR}} = \overline{v}^{(n)}_{\mathrm{SUR}} - \overline{v}^{(n-1)}_{\mathrm{SUR}}, \quad \zeta^{(n)} = \overline{v}^{(n)}_{\mathrm{SUR}} - \overline{v}^{(n)}_{\omega\mathrm{JAC}}.$$

From (4.2) and (4.3) with $n = 1$, we have

$$\begin{aligned}
(d(p) + \omega c^*)\zeta^{(1)}(p) &= (d(p) + \omega c^*)\overline{v}^{(1)}_{\mathrm{SUR}}(p) - (d(p) + \omega c^*)\overline{v}^{(1)}_{\omega\mathrm{JAC}}(p) \\
&= (d(p) + \omega c^*)(z^{(1)}_{\mathrm{SUR}}(p) + \overline{v}^{(0)}) \\
&\quad -(d(p) + \omega c^*)(z^{(1)}_{\omega\mathrm{JAC}}(p) + \overline{v}^{(0)}) \\
&= (d(p) + \omega c^*)z^{(1)}_{\mathrm{SUR}}(p) - (d(p) + \omega c^*)z^{(1)}_{\omega\mathrm{JAC}}(p) \\
&= \omega \sum_{p' \in \sigma_L(p)} e(p, p')z^{(1)}_{\mathrm{SUR}}(p').
\end{aligned}$$

Using (1.16), (4.7), (A.3) and $\omega \in (0, 1]$, we have $(d(p)+\omega c^*)\zeta^{(1)}(p) \le 0$. By construction, $\zeta^{(1)}(p) = 0$, $p \in \partial\Omega^h$, and applying the maximum principle in Lemma 1, we conclude that

$$\zeta^{(1)}(p) = \overline{v}^{(1)}_{\mathrm{SUR}}(p) - \overline{v}^{(1)}_{\omega\mathrm{JAC}}(p) \le 0, \quad p \in \overline{\Omega}^h. \tag{4.10}$$

From (4.2) and (4.3) for $n = 2$, we have

$$\begin{aligned}
(d(p)+\omega c^*)\zeta^{(2)}(p) &= (d(p)+\omega c^*)\overline{v}^{(2)}_{\rm SUR}(p) - (d(p)+\omega c^*)\overline{v}^{(2)}_{\omega\rm JAC}(p) \\
&= (d(p)+\omega c^*)\overline{z}^{(2)}_{\rm SUR}(p) - (d(p)+\omega c^*)\overline{z}^{(2)}_{\omega\rm JAC}(p) \\
&\quad +(d(p)+\omega c^*)\zeta^{(1)}(p) \\
&= -\omega[\mathcal{R}(p,\overline{v}^{(1)}_{\rm SUR}) - \mathcal{R}(p,\overline{v}^{(1)}_{\omega\rm JAC})] + \omega\sum_{p'\in\sigma_L(p)} e(p,p')z^{(1)}_{\rm SUR}(p') \\
&\quad +(d(p)+\omega c^*)\zeta^{(1)}(p) \\
&= -\omega[d(p)\zeta^{(1)}(p) - \sum_{p'\in\sigma'(p)} e(p,p')\zeta^{(1)}(p') \\
&\quad -(f(p,\overline{v}^{(1)}_{\omega\rm JAC}) - f(p,\overline{v}^{(1)}_{\rm SUR}))] + \omega\sum_{p'\in\sigma_L(p)} e(p,p')z^{(1)}_{\rm SUR}(p') \\
&\quad +(d(p)+\omega c^*)\zeta^{(1)}(p) \\
&= \omega[f(p,\overline{v}^{(1)}_{\omega\rm JAC}) - f(p,\overline{v}^{(1)}_{\rm SUR})] + \omega c^*\zeta^{(1)}(p) \\
&\quad +d(p)(1-\omega)\zeta^{(1)}(p) + \omega\sum_{p'\in\sigma'(p)} e(p,p')\zeta^{(1)}(p') \\
&\quad +\omega\sum_{p'\in\sigma_L(p)} e(p,p')z^{(1)}_{\rm SUR}(p').
\end{aligned}$$

From here, and the mean value theorem, we have,

$$\begin{aligned}
(d(p)+\omega c^*)\zeta^{(2)}(p) &= \omega(c^* - f^{(1)}_v(p))\zeta^{(1)}(p) + d(p)(1-\omega)\zeta^{(1)}(p) \\
&\quad +\omega\sum_{p'\in\sigma'(p)} e(p,p')\zeta^{(1)}(p') + \omega\sum_{p'\in\sigma_L(p)} e(p,p')z^{(1)}_{\rm SUR}(p'),
\end{aligned}$$

where

$$f(p,\overline{v}^{(1)}_{\omega\rm JAC}) - f(p,\overline{v}^{(1)}_{\rm SUR}) = -f^{(1)}_v(p)\zeta^{(1)}(p),$$

$$f^{(1)}_v(p) = f_v[p,\overline{v}^{(1)}_{\omega\rm JAC}(p) + \theta^{(1)}(p)\zeta^{(1)}(p)], \quad 0 < \theta^{(1)}(p) < 1.$$

Taking into account (1.16), (1.20), (A.3), (4.10) and $\omega \in (0,1]$, we have

$$(d(p)+\omega c^*)\zeta^{(2)}(p) \le 0.$$

By construction, $\zeta^{(2)}(p) = 0$, $p \in \partial\Omega^h$, and applying the maximum principle in Lemma 1, we have

$$\zeta^{(2)}(p) = \overline{v}^{(2)}_{\rm SUR}(p) - \overline{v}^{(2)}_{\omega\rm JAC}(p) \le 0, \quad p \in \overline{\Omega}^h.$$

By induction on n, we can show

$$\zeta^{(n)}(p) = \overline{v}^{(n)}_{\rm SUR}(p) - \overline{v}^{(n)}_{\omega\rm JAC}(p) \le 0, \quad p \in \overline{\Omega}^h, \quad n \ge 1.$$

Thus, we prove the theorem. □

Remark 7. Theorem 10 shows that if the same initial upper or lower solution and the same value of the relaxation parameter ω are used, the SUR method converges faster than the ω-Jacobi method.

4.3 Block monotone iterative methods

We now solve the nonlinear difference scheme (1.15) on a two dimensional rectangular mesh $\overline{\Omega}^h = \overline{\Omega}^{hx} \times \overline{\Omega}^{hy}$:

$$\overline{\Omega}^{hx} = \{x_i,\ 0 \le i \le N_x;\ x_0 = 0,\ x_{N_x} = 1;\ h_{xi} = x_{i+1} - x_i\},$$
$$\overline{\Omega}^{hy} = \{y_j,\ 0 \le j \le N_y;\ y_0 = 0,\ y_{N_y} = 1;\ h_{yj} = y_{j+1} - y_j\}.$$

We write down the difference scheme (1.15) at an interior mesh point $(x_i, y_j) \in \Omega^h$ in the form

$$d_{ij}v_{ij} - w_{ij}v_{i-1,j} - e_{ij}v_{i+1,j} - s_{ij}v_{i,j-1} - n_{ij}v_{i,j+1} + f(x_i, y_j, v_{ij}) + g^*_{ij} = 0, \qquad (4.11)$$

where g^*_{ij} is associated with the boundary function $g(p)$.

Assume that the coefficients of the difference scheme satisfy the inequalities

$$d_{ij} > 0, \quad w_{ij}, e_{ij}, s_{ij}, n_{ij} \ge 0, \qquad (4.12)$$

$$d_{ij} - (w_{ij} + e_{ij} + s_{ij} + n_{ij}) > 0, \quad i = 1, \ldots, N_x - 1, \quad j = 1, \ldots, N_y - 1,$$

and define vectors and diagonal matrices by

$$V_i = \left(v_{i,1}, \ldots, v_{i,N_y-1}\right)^T, \quad G^*_i = \left(g^*_{i,1}, \ldots, g^*_{i,N_y-1}\right)^T,$$
$$F_i(v_i) = \left(f(x_i, y_1, v_{i,1}), \ldots, f\left(x_i, y_{N_y-1}, v_{i,N_y-1}\right)\right)^T,$$
$$W_i = \operatorname{diag}\left(w_{i,1}, \ldots, w_{i,N_y-1}\right), \quad E_i = \operatorname{diag}\left(e_{i,1}, \ldots, e_{i,N_y-1}\right).$$

Then the difference scheme is represented in the vector-matrix form

$$A_iV_i - W_iV_{i-1} - E_iV_{i+1} + F_i(V_i) + G^*_i = 0, \quad i = 1, \ldots, N_x - 1, \qquad (4.13)$$

with the tridiagonal matrices A_i, $i = 1, \ldots, N_x - 1$,

$$A_i = \begin{bmatrix} d_{i,1} & -n_{i,1} & & & 0 \\ -s_{i,2} & d_{i,2} & -n_{i,2} & & \\ & \ddots & \ddots & \ddots & \\ & -s_{i,N_y-2} & d_{i,N_y-2} & -n_{i,N_y-2} & \\ 0 & & -s_{i,N_y-1} & d_{i,N_y-1} & \end{bmatrix}.$$

Matrices W_i and E_i contain the coupling coefficients of a mesh point, respectively, to the mesh point on the left (west) line and the mesh point on the right (east) line.

The block ω-Jacobi (BJAC) iterative method for solving the linear system $A_iV_i - W_iV_{i-1} - E_iV_{i+1} + F_i = 0$ has the form (see Varga [69, p.223] for details)

$$A_iZ_i^{(n)} = -\omega\mathcal{R}_i(V^{(n-1)}),$$

$$Z_i^{(n)} = V_i^{(n)} - V_i^{(n-1)}, \quad i = 1, \ldots, N_x - 1, \quad \omega \in (0, 1],$$

$$\mathcal{R}_i(V^{(n-1)}) = A_iV_i^{(n-1)} - W_iV_{i-1}^{(n-1)} - E_iV_{i+1}^{(n-1)} + F_i,$$

where ω is the relaxation parameter. We define the BJAC method for solving the nonlinear system (4.13) in a similar way with the diagonal preconditioner $\omega c^* I$, where I is the $(N_y - 1) \times (N_y - 1)$ identity matrix,

$$(A_i + \omega c^* I)Z_i^{(n)} = -\omega\mathcal{R}_i(V^{(n-1)}), \tag{4.14}$$

$$Z_i^{(n)} = V_i^{(n)} - V_i^{(n-1)}, \quad i = 1, \ldots, N_x - 1, \quad \omega = \text{const} \in (0, 1],$$

$$\mathcal{R}_i(V^{(n-1)}) = A_iV_i^{(n-1)} - W_iV_{i-1}^{(n-1)} - E_iV_{i+1}^{(n-1)} + F_i(V_i^{(n-1)}) + G_i^*$$

Remark 8. An advantage of the BJAC method is that each block (line) may be solved independently of each other and this method if fully parallisable.

The block successive underrelaxation ($0 < \omega \leq 1$) or overrelaxation ($1 \leq \omega \leq 2$) iterative methods for solving the linear system $A_iV_i - W_iV_{i-1} - E_iV_{i+1} + F_i = 0$ has the form (see Varga [69, p.223] for details)

$$A_iZ_i^{(n)} - \omega W_iZ_{i-1}^{(n)} = -\omega\mathcal{R}_i(V^{(n-1)}),$$

$$Z_i^{(n)} = V_i^{(n)} - V_i^{(n-1)}, \quad i = 1, \ldots, N_x - 1, \quad \omega \in (0, 2]$$

$$\mathcal{R}_i(V^{(n-1)}) = A_iV_i^{(n-1)} - W_iV_{i-1}^{(n-1)} - E_iV_{i+1}^{(n-1)} + F_i,$$

where ω is the relaxation parameter. We define the block successive underrelaxation (BSUR) method for solving the nonlinear system (4.13) in a similar way with the diagonal preconditioner $\omega c^* I$, where I is the $(N_y - 1) \times (N_y - 1)$ identity matrix,

$$(A_i + \omega c^* I) Z_i^{(n)} - \omega W_i Z_{i-1}^{(n)} = -\omega \mathcal{R}_i(V^{(n-1)}), \tag{4.15}$$

$$Z_i^{(n)} = V_i^{(n)} - V_i^{(n-1)}, \quad i = 1, \ldots, N_x - 1, \quad \omega = \text{const} \in (0, 1],$$

$$\mathcal{R}_i(V^{(n-1)}) = A_i V_i^{(n-1)} - W_i V_{i-1}^{(n-1)} - E_i V_{i+1}^{(n-1)} + F_i(V_i^{(n-1)}) + G_i^*.$$

Remark 9. An advantage of the BSUR method is that starting from $i = 1$ and finishing off with $i = N_x - 1$, in serial, we only solve tridiagonal systems for $V_i^{(n)}$, $i = 1, \ldots, N_x - 1$. Therefore a simple tridiagonal solver such as the Thomas algorithm can be used.

In the notation $V = (V_1, ..., V_{Nx-1})^T$, we have the following theorem.

Theorem 11. *Let $\overline{V}^{(0)}, \underline{V}^{(0)}$ be upper and lower solutions of (4.13). Then the upper and lower sequences $\{\overline{V}^{(n)}\}$ and $\{\underline{V}^{(n)}\}$ generated by the* BJAC *method (4.14) converge monotonically from above and below, respectively, to the unique solution V of (4.13):*

$$\underline{V}^{(n-1)} \leq \underline{V}^{(n)} \leq V \leq \overline{V}^{(n)} \leq \overline{V}^{(n-1)}, \quad n \geq 1. \tag{4.16}$$

Proof. We consider only the case of the upper sequences obtained by the BJAC method. The case of the lower solutions may be proved in a similar way.

From (4.12), it follows that A_i is an M-matrix [69], that is, $A_i^{-1} > 0$. Since $\omega c^* I$ is a positive diagonal matrix, then $A_i + \omega c^* I$ is also an M-matrix, that is,

$$(A_i + \omega c^* I)^{-1} > 0. \tag{4.17}$$

If $\overline{V}^{(0)}$ is an upper solution, then from (4.14) it follows that

$$(A_i + \omega c^* I) Z_i^{(1)} \leq 0 \quad i = 1, \ldots, N_x - 1. \tag{4.18}$$

From (4.17), we have $Z_i^{(1)} \leq 0$ $i = 1, \ldots, N_x - 1$, that is,

$$\overline{V}_i^{(1)} \leq \overline{V}_i^{(0)}, \quad i = 1, \ldots, N_x - 1. \tag{4.19}$$

We now need to prove that $\overline{V}^{(1)}$ is an upper solution, that is,

$$\mathcal{R}_i(\overline{V}^{(1)}) \geq 0, \quad i = 1, \ldots, N_x - 1. \tag{4.20}$$

From (4.14), we have

$$\begin{aligned}
\omega\mathcal{R}_i(\overline{V}^{(1)}) &= \omega\mathcal{R}_i(\overline{V}^{(0)} + Z^{(1)}) \\
&= \omega(A_i Z_i^{(1)} - W_i Z_{i-1}^{(1)} - E_i Z_{i+1}^{(1)} + F(\overline{V}_i^{(0)} + Z_i^{(1)}) + G_i^*) \\
&\quad +\omega(A_i \overline{V}_i^{(0)} - W_i \overline{V}_{i-1}^{(0)} - E_i \overline{V}_{i+1}^{(0)}) \\
&= \omega(A_i Z_i^{(1)} - W_i Z_{i-1}^{(1)} - E_i Z_{i+1}^{(1)} + F(\overline{V}_i^{(0)} + Z_i^{(1)}) - F(\overline{V}_i^{(0)})) \\
&\quad +\omega\mathcal{R}_i(\overline{V}^{(0)}) \\
&= \omega(A_i Z_i^{(1)} - W_i Z_{i-1}^{(1)} - E_i Z_{i+1}^{(1)} + F(\overline{V}_i^{(0)} + Z_i^{(1)}) - F(\overline{V}_i^{(0)})) \\
&\quad -(A_i + \omega c^* I) Z_i^{(1)} \\
&= -(1-\omega) A_i Z_i^{(1)} - \omega c^* Z_i^{(1)} - \omega E_i Z_{i+1}^{(1)} - \omega W_i Z_{i-1}^{(1)} \\
&\quad +\omega(F(\overline{V}_i^{(0)} + Z_i^{(1)}) - F(\overline{V}_i^{(0)})) \\
&= -(1-\omega) A_i Z_i^{(1)} - \omega(c^* I - F_{i,u}^{(1)}) Z_i^{(1)} - \omega E_i Z_{i+1}^{(1)} - \omega W_i Z_{i-1}^{(1)},
\end{aligned}$$

where we use the mean value theorem

$$F(V_i^{(0)} + Z_i^{(1)}) - F(V_i^{(0)}) = F_{i,u}^{(1)} Z_i^{(1)}.$$

$F_{i,u}^{(1)}$ is a diagonal $(N_y - 1) \times (N_y - 1)$ matrix with diagonal entries

$$(F_{i,u}^{(1)})_{jj} = \frac{\partial F_{ij}(\overline{V}_j^{(0)} + \Theta_j^{(0)} Z_j^{(1)})}{\partial u}, \quad j = 1, \ldots, N_y - 1,$$

where $\Theta_j^{(0)}$, $j = 1, \ldots, N_y - 1$, are diagonal matrices with entries lying in $(0, 1)$. From here, (4.12), (4.17), (4.19) and $\omega \in (0, 1]$, we conclude (4.20).

By induction on n, we are able to conclude that $\{\overline{V}^{(n)}\}$ is a monotonically decreasing sequence of upper solutions. By (1.19), this sequence is bounded below by $\underline{V}$, where $\underline{V}$ is any lower solution. Therefore the sequence converges. Let V be defined by

$$V = \lim_{n \to \infty} \overline{V}^{(n)}.$$

We now prove that $V = (V_1, \ldots, V_{N_x - 1})$ is the solution to (4.13). From (4.14), we have

$$\lim_{n \to \infty} Z_i^{(n)} = \lim_{n \to \infty} (\overline{V}_i^{(n)} - \overline{V}_i^{(n-1)}) = 0, \quad i = 1, \ldots, N_x - 1.$$

From here and (4.14), it follows that

$$\lim_{n \to \infty} -\omega\mathcal{R}_i(V^{(n-1)}) = \lim_{n \to \infty} ((A_i + \omega c^* I) Z_i^{(n)}) = 0, \quad i = 1, \ldots, N_x - 1.$$

Thus,

$$\mathcal{R}_i(V) = \lim_{n\to\infty} \mathcal{R}_i(V^{(n)}) = 0, \quad i = 1, \ldots, N_x - 1,$$

and hence, $V = (V_1, \ldots, V_{N_x-1})$ is a solution to (4.13). We prove the theorem. □

Theorem 12. *Let $\overline{V}^{(0)}, \underline{V}^{(0)}$ be upper and lower solutions of (4.13). Then the upper and lower sequences $\{\overline{V}^{(n)}\}$ and $\{\underline{V}^{(n)}\}$ generated by the* BSUR *method (4.15) converge monotonically from above and below, respectively, to the unique solution V of (4.13):*

$$\underline{V}^{(n-1)} \le \underline{V}^{(n)} \le V \le \overline{V}^{(n)} \le \overline{V}^{(n-1)}, \quad n \ge 1. \tag{4.21}$$

Proof. This theorem is proved in a similar manner as Theorem 11. The full proof can be found in Appendix A. □

The following theorem gives a comparison result on convergence of the monotone sequences obtained by the BJAC method (4.14) and the BSUR method (4.15).

Theorem 13. *Let $\{\overline{V}^{(n)}_{\text{BJAC}}\}$, $\{\underline{V}^{(n)}_{\text{BJAC}}\}$ and $\{\overline{V}^{(n)}_{\text{BSUR}}\}$, $\{\underline{V}^{(n)}_{\text{BSUR}}\}$ be the sequences of upper and lower solutions generated by the* BJAC *method (4.14) and the* BSUR *method (4.15), respectively, with $\overline{V}^{(0)}_{\text{BSUR}} = \overline{V}^{(0)}_{\text{BJAC}} = \overline{V}^{(0)}$ and $\underline{V}^{(0)}_{\text{BSUR}} = \underline{V}^{(0)}_{\text{BJAC}} = \underline{V}^{(0)}$. We assume that the relaxation parameter ω used in each method is the same. Then*

$$\overline{V}^{(n)}_{\text{BSUR}} \le \overline{V}^{(n)}_{\text{BJAC}}, \quad \underline{V}^{(n)}_{\text{BSUR}} \ge \underline{V}^{(n)}_{\text{BJAC}}, \quad n \ge 1.$$

Proof. We prove this theorem only for the case of the upper solutions. The case for the lower solutions may be proved in a similar way. We first introduce the following notation:

$$Z^{(n)}_{\text{BJAC},i} = \overline{V}^{(n)}_{\text{BJAC},i} - \overline{V}^{(n-1)}_{\text{BJAC},i}, \quad Z^{(n)}_{\text{BSUR},i} = \overline{V}^{(n)}_{\text{BSUR},i} - \overline{V}^{(n-1)}_{\text{BSUR},i},$$
$$\zeta^{(n)}_i = \overline{V}^{(n)}_{\text{BSUR},i} - \overline{V}^{(n)}_{\text{BJAC},i}, \quad i = 1, \ldots, N_x - 1.$$

For $n = 1$ and $i = 1, \ldots, N_x - 1$, using (4.14) and (4.15), we have

$$\begin{aligned}
(A_i + \omega c^* I)\zeta^{(n)}_i &= (A_i + \omega c^* I)\overline{V}^{(1)}_{\text{BSUR},i} - (A_i + \omega c^* I)\overline{V}^{(1)}_{\text{BJAC},i} \qquad (4.22)\\
&= (A_i + \omega c^* I)(Z^{(1)}_{\text{BSUR},i} + \overline{V}^{(0)}_i) - (A_i + \omega c^* I)(Z^{(1)}_{\text{BJAC},i} + \overline{V}^{(0)}_i)\\
&= (A_i + \omega c^* I)Z^{(1)}_{\text{BSUR},i} - (A_i + \omega c^* I)Z^{(1)}_{\text{BJAC},i}\\
&= \omega W_i Z^{(1)}_{\text{BSUR},i} - \omega\mathcal{R}_i(\overline{V}^{(0)}) + \omega\mathcal{R}_i(\overline{V}^{(0)})\\
&= \omega W_i Z^{(1)}_{\text{BSUR},i}.
\end{aligned}$$

From (4.12), (A.6), and $\omega \in (0,1]$, we have

$$(A_i + \omega c^* I)\zeta_i^{(1)} \le 0, \quad i = 1, \ldots, N_x - 1.$$

From here and (4.17), we conclude that

$$\zeta_i^{(1)} \le 0, \quad i = 1, \ldots, N_x - 1. \tag{4.23}$$

Thus, $\overline{V}^{(1)}_{\text{BSUR},i} \le \overline{V}^{(1)}_{\text{BJAC},i}$, $i = 1, \ldots, N_x - 1$.

Using (4.22) with $n = 2$, we have

$$\begin{aligned}
(A_i + \omega c^* I)\zeta_i^{(2)} &= (A_i + \omega c^* I)\overline{V}^{(2)}_{\text{BSUR},i} - (A_i + \omega c^* I)\overline{V}^{(2)}_{\text{BJAC},i} \\
&= (A_i + \omega c^* I)(Z^{(2)}_{\text{BSUR},i} + \overline{V}^{(1)}_{\text{BSUR},i}) \\
&\quad -(A_i + \omega c^* I)(Z^{(2)}_{\text{BJAC},i} + \overline{V}^{(1)}_{\text{BJAC},i}) \\
&= (A_i + \omega c^* I)Z^{(2)}_{\text{BSUR},i} - (A_i + \omega c^* I)Z^{(2)}_{\text{BJAC},i} \\
&\quad +(A_i + \omega c^* I)\zeta_i^{(1)} \\
&= \omega W_i Z^{(2)}_{\text{BSUR},i} - \omega \mathcal{R}_i(\overline{V}^{(1)}_{\text{BSUR}}) + \omega \mathcal{R}_i(\overline{V}^{(1)}_{\text{BJAC}}) \\
&\quad +(A_i + \omega c^* I)\zeta_i^{(1)} \\
&= \omega W_i Z^{(2)}_{\text{BSUR},i} + (A_i + \omega c^* I)\zeta_i^{(1)} \\
&\quad -\omega(\mathcal{R}_i(\overline{V}^{(1)}_{\text{BSUR}}) - \omega \mathcal{R}_i(\overline{V}^{(1)}_{\text{BJAC}})) \\
&= \omega W_i Z^{(2)}_{\text{BSUR},i} + (A_i + \omega c^* I)\zeta_i^{(1)} \\
&\quad -\omega(A_i\zeta_i^{(1)} - W_i\zeta_{i-1}^{(1)} - E_i\zeta_{i+1}^{(1)} + F_i(\overline{V}^{(1)}_{\text{BSUR},i}) - F_i(\overline{V}^{(1)}_{\text{BJAC},i})) \\
&= \omega W_i Z^{(2)}_{\text{BSUR},i} + (1-\omega)A_i\zeta_i^{(1)} + \omega W_i\zeta_{i-1}^{(1)} + \omega E_i\zeta_{i+1}^{(1)} \\
&\quad +\omega c^*\zeta_i^{(1)} - \omega(F_i(V^{(1)}_{\text{BSUR},i}) - F_i(V^{(1)}_{\text{BJAC},i})) \\
&= \omega W_i Z^{(2)}_{\text{BSUR},i} + (1-\omega)A_i\zeta_i^{(1)} + \omega W_i\zeta_{i-1}^{(1)} + \omega E_i\zeta_{i+1}^{(1)} \\
&\quad +\omega(c^* - F^{(2)}_{i,u})\zeta_i^{(1)},
\end{aligned}$$

where we use the mean value theorem similar to (A.8). From (1.20), (4.12), (A.6), (4.23), and $\omega \in (0,1]$, we conclude that

$$(A_i + \omega c^* I)\zeta_i^{(2)} \le 0, \quad i = 1, \ldots, N_x.$$

From here and (4.17), it follows that

$$\zeta_i^{(2)} \le 0, \quad i = 1, \ldots, N_x - 1.$$

By induction on n, we conclude that

$$\zeta_i^{(n)} = \overline{V}_{\text{BSUR},i}^{(n)} - \overline{V}_{\text{BJAC},i}^{(n)} \leq 0, \quad i = 1, \ldots, N_x - 1, \quad n \geq 1,$$

and prove the theorem. □

Remark 10. Theorem 13 shows that if the same initial upper and lower solution and the same value of the relaxation parameter ω are used, the BSUR method converges faster than the ω-Jacobi method.

4.4 Comparison of the point monotone and block monotone iterative methods

To compare the point monotone methods with the block monotone methods, we first write the point iterative methods in the same form as the block iterative methods. Let

$$A_i = D_i - S_i - N_i, \tag{4.24}$$

where D_i is a diagonal matrix, S_i is a lower off-diagonal matrix and N_i is a upper off-diagonal matrix. The point monotone ω-Jacobi method (4.2) may be written in the form

$$(D_i + \omega c^* I) Z_i^{(n)} = -\omega \mathcal{R}_i(V^{(n-1)}), \tag{4.25}$$

$$Z_i^{(n)} = V_i^{(n)} - V_i^{(n-1)}, \quad i = 1, \ldots, N_x - 1, \quad \omega = \text{const} \in (0, 1],$$

$$\mathcal{R}_i(V^{(n-1)}) = A_i V_i^{(n-1)} - W_i V_{i-1}^{(n-1)} - E_i V_{i+1}^{(n-1)} + F_i(V_i^{(n-1)}) + G_i^*.$$

The point monotone SUR method (4.3) may be written in the form

$$(D_i + \omega c^*) Z_i^{(n)} - \omega (S_i Z_i^{(n)} + W_i Z_{i-1}^{(n)}) = -\omega \mathcal{R}_i(V^{(n-1)}), \tag{4.26}$$

$$Z_i^{(n)} = V_i^{(n)} - V_i^{(n-1)}, \quad i = 1, \ldots, N_x - 1, \quad \omega = \text{const} \in (0, 1],$$

$$\mathcal{R}_i(V_i^{(n-1)}) = A_i V_i^{(n-1)} - W_i V_{i-1}^{(n-1)} - E_i V_{i+1}^{(n-1)} + F_i(V_i^{(n-1)}) + G_i^*.$$

The following theorem gives a comparison result on convergence of the monotone sequences obtained by the BJAC method (4.14) and the point monotone ω-Jacobi method (4.25).

Theorem 14. *Let* $\{\overline{V}^{(n)}_{\text{BJAC}}\}$, $\{\underline{V}^{(n)}_{\text{BJAC}}\}$, *and* $\{\overline{V}^{(n)}_{\omega\text{JAC}}\}$, $\{\underline{V}^{(n)}_{\omega\text{JAC}}\}$ *be the sequences of upper and lower solutions generated by the* BJAC *method (4.14) and by the monotone* ω*-Jacobi method (4.25), respectively, with* $\overline{V}^{(0)}_{\text{BJAC}} = \overline{V}^{(0)}_{\omega\text{JAC}} = \overline{V}^{(0)}$ *and* $\underline{V}^{(0)}_{\text{BJAC}} = \underline{V}^{(0)}_{\omega\text{JAC}} = \underline{V}^{(0)}$. *We assume that the relaxation parameter* ω *used in each method is the same . Then*

$$\overline{V}^{(n)}_{\text{BJAC}} \leq \overline{V}^{(n)}_{\omega\text{JAC}}, \quad \underline{V}^{(n)}_{\omega\text{JAC}} \leq \underline{V}^{(n)}_{\text{BJAC}}, \quad n \geq 1.$$

Proof. This theorem is proved in a similar manner as Theorem 13. The full proof can be found in Appendix A. □

The following theorem gives a comparison result on convergence of the monotone sequences obtained by the BSUR method (4.15) and the point monotone SUR method (4.26).

Theorem 15. *Let* $\{\overline{V}^{(n)}_{\text{BSUR}}\}$, $\{\underline{V}^{(n)}_{\text{BSUR}}\}$, *and* $\{\overline{V}^{(n)}_{\text{SUR}}\}$, $\{\underline{V}^{(n)}_{\text{SUR}}\}$ *be the sequences of upper and lower solutions generated by the* BSUR *method (4.15) and by the monotone* SUR *method (4.26), respectively, with* $\overline{V}^{(0)}_{\text{BSUR}} = \overline{V}^{(0)}_{\text{SUR}} = \overline{V}^{(0)}$ *and* $\underline{V}^{(0)}_{\text{BSUR}} = \underline{V}^{(0)}_{\text{SUR}} = \underline{V}^{(0)}$. *We assume that the relaxation parameter* ω *used in each method is the same . Then*

$$\overline{V}^{(n)}_{\text{BSUR}} \leq \overline{V}^{(n)}_{\text{SUR}}, \quad \underline{V}^{(n)}_{\text{SUR}} \leq \underline{V}^{(n)}_{\text{BSUR}}, \quad n \geq 1.$$

Proof. This theorem is proved in a similar manner as Theorem 13. The full proof can be found in Appendix A. □

Remark 11. Theorems 14 and 15 show that if the same initial upper or lower solutions and the same value of the relaxation parameter ω are used, the BJAC and BSUR methods converge faster than the ω-Jacobi and SUR methods, respectively, in regards to iteration counts.

4.5 Numerical experiments

We apply the monotone BSUR method and the monotone SUR method to two test problems. The number of mesh points in the x- and y-directions are set equal to N. In our numerical experiments, the stopping criteria for the iterates is

$$\max_{p \in \overline{\Omega}^h} \|v^{(n)}(p) - v^{(n-1)}(p)\| \leq 10^{-6}.$$

4.5.1 Convection-diffusion problem with parabolic layers

Consider the test problem (3.6). This problem is the convection-diffusion problem with parabolic boundary layers (1.9), which is characterised by an elliptic boundary layer close to $x = 1$ and by a parabolic boundary layers close to $y = 0$ and $y = 1$. To solve this problem numerically, we use the two dimensional piecewise uniform Shishkin mesh (1.11) constructed in Section 1.4.2. We also require the constants c_*, c^* from (1.20), and an initial solution. From (3.7) we have

$$c_* = \exp(-1), \quad c^* = 1.$$

Using (1.6) and (1.7), we can see that $\overline{u}^{(0)}(p) = 1$, $p \in \overline{\Omega}^h$ is an initial upper solution.

Figure 4.1 displays the iteration counts and the execution times of the monotone BSUR and monotone SUR methods over varying values of ω , with $N = 128$ and $\varepsilon = 10^{-3}$. From this figure, we can see that the iteration counts and the execution times are minimal when $\omega = 1$. From numerical experiments, $\omega = 1$ results in the minimal execution times and iteration counts for varying values of mesh points N and values of the perturbation parameter ε. From this, we conclude that $\omega = 1$ is optimal. All of the following numerical experiments have $\omega = 1$.

Table 4.1 displays the iteration counts of the monotone BSUR and monotone SUR methods for the test problems (3.6). From this table, we can see that for both methods the iteration counts stay constant for ε smaller that 10^{-5}. This indicates that both methods converge uniformly with respect to ε. From Table 4.1, we can also see that the iteration counts are smaller for the monotone BSUR method than for the monotone SUR method. This is consistent with Theorem 15.

Figure 4.2 displays the execution times of the monotone BSUR and monotone SUR methods for different values of N. From Figure 4.2, we can see that for ε smaller than 10^{-2} the monotone SUR method converges slightly faster than the monotone BSUR method.

4.5.2 Anisotropic convection-diffusion problem

Consider the test problem (3.8). This problem is the anisotropic convection-diffusion problem (1.12) and is characterised by an elliptic boundary layer close to $x = 1$. To solve this problem numerically, we use the two dimensional piecewise uniform Shishkin mesh (1.8) constructed in Section 1.4.3. We also require the constants c_*, c^* from (1.20) and an

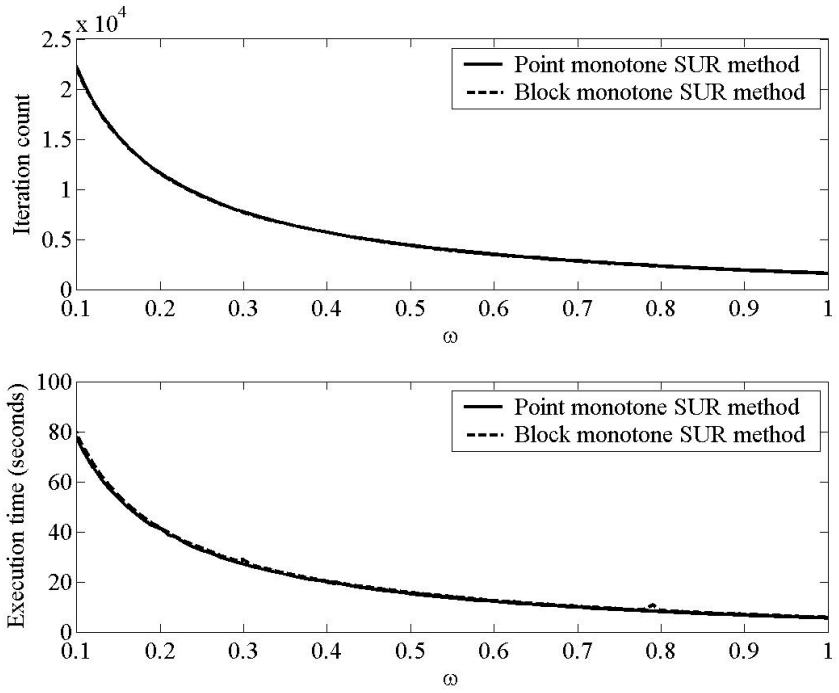

Figure 4.1: Execution times of the monotone SUR and monotone BSUR methods, for the test problem (3.6), over varying values of ω, $N = 128$ and $\varepsilon = 0.001$.

N/ε	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
	Monotone SUR iteration counts					
32	444	235	244	246	247	247
64	1477	598	617	625	626	626
128	4874	1635	1643	1673	1676	1677
	Monotone BSUR iteration counts					
32	252	225	243	245	245	246
64	820	563	614	622	623	623
128	2707	1505	1634	1665	1668	1669

Table 4.1: Iteration counts of the monotone SUR and monotone BSUR methods for the test problem (3.6).

initial solution. From (3.9) we have

$$c_* = \exp(-1), \quad c^* = 1.$$

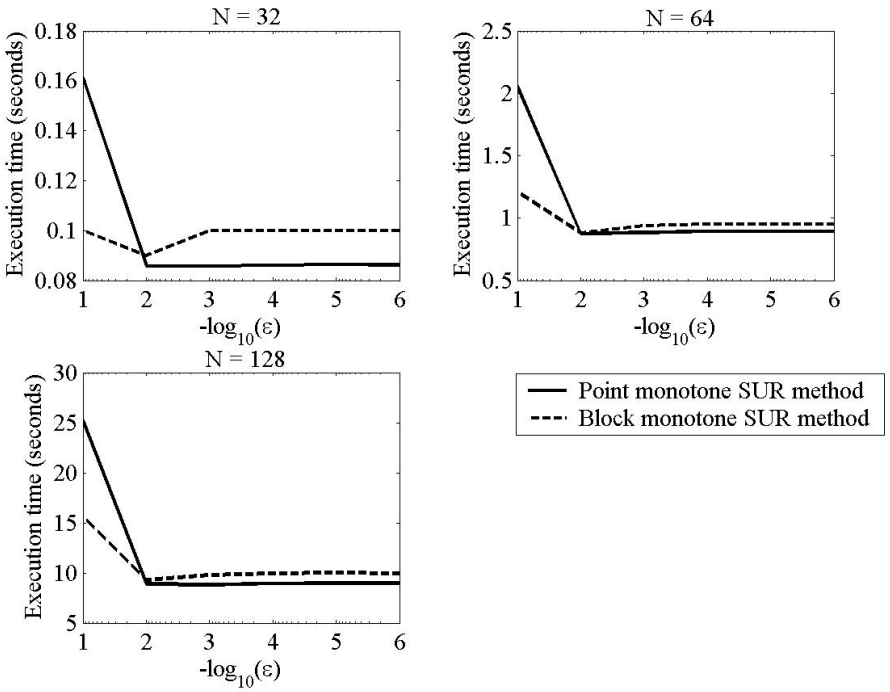

Figure 4.2: Execution times of the monotone SUR and monotone BSUR method for the test problem (3.6).

From (1.6) and (1.7), we can see that $\overline{u}^{(0)}(p) = 1$, $p \in \overline{\Omega}^h$ is an upper solution.

Figure 4.3 displays the iteration counts and the execution times of the monotone BSUR and monotone SUR methods over varying values of ω, with $N = 128$ and $\varepsilon = 10^{-3}$. From this figure, we can see that both the iteration counts and the execution times are minimal when $\omega = 1$. From numerical experiments, $\omega = 1$ results in the minimal execution times and iteration counts for varying values of mesh points N and values of the perturbation parameter ε. From this, we conclude that $\omega = 1$ is optimal. All of the following numerical experiments have $\omega = 1$. Table 4.2 displays the iteration counts of the monotone BSUR and monotone SUR methods. From this table, we can see that for monotone BSUR method and the monotone SUR method, the iteration counts stay constant for ε smaller than 10^{-4} and 10^{-3}, respectively. This indicates that both the methods converge uniformly with respect to ε. From Table 4.2, we can also see that the iteration counts are smaller for the monotone BSUR method that for the monotone SUR method. This is consistent with Theorem 15.

Figure 4.4 displays the execution times of the monotone BSUR and monotone SUR meth-

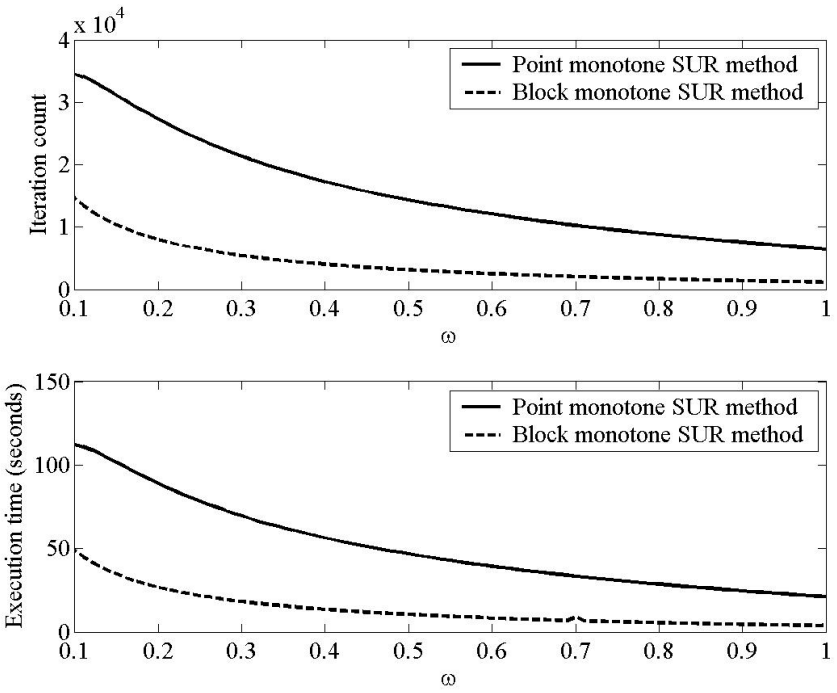

Figure 4.3: Execution times of the monotone SUR and monotone BSUR methods, for the test problem (3.8), over varying values of ω, $N = 128$ and $\varepsilon = 0.001$.

ods for different values of N . From Figure 4.4, we can see that the monotone BSUR method converges faster than the monotone SUR method.

N/ε	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
	Monotone SUR iteration counts					
32	649	685	690	690	690	690
64	2102	2194	2201	2202	2203	2203
128	6344	6510	6531	6536	6537	6537
	Monotone BSUR iteration counts					
32	86	132	166	171	171	171
64	278	343	449	464	466	466
128	908	923	1229	1279	1284	1285

Table 4.2: Iteration counts of the monotone SUR and monotone BSUR methods for the test problem (3.8).

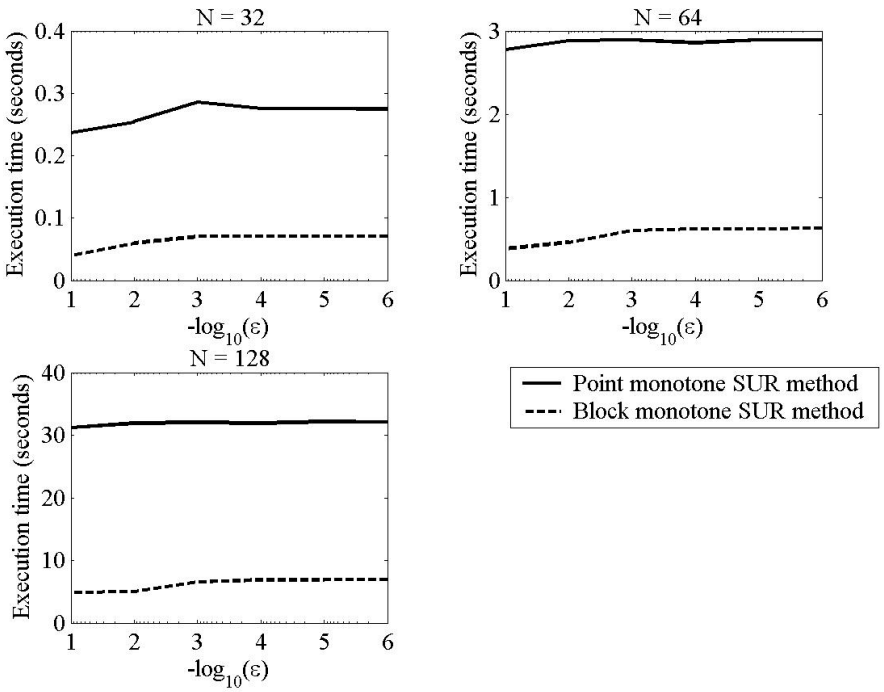

Figure 4.4: Execution times of the monotone SUR and monotone BSUR method for the test problem (3.8).

4.5.3 Numerical observations

From the numerical experiments, we observe that the monotone BSUR and monotone SUR methods are both uniformly convergent in their iteration counts with respect to the perturbation parameter ε. The numerical experiments also confirm Theorem 15 for both of the test problems that the iteration counts for the monotone BSUR are smaller than that of the monotone SUR method. We also observe that for both methods $\omega = 1$ is the optimal relaxation parameter. When $\omega = 1$, the monotone BSUR and monotone SUR methods are actually the block monotone Gauss-Seidel and point monotone Gauss-Seidel methods for solving the nonlinear difference scheme. When comparing the execution times of the monotone BSUR and monotone SUR methods, we can see that for the test problem (3.8), which is the anisotropic convection-diffusion problem, the monotone BSUR method converges the faster than the monotone SUR method, whereas for the test problem (3.6), which is the convection-diffusion problem with parabolic layers, the monotone SUR method converges slightly faster than the monotone BSUR method. It can be shown that point relaxation methods are not good for anisotropic problems due to poor error smoothing in

the x direction. Point relaxation methods only smooth in the direction which has 'strong coupling' in the operator [68], for anisotropic problems this is in the y-direction.

4.6 Conclusions

In this chapter, we investigate the point monotone relaxation and block monotone relaxation iterative methods. In Theorems 8 and 9, monotone convergence of the point monotone ω-Jacobi and point successive underrelaxation iterative methods, respectively, is proven. In Theorem 10, we prove that the successive underrelaxation iterative method converges to the solution of the difference scheme faster than the point monotone ω-Jacobi iterative method.

In Theorem 11 and Theorem 12, we prove monotone convergence of the block monotone ω-Jacobi and the block monotone successive underrelaxation methods, respectively. In Theorem 13, the convergence of the two block monotone methods are compared, and it is shown that the block successive underrelaxation method is the fastest. In Theorem 14, we compare the convergence of the point and block monotone ω-Jacobi methods. It is proven that the block monotone ω-Jacobi method is the fastest. Similarly, in Theorem 15, we compare the point and block monotone successive underrelaxation methods, where the block monotone successive underrelaxation method is the fastest.

From our numerical experiments, we observe that the point and block monotone successive underrelaxation methods are both ε-uniformly convergent with respect to their iteration counts. The numerical experiments also confirm Theorem 15 for both of the test problems. For both of the methods, it is observed that $\omega = 1$ is the optimal relaxation parameter. When comparing the execution times of the point and block monotone successive underrelaxation methods, it is observed that for the anisotropic convection-diffusion problem, the block monotone successive underrelaxation method is the fastest, whereas for the convection-diffusion problem with parabolic layers, the point monotone successive underrelaxation method converges slightly faster than the block monotone successive underrelaxation method.

Chapter 5

Composite monotone domain decomposition algorithms

In this chapter, we describe composite monotone domain decomposition algorithms based on the Jacobi, Gauss-Seidel, block Jacobi and block Gauss-Seidel methods for solving nonlinear singularly perturbed convection-diffusion equations. These algorithms are combinations of the monotone domain decomposition algorithms presented in Chapter 3 and the monotone relaxation methods presented in Chapter 4. The advantages of these composite monotone domain decomposition algorithms are that the algorithms solve only linear discrete systems at each iterative step of the iterative process and converge monotonically to the exact solution of the nonlinear problems. Numerical experiments are presented.

5.1 Introduction

In this chapter, composite monotone domain decomposition algorithms are constructed. Each of these domain decomposition algorithms is based on the combination of one of the domain decomposition algorithms presented in Chapter 3 and one of the relaxation methods presented in Chapter 4.

As with the domain decomposition algorithm, we split the domain into a set of overlapping vertical strips $\overline{\Omega}_m, \quad m = 1, \ldots, M$. One complete iterative step involves solving M problems on subdomains $\overline{\Omega}_m^h, \quad m = 1, \ldots, M$, in serial. The first subdomain $\overline{\Omega}_1^h$ is solved using one of the relaxation methods Jacobi, Gauss-Seidel, block Jacobi and block Gauss-Seidel which are presented in Chapter 4. The following subdomains $\overline{\Omega}_m^h, \quad m = 2, \ldots, M$,

are solved using one of the monotone domain decomposition algorithms presented in Chapter 3.

In Section 5.2 and 5.3, we investigate the monotone composite domain decomposition algorithm based on the one-level monotone domain decomposition algorithm presented in Section 3.2. In Section 5.2, two monotone composite domain decomposition algorithms based on the Jacobi and Gauss-Seidel methods are constructed. Monotone convergence of these algorithms is proven. A comparison of convergence of the two algorithms is presented. In Section 5.3, the composite monotone domain decomposition algorithms based the block Jacobi and block Gauss-Seidel methods are constructed. Monotone convergence of the two algorithms is proved, and a comparison of convergence of the algorithms is presented. A comparison of convergence of the composite monotone domain decomposition algorithms based on the point relaxation methods and the composite monotone domain decomposition algorithms based on the block relaxation methods is presented. In Section 5.4, we construct a two-level composite monotone domain decomposition algorithm based on the two-level monotone domain decomposition algorithm in Section 3.3. Monotone convergence of this algorithm is proven, and a remark is given illustrating the fully parallelisable nature of this algorithm. In Section 5.5, numerical experiments are presented for the convection-diffusion problem with parabolic boundary layers and the anisotropic convection-diffusion problem.

Some of the content of this chapter is published in [26].

5.2 Composite monotone domain decomposition algorithms based on the Jacobi and Gauss-Seidel methods

In this section, we are interested in the Jacobi domain decomposition algorithm (JAC-DD) and the Gauss-Seidel algorithm (GS-DD). The JAC-DD algorithm and the GS-DD algorithm are combinations of the Jacobi (JAC) method and the Gauss-Seidel (GS) method, respectively, with the domain decomposition (DD) algorithm based on the multiplicative Schwarz method.

As with the domain decomposition algorithm, we split the domain into a set of overlapping vertical strips $\overline{\Omega}_m$, $m = 1, \ldots, M$. One complete iterative step involves solving M problems on subdomains $\overline{\Omega}_m^h$, $m = 1, \ldots, M$, in serial. The first subdomain $\overline{\Omega}_1^h$ is

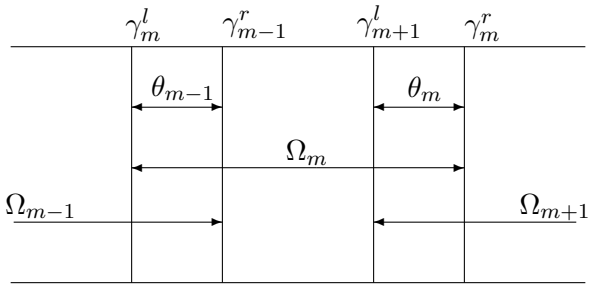

Figure 5.1: Fragment of the domain decomposition with overlapping subdomains Ω_{m-1}, Ω_m, Ω_{m+1} and overlaps θ_{m-1}, θ_m.

solved using either the JAC method or the GS method. The following subdomains $\overline{\Omega}_m^h$, $m = 2, \ldots, M$, are solved using the monotone domain decomposition algorithm. The algorithm for the monotone JAC-DD and the GS-DD algorithms are as follows:

1. Initialisation: On the whole mesh $\overline{\Omega}^h$, choose an initial function $v^{(0)}(p)$, $p \in \overline{\Omega}^h$, satisfying the boundary condition $v^{(0)}(p) = g(p)$ on $\partial\Omega^h$.

2. On the first subdomain $\overline{\Omega}_1^h$, compute mesh function $z_1^{(n)}(p)$ by using either the Jacobi method (4.2) or the Gauss-Seidel method (4.3), and denote

$$v_1^{(n)}(p) = v^{(n-1)}(p) + z_1^{(n)}(p), \quad p \in \overline{\Omega}_1^h. \tag{5.1}$$

3. On subdomains $\overline{\Omega}_m^h$, $m = 2, \ldots, M$, compute in serial mesh functions $v_m^{(n)}(p)$, $m = 2, \ldots, M$, satisfying the difference schemes

$$(\mathcal{L} + c^*) z_m^{(n)}(p) = -\mathcal{R}(p, v^{(n-1)}), \quad p \in \Omega_m^h, \tag{5.2}$$

$$z_m^{(n)}(p) = v_m^{(n)}(p) - v^{(n-1)}(p),$$

$$\mathcal{R}(p, v^{(n-1)}) = \mathcal{L} v^{(n-1)}(p) + f(p, v^{(n-1)}),$$

with the boundary conditions

$$z_m^{(n)}(p) = \begin{cases} z_{m-1}^{(n)}(p), & p \in \gamma_m^{hl}, \\ 0, & p \in \gamma_m^{hr} \cup \gamma_m^{h0}, \end{cases}$$

where

$$\gamma_m^{hl} = \gamma_m^l \cap \overline{\Omega}_m^h, \quad \gamma_m^{hr} = \gamma_m^r \cap \overline{\Omega}_m^h, \quad \gamma_m^{h0} = \gamma_m^0 \cap \overline{\Omega}_m^h.$$

4. Compute the solution $v^{(n)}(p)$, $p \in \overline{\Omega}^h$ by piecing together the solutions on the subdomains

$$v^{(n)}(p) = \begin{cases} v_m^{(n)}(p), & p \in \overline{\Omega}_m^h \setminus \theta_m^h, \ m = 1, \ldots, M-1; \\ v_M^{(n)}(p), & p \in \overline{\Omega}_M^h. \end{cases} \tag{5.3}$$

5. Stopping criterion: if

$$\max_{p \in \overline{\Omega}^h} |v^{(n)}(p) - v^{(n-1)}(p)| \le \Delta,$$

where Δ is the required accuracy, then stop; otherwise continue iteration by going to Step 2.

The following theorem gives the convergence property of the Jacobi domain decomposition algorithm (JAC-DD).

Theorem 16. *Let $\overline{v}^{(0)}$, $\underline{v}^{(0)}$ be upper and lower solutions of (1.15). Then the upper and lower sequences $\{\overline{v}^{(n)}\}$ and $\{\underline{v}^{(n)}\}$ generated by the monotone* JAC-DD *algorithm (5.1)–(5.3), converge monotonically from above and below, respectively, to the unique solution v of (1.15):*

$$\underline{v}^{(n-1)}(p) \le \underline{v}^{(n)}(p) \le v(p) \le \overline{v}^{(n)}(p) \le \overline{v}^{(n-1)}(p), \quad p \in \overline{\Omega}^h, \quad n \ge 1.$$

Proof. We consider only the case of upper sequences, as the case of lower sequences is proved in a similar manner.

Let $\overline{v}^{(0)}(p)$, $p \in \overline{\Omega}^h$ be an upper solution of (1.15) satisfying the boundary condition. From (5.1) with $n = 1$, $m = 1$ and using (4.4) from Theorem 8, we have

$$z_1^{(1)}(p) \le 0, \quad p \in \overline{\Omega}_1^h. \tag{5.4}$$

From here and (5.2) with $n = 1$ and $m = 2$, we have

$$(\mathcal{L} + c^*) z_2^{(n)}(p) = -\mathcal{R}(p, \overline{v}^{(0)}) \le 0, \quad p \in \Omega_2^h,$$

$$z_2^{(1)}(\gamma_2^{hl}) = z_1^{(1)}(\gamma_2^{hl}) \le 0, \quad z_2^{(1)}(\gamma_2^{hr}) = 0, \quad z_2^{(1)}(\gamma_2^{h0}) = 0.$$

By the maximum principle in Lemma 1, we conclude

$$z_2^{(1)}(p) \le 0, \quad p \in \overline{\Omega}_2^h,$$

and by induction on m, $z_m^{(1)}(p) \le 0$, $p \in \overline{\Omega}_m^h$, $m = 2, \ldots, M$. From here and (5.4), it follows that

$$z_m^{(1)}(p) \le 0, \quad p \in \overline{\Omega}_m^h, \quad m = 1, \ldots, M. \tag{5.5}$$

We now prove that $v^{(1)}(p)$ defined in Step 4 is an upper solution, that is,

$$\mathcal{R}(p, v^{(1)}) \ge 0, \quad p \in \Omega^h. \tag{5.6}$$

From (5.2) and the mean value theorem,

$$\begin{aligned}
\mathcal{R}(p, v_m^{(1)}) &= \mathcal{R}(p, \overline{v}^{(0)} + z_m^{(1)}) \\
&= \mathcal{L}\overline{v}^{(0)}(p) + \mathcal{L}z_m^{(1)}(p) + f(p, \overline{v}^{(0)} + z_m^{(1)}) \\
&= \mathcal{L}\overline{v}^{(0)}(p) + f(p, \overline{v}^{(0)}) + \mathcal{L}z_m^{(1)}(p) + f(p, \overline{v}^{(0)} + z_m^{(1)}) - f(p, \overline{v}^{(0)}) \\
&= \mathcal{R}(p, \overline{v}^{(0)}) + \mathcal{L}z_m^{(1)}(p) + f_v^{(1)}(p) z_m^{(1)}(p) \\
&= -(\mathcal{L} + c^*) z_m^{(1)}(p) + \mathcal{L}z_m^{(1)}(p) + f_v^{(1)}(p) z_m^{(1)}(p) \\
&= -(c^* - f_v^{(1)}(p)) z_m^{(1)}(p), \quad p \in \Omega_m^h, \quad m = 1, \ldots, M,
\end{aligned}$$

where $f_v^{(1)}(p) = f_v[p, \overline{v}^{(0)}(p) + \Theta_m^{(1)}(p) z_m^{(1)}(p)]$, $0 < \Theta_m^{(1)}(p) < 1$. From here, (1.20) and (5.5), we have

$$\mathcal{R}(p, v_m^{(1)}) = -(c^* - f_v^{(1)}(p)) z_m^{(1)} \ge 0 \quad p \in \Omega_m^h, \quad m = 1, \ldots, M.$$

From here and (5.3), we have

$$\mathcal{R}(p, v^{(1)}) \ge 0, \quad p \in \Omega^h \setminus \gamma^{hl}, \quad \gamma^{hl} = \bigcup_{m=2}^{M} \gamma_m^{hl}.$$

We therefore only need to verify (5.6) on the boundaries γ_m^{hl}, $m = 2, \ldots, M$. Let

$$\zeta_m^{(n)}(p) = v_m^{(n)}(p) - v_{m+1}^{(n)}(p) = z_m^{(n)}(p) - z_{m+1}^{(n)}, \quad p \in \overline{\theta}_m^h.$$

Using (4.2) with $\omega = 1$ and (5.2) with $m = 2$, we obtain

$$\begin{aligned}
(\mathcal{L} + c^*)\zeta_1^{(1)}(p) &= (\mathcal{L} + c^*) z_1^{(1)}(p) - (\mathcal{L} + c^*) z_2^{(1)}(p) \\
&= (\mathcal{L} + c^*) z_1^{(1)}(p) + \mathcal{R}(p, v^{(0)}) \\
&= (d(p) + c^*) z_1^{(1)}(p) - \sum_{p' \in \sigma'(p)} e(p, p') z_1^{(1)}(p') + \mathcal{R}(p, v^{(0)}) \\
&= -\sum_{p' \in \sigma(p)} e(p, p') z_1^{(1)}(p'), \quad p \in \theta_1^h.
\end{aligned}$$

From here, (5.4) and (1.16), it follows that

$$(\mathcal{L}+c^*)\zeta_1^{(1)}(p)\geq 0,\quad p\in\theta_1^h.$$

Using (5.2), we have

$$\zeta_1^{(1)}(\gamma_1^{hr})\geq 0,\quad \zeta_1^{(1)}(\gamma_2^{hl})=0,\quad \zeta_1^{(1)}(\gamma_1^{h0}\cap\gamma_2^{h0})=0,$$

and by the maximum principle in Lemma 1,

$$\zeta_1^{(1)}(p)\geq 0,\quad p\in\overline{\theta}_1^h. \tag{5.7}$$

For $m=2,\ldots,M-1$, from (5.2) we have

$$\begin{gathered}(\mathcal{L}+c^*)\zeta_m^{(1)}(p)=0,\quad p\in\theta_m^h,\\ \zeta_m^{(1)}(\gamma_m^{hr})\geq 0,\quad \zeta_m^{(1)}(\gamma_{m+1}^{hl})=0,\quad \zeta_m^{(1)}(\gamma_1^{h0}\cap\gamma_2^{h0})=0,\end{gathered}$$

and by the maximum principle in Lemma 1, conclude $\zeta_m^{(1)}\geq 0$, $p\in\overline{\theta}_m^h$, $m=2,\ldots,M$. From here and (5.7), it follows that

$$v_m^{(1)}(p)\geq v_{m+1}^{(1)}(p),\quad p\in\overline{\theta}_m^h,\quad m=1,\ldots,M-1.$$

For $p\in\gamma_{m+1}^{hl}$, $m=1,\ldots,M-1$, we have

$$\begin{aligned}\mathcal{R}(p,v^{(1)}) &= (d(p)+c^*)v^{(1)}(p)-\sum_{p'\in\sigma'(p)}e(p,p')v^{(1)}(p')\\ &= (d(p)+c^*)v^{(1)}(p)-\sum_{p'\in\sigma_L(p)}e(p,p')v^{(1)}(p')\\ &\quad -\sum_{p'\in\sigma_U(p)}e(p,p')v^{(1)}(p')\\ &= (d(p)+c^*)v_m^{(1)}(p)-\sum_{p'\in\sigma_L(p)}e(p,p')v_m^{(1)}(p')\\ &\quad -\sum_{p'\in\sigma_U(p)}e(p,p')v_{m+1}^{(1)}(p')\\ &\geq (d(p)+c^*)v_m^{(1)}(p)-\sum_{p'\in\sigma_L(p)}e(p,p')v_m^{(1)}(p')\\ &\quad -\sum_{p'\in\sigma_U(p)}e(p,p')v_m^{(1)}(p')\\ &= \mathcal{R}(p,v_m^{(1)})\geq 0.\end{aligned}$$

Thus, $v^{(1)}$ is an upper solution to (1.15). By induction on n, $\{\overline{v}^{(n)}\}$ is a monotonically decreasing sequence of upper solutions bounded below by $\underline{v}$ (1.19), where $\underline{v}$ is any lower

solution. Let $v(p)$ be defined by

$$v(p) = \lim_{n\to\infty} \overline{v}^{(n)}(p), \quad p \in \overline{\Omega}^h.$$

We now prove that $v(p)$ is the solution to (1.15). From (5.1)–(5.3),

$$\lim_{n\to\infty} z_m^{(n)} = \lim_{n\to\infty} (v_m^{(n)} - v_m^{(n-1)}) = 0, \quad m = 1, \ldots, M.$$

From here and (4.2),

$$\mathcal{R}(p, v) = \lim_{n\to\infty} \mathcal{R}(p, v^{(n-1)}) = -[\lim_{n\to\infty} ((d(p) + c^*) z_1^{(n)}(p)] = 0, \quad p \in \Omega_1^h \setminus \theta_1^h.$$

Using (5.2) and (5.3), we have

$$\begin{aligned} \mathcal{R}(p, v) &= \lim_{n\to\infty} \mathcal{R}(p, v^{(n)}) = \lim_{n\to\infty} \mathcal{R}(p, v_m^{(n)}) = -\lim_{n\to\infty} (\mathcal{L} + c^*) z_m^{(n)}(p) = 0, \\ & p \in \Omega_m^h \setminus \theta_m^h, \quad m = 2, \ldots, M-1, \quad p \in \Omega_M. \end{aligned}$$

Thus,

$$\mathcal{R}(p, v) = 0, \quad p \in \Omega^h.$$

Since $v(p) = g(p)$, $p \in \partial\Omega^h$, we conclude that $v(p)$ is the solution of (1.15). □

The following theorem gives the convergence property of the Gauss-Seidel domain decomposition algorithm (GS-DD).

Theorem 17. *Let $\overline{v}^{(0)}$, $\underline{v}^{(0)}$ be upper and lower solutions of (1.15). Then the upper and lower sequences $\{\overline{v}^{(n)}\}$ and $\{\underline{v}^{(n)}\}$ generated by the monotone* GS-DD *algorithm (5.1)–(5.3), converge monotonically from above and below, respectively, to the unique solution v of (1.15):*

$$\underline{v}^{(n-1)}(p) \le \underline{v}^{(n)}(p) \le v(p) \le \overline{v}^{(n)}(p) \le \overline{v}^{(n-1)}(p), \quad p \in \overline{\Omega}^h, \quad n \ge 1.$$

Proof. This theorem is proved in a similar manner as Theorem 16. The full proof can be found in Appendix B. □

The following theorem gives a comparison result on convergence of the monotone sequences obtained by the JAC-DD algorithm (5.1)–(5.3) and the GS-DD algorithm (5.1)–(5.3).

Theorem 18. *Let* $\{\overline{v}^{(n)}_{\text{JAC-DD}}\}$, $\{\underline{v}^{(n)}_{\text{JAC-DD}}\}$ *and* $\{\overline{v}^{(n)}_{\text{GS-DD}}\}$, $\{\underline{v}^{(n)}_{\text{GS-DD}}\}$ *be the sequences of upper and lower solutions generated by the* JAC-DD *algorithm (5.1)–(5.3) and the* GS-DD *algorithm (5.1)–(5.3), respectively, where* $\overline{v}^{(0)}_{\text{JAC-DD}} = \overline{v}^{(0)}_{\text{GS-DD}} = \overline{v}^{(0)}$ *and* $\underline{v}^{(0)}_{\text{JAC-DD}} = \underline{v}^{(0)}_{\text{GS-DD}} = \underline{v}^{(0)}$. *Then*

$$\underline{v}^{(n)}_{\text{JAC-DD}}(p) \leq \underline{v}^{(n)}_{\text{GS-DD}}(p) \leq \overline{v}^{(n)}_{\text{GS-DD}}(p) \leq \overline{v}^{(n)}_{\text{JAC-DD}}(p), \quad p \in \overline{\Omega}^h, \quad n \geq 1.$$

Proof. We prove this theorem only for the case of upper solutions. The case for lower solutions may be shown in a similar way. We introduce the following notation:

$$\zeta^{(n)}_m(p) = \overline{v}^{(n)}_{\text{JAC-DD},m}(p) - \overline{v}^{(n)}_{\text{GS-DD},m}(p), \quad p \in \overline{\Omega}_m, \quad m = 1, \ldots, M.$$

From (5.1) and Theorem 10 with $m = 1$ and $n = 1$, we have

$$\zeta^{(1)}_1(p) = \overline{v}^{(1)}_{\text{JAC-DD},1}(p) - \overline{v}^{(1)}_{\text{GS-DD},1}(p) \geq 0, \quad p \in \overline{\Omega}^h_1. \tag{5.8}$$

From (5.2), for $m = 2$ and $n = 1$ it follows that,

$$\begin{aligned}
(\mathcal{L} + c^*)\zeta^{(1)}_2(p) &= (\mathcal{L} + c^*)\overline{v}^{(1)}_{\text{JAC-DD},2}(p) - (\mathcal{L} + c^*)\overline{v}^{(1)}_{\text{GS-DD},2}(p) \\
&= (\mathcal{L} + c^*)z^{(1)}_{\text{JAC-DD},2}(p) - (\mathcal{L} + c^*)\overline{v}^{(0)}(p) \\
&\quad -(\mathcal{L} + c^*)z^{(1)}_{\text{GS-DD},2}(p) + (\mathcal{L} + c^*)\overline{v}^{(0)}(p) \\
&= -\mathcal{R}(p, \overline{v}^{(0)}) + \mathcal{R}(p, \overline{v}^{(0)}) = 0, \quad p \in \Omega^h_2,
\end{aligned}$$

Using (5.8), we get

$$\zeta^{(1)}_2(\gamma^{hl}_2) = \overline{v}^{(1)}_{\text{JAC-DD},1}(\gamma^{hl}_2) - \overline{v}^{(1)}_{\text{GS-DD},1}(\gamma^{hl}_2) \geq 0, \quad \zeta^{(1)}_2(\gamma^{hr}_2 \cup \gamma^{h0}_2) = 0.$$

By the maximum principle in Lemma 1, we conclude

$$\zeta^{(1)}_2(p) \geq 0, \quad p \in \overline{\Omega}^h_2,$$

and, by induction on m, $\zeta^{(1)}_m(p) \geq 0$, $p \in \overline{\Omega}^h_m$, $m = 2, \ldots, M$. From here and (5.8), it follows that

$$\zeta^{(1)}_m(p) = \overline{v}^{(1)}_{\text{JAC-DD},m}(p) - \overline{v}^{(1)}_{\text{GS-DD},m}(p) \geq 0, \quad p \in \overline{\Omega}^h_m, \quad m = 1, \ldots, M. \tag{5.9}$$

From here and (5.3), we conclude that

$$\overline{v}^{(1)}_{\text{JAC-DD}}(p) \geq \overline{v}^{(1)}_{\text{GS-DD}}(p), \quad p \in \overline{\Omega}^h. \tag{5.10}$$

From (5.1) and the mean value theorem, for $m = 1$ and $n = 2$ it follows that,

$$\begin{aligned}(d(p)+c^*)\zeta_1^{(2)}(p) &= (d(p)+c^*)\overline{v}^{(2)}_{\text{JAC-DD},1}(p) - (d(p)+c^*)\overline{v}^{(2)}_{\text{GS-DD},1}(p)\\ &= (d(p)+c^*)z^{(2)}_{\text{JAC-DD},1}(p) - (d(p)+c^*)z^{(2)}_{\text{GS-DD},1}(p)\\ &\quad +(d(p)+c^*)\zeta_1^{(1)}(p), \quad p\in\Omega_1^h,\end{aligned}$$

From here, (4.2) and (4.3), we get

$$\begin{aligned}(d(p)+c^*)\zeta_1^{(2)}(p) &= -\sum_{p'\in\sigma'(p)} e(p,p')z^{(2)}_{\text{GS-DD},1}(p') + \mathcal{R}(p,\overline{v}^{(1)}_{\text{GS-DD}}) - \mathcal{R}(p,\overline{v}^{(1)}_{\text{JAC-DD}})\\ &\quad +(d(p)+c^*)\zeta_1^{(1)}(p)\\ &= -\sum_{p'\in\sigma'(p)} e(p,p')z^{(2)}_{\text{GS-DD},1}(p') - d(p)\zeta_1^{(1)}(p)\\ &\quad +\sum_{p'\in\sigma'(p)} e(p,p')\zeta_1^{(1)}(p') + f(p,\overline{v}^{(1)}_{\text{GS-DD}}) - f(p,\overline{v}^{(1)}_{\text{JAC-DD}})\\ &\quad +(d(p)+c^*)\zeta_1^{(1)}(p)\\ &= -\sum_{p'\in\sigma'(p)} e(p,p')z^{(2)}_{\text{GS-DD},1}(p') + \sum_{p'\in\sigma'(p)} e(p,p')\zeta_1^{(1)}(p')\\ &\quad +(c^*-f_v^{(2)}(p))\zeta_1^{(1)}(p), \quad p\in\Omega_1^h,\end{aligned}$$

where

$$f_v^{(2)}(p) = f_v[p,\overline{v}^{(1)}_{\text{JAC-DD}}(p) + \Theta_1^{(2)}(p)\zeta_1^{(1)}(p)], \quad 0<\Theta_1^{(2)}(p)<1.$$

Taking into account (1.16), (1.20), (5.10) and Theorem 9, we have

$$(d(p)+c^*)\zeta_1^{(2)}(p)\geq 0, \quad p\in\Omega_1^h,$$

$$\zeta_1^{(2)}(\gamma_1^{hr}) = \overline{v}^{(1)}_{\text{JAC-DD}}(\gamma_1^{hr}) - \overline{v}^{(1)}_{\text{GS-DD}}(\gamma_1^{hr}) \geq 0, \quad \zeta_1^{(2)}(\gamma_1^{hl}\cup\gamma_1^{h0}) = 0.$$

By the maximum principle in Lemma 1, we conclude

$$\zeta_1^{(2)}(p) = \overline{v}^{(2)}_{\text{JAC-DD},1}(p) - \overline{v}^{(2)}_{\text{GS-DD},1}(p) \geq 0, \quad p\in\overline{\Omega}_1^h. \tag{5.11}$$

From (5.2) for $n = 2$ and $m = 2$, it follows that

$$\begin{aligned}(\mathcal{L}+c^*)\zeta_2^{(2)}(p) &= (\mathcal{L}+c^*)\overline{v}^{(2)}_{\text{JAC-DD},2}(p) - (\mathcal{L}+c^*)\overline{v}^{(2)}_{\text{GS-DD},2}(p)\\ &= +(\mathcal{L}+c^*)z^{(2)}_{\text{JAC-DD},2}(p) - (\mathcal{L}+c^*)z^{(2)}_{\text{GS-DD},2}(p)\\ &\quad (\mathcal{L}+c^*)\overline{v}^{(1)}_{\text{JAC-DD}}(p) - (\mathcal{L}+c^*)\overline{v}^{(1)}_{\text{GS-DD}}(p)\\ &= -\mathcal{R}(p,\overline{v}^{(1)}_{\text{JAC-DD}}) + \mathcal{R}(p,\overline{v}^{(1)}_{\text{GS-DD}})\\ &\quad (\mathcal{L}+c^*)\overline{v}^{(1)}_{\text{JAC-DD}}(p) - (\mathcal{L}+c^*)\overline{v}^{(1)}_{\text{GS-DD}}(p).\end{aligned}$$

From here and by the mean value theorem, we have

$$\begin{aligned}(\mathcal{L}+c^*)\zeta_2^{(2)}(p) &= -\mathcal{L}\overline{v}^{(1)}_{\text{JAC-DD}}(p) - f(p,\overline{v}^{(1)}_{\text{JAC-DD}}) + \mathcal{L}\overline{v}^{(1)}_{\text{GS-DD}}(p) + f(p,\overline{v}^{(1)}_{\text{GS-DD}})\\ &\quad (\mathcal{L}+c^*)\overline{v}^{(1)}_{\text{JAC-DD}}(p) - (\mathcal{L}+c^*)\overline{v}^{(1)}_{\text{GS-DD}}(p)\\ &= -f(p,\overline{v}^{(1)}_{\text{JAC-DD}}) + f(p,\overline{v}^{(1)}_{\text{GS-DD}})\\ &\quad +c^*\overline{v}^{(1)}_{\text{JAC-DD}}(p) - c^*\overline{v}^{(1)}_{\text{GS-DD}}(p)\\ &= -\zeta_2^{(1)}(p)f_v^{(2)}(p) + c^*\zeta_2^{(1)}(p)\\ &= (c^* - f_v^{(2)}(p))\zeta_2^{(1)}(p), \quad p\in\Omega_2^h,\end{aligned}$$

where $f_v^{(2)}(p) = f_v(p,\overline{v}^{(1)}_{\text{JAC-DD},2} + \Theta_2^{(2)}\zeta_2^{(2)})$, $0<\Theta_2^{(2)}(p)<1$. From here, (1.20), (5.9) and (5.11), we have

$$(\mathcal{L}+c^*)\zeta_2^{(2)}(p)\geq 0,$$

$$\zeta_2^{(2)}(\gamma_2^{hl}) = \overline{v}^{(2)}_{\text{JAC-DD},1}(\gamma_2^{hl}) - \overline{v}^{(2)}_{\text{GS-DD},1}(\gamma_2^{hl})\geq 0, \quad \zeta_2^{(2)}(\gamma_2^{h0}) = 0,$$

$$\zeta_2^{(2)}(\gamma_2^{hr}) = \overline{v}^{(2)}_{\text{JAC-DD}}(\gamma_2^{hr}) - \overline{v}^{(2)}_{\text{GS-DD}}(\gamma_2^{hr})\geq 0.$$

By the maximum principle in Lemma 1, we conclude that

$$\zeta_2^{(2)}(p)\geq 0, \quad p\in\overline{\Omega}_2^h,$$

and by induction on m, $\zeta_m^{(2)}(p)\geq 0$, $p\in\overline{\Omega}_m^h$, $m=2,\ldots,M$. From here and (5.11), it follows that

$$\zeta_m^{(2)}(p) = \overline{v}^{(2)}_{\text{JAC-DD},m}(p) - \overline{v}^{(2)}_{\text{GS-DD},m}(p)\geq 0, \quad p\in\overline{\Omega}_m^h, \quad m=1,\ldots,M.$$

Using (5.3), we conclude that

$$\overline{v}^{(2)}_{\text{JAC-DD}}(p)\geq \overline{v}^{(2)}_{\text{GS-DD}}(p), \quad p\in\overline{\Omega}^h,$$

and by induction on n, we prove the theorem. □

5.3 Composite monotone domain decomposition algorithms based on the block Jacobi and block Gauss-Seidel methods

In this section, we are interested in the block Jacobi domain decomposition algorithm (BJAC-DD) and the block Gauss-Seidel algorithm (BGS-DD). The BJAC-DD algorithm and

the BGS-DD algorithm are combinations of the block Gauss-Seidel (BGS) method and the block Jacobi (BJAC) method, respectively, with the domain decomposition (DD) algorithm. A basic advantage of the BJAC method and the BGS method is that a simple tridiagonal solver such as the Thomas algorithm may be used for each of the linear subsystems defined on these subdomains.

As with the domain decomposition algorithm, we split the domain into a set of overlapping vertical strips $\overline{\Omega}_m$, $m = 1, \ldots, M$. One complete iterative step involves solving M problems on subdomains $\overline{\Omega}_m^h$, $m = 1, \ldots, M$, in serial. The first subdomain $\overline{\Omega}_1^h$ is solved using the BJAC method or the BGS method. The following subdomains $\overline{\Omega}_m^h$, $m = 2, \ldots, M$, are solved using the monotone domain decomposition algorithm. The algorithm for the monotone BJAC-DD and BGS-DD is as follows:

1. Initialisation: On the whole mesh $\overline{\Omega}^h$, choose an initial function $v^{(0)}(p)$, $p \in \overline{\Omega}^h$, satisfying the boundary condition $v^{(0)}(p) = g(p)$ on $\partial\Omega^h$.

2. On the first subdomain $\overline{\Omega}_1^h$, compute mesh function $z_1^{(n)}(p)$ by using either the BJAC method (4.14) or the BGS method (4.15), and denote

$$v_1^{(n)}(p) = v^{(n-1)}(p) + z_1^{(n)}(p), \quad p \in \overline{\Omega}_1^h. \tag{5.12}$$

3. On subdomains $\overline{\Omega}_m^h$, $m = 2, \ldots, M$, compute in serial mesh functions $v_m^{(n)}(p)$, $m = 2, \ldots, M$, satisfying the difference schemes

$$(\mathcal{L} + c^*) z_m^{(n)}(p) = -\mathcal{R}(p, v^{(n-1)}), \quad p \in \Omega_m^h, \tag{5.13}$$

$$z_m^{(n)}(p) = v_m^{(n)}(p) - v^{(n-1)}(p),$$

$$\mathcal{R}(p, v^{(n-1)}) = \mathcal{L} v^{(n-1)}(p) + f(p, v^{(n-1)}),$$

with the boundary conditions

$$z_m^{(n)}(p) = \begin{cases} z_{m-1}^{(n)}(p), & p \in \gamma_m^{hl}, \\ 0, & p \in \gamma_m^{hr} \cup \gamma_m^{h0}, \end{cases}$$

where

$$\gamma_m^{hl} = \gamma_m^l \cap \overline{\Omega}_m^h, \quad \gamma_m^{hr} = \gamma_m^r \cap \overline{\Omega}_m^h, \quad \gamma_m^{h0} = \gamma_m^0 \cap \overline{\Omega}_m^h.$$

4. Compute the solution $v^{(n)}(p)$, $p \in \overline{\Omega}^h$ by piecing together the solutions on the subdomains

$$v^{(n)}(p) = \begin{cases} v_m^{(n)}(p), & p \in \overline{\Omega}_m^h \setminus \theta_m^h,\ m = 1, \ldots, M-1; \\ v_M^{(n)}(p), & p \in \overline{\Omega}_M^h. \end{cases} \tag{5.14}$$

5. Stopping criterion: if

$$\max_{p \in \overline{\Omega}^h} |v^{(n)}(p) - v^{(n-1)}(p)| \leq \Delta,$$

where Δ is the required accuracy, then stop; otherwise continue iteration by going to Step 2.

The following theorem gives the convergence properties of the block Jacobi domain decomposition (BJAC-DD) algorithm.

Theorem 19. *Let $\overline{v}^{(0)}$, $\underline{v}^{(0)}$ be upper and lower solutions of (1.15). Then the upper and lower sequences $\{\overline{v}^{(n)}\}$ and $\{\underline{v}^{(n)}\}$ generated by the monotone* BJAC-DD *algorithm (5.12)–(5.14), converge monotonically from above and below, respectively, to the unique solution v of (1.15):*

$$\underline{v}^{(n-1)}(p) \leq \underline{v}^{(n)}(p) \leq v(p) \leq \overline{v}^{(n)}(p) \leq \overline{v}^{(n-1)}(p), \quad p \in \overline{\Omega}^h, \quad n \geq 1.$$

Proof. We consider only the case of upper sequences, as the case of lower sequences is proved in a similar manner.

Let $\overline{v}^{(0)}(p)$, $p \in \overline{\Omega}^h$ be an upper solution of (1.15) satisfying the boundary condition. From (5.12) with $n = 1$, $m = 1$ and using (4.16) from Theorem 11, we have

$$z_1^{(1)}(p) \leq 0, \quad p \in \overline{\Omega}_1^h. \tag{5.15}$$

From (5.13) with $n = 1$ and $m = 2$, we have

$$\begin{gathered} (\mathcal{L} + c^*) z_2^{(n)}(p) = -\mathcal{R}(p, \overline{v}^{(0)}) \leq 0, \quad p \in \Omega_2^h, \\ z_2^{(1)}(\gamma_2^{hl}) = z_1^{(1)}(\gamma_2^{hl}), \quad z_2^{(1)}(\gamma_2^{hr}) = 0, \quad z_2^{(1)}(\gamma_2^{h0}) = 0. \end{gathered}$$

From (5.15), we have $z_2^{(1)}(\gamma_2^{hl}) \leq 0$. By the maximum principle in Lemma 1, we conclude

$$z_2^{(1)}(p) \leq 0, \quad p \in \overline{\Omega}_2^h.$$

By induction on m, we conclude $z_m^{(1)}(p) \le 0$, $p \in \overline{\Omega}_m^h$, $m = 2, \ldots, M$. From here and (5.15), it follows that

$$z_m^{(1)}(p) \le 0, \quad p \in \overline{\Omega}_m^h, \quad m = 1, \ldots, M. \tag{5.16}$$

We now prove that $v^{(1)}(p)$ defined in Step 4 is an upper solution, that is,

$$\mathcal{R}(p, v^{(1)}) \ge 0, \quad p \in \Omega. \tag{5.17}$$

From (5.13) and the mean value theorem,

$$\begin{aligned}
\mathcal{R}(p, v_m^{(1)}) &= \mathcal{R}(p, \overline{v}^{(0)} + z_m^{(1)}) \\
&= \mathcal{L}\overline{v}^{(0)}(p) + \mathcal{L}z_m^{(1)}(p) + f(p, \overline{v}^{(0)} + z_m^{(1)}) \\
&= \mathcal{L}\overline{v}^{(0)}(p) + f(p, \overline{v}^{(0)}) + \mathcal{L}z_m^{(1)}(p) + f(p, \overline{v}^{(0)} + z_m^{(1)}) - f(p, \overline{v}^{(0)}) \\
&= \mathcal{R}(p, \overline{v}^{(0)}) + \mathcal{L}z_m^{(1)}(p) + f_v^{(1)}(p) z_m^{(1)}(p) \\
&= -(\mathcal{L} + c^*) z_m^{(1)}(p) + \mathcal{L}z_m^{(1)}(p) + f_v^{(1)}(p) z_m^{(1)}(p) \\
&= -(c^* - f_v^{(1)}(p)) z_m^{(1)}(p), \quad p \in \Omega_m^h, \quad m = 1, \ldots, M,
\end{aligned}$$

where $f_v^{(1)}(p) = f_v[p, \overline{v}^{(0)}(p) + \Theta_m^{(1)}(p) z_m^{(1)}(p)]$, $0 < \Theta_m^{(1)}(p) < 1$. From here, (5.16) and (1.20), we have

$$\mathcal{R}(p, v_m^{(1)}) = -(c^* - f_v^{(1)}(p)) z_m^{(1)} \ge 0 \quad p \in \Omega_m^h, \quad m = 1, \ldots, M.$$

From here and (5.14), it follows that

$$\mathcal{R}(p, v^{(1)}) \ge 0, \quad p \in \Omega^h \setminus \gamma^{hl}, \quad \gamma^{hl} = \bigcup_{m=2}^{M} \gamma_m^{hl}. \tag{5.18}$$

We therefore only need to verify (5.17) on the boundaries $\gamma_m^{hl}, m = 2, \ldots, M$. Let

$$\zeta_m^{(n)}(p) = v_m^{(n)}(p) - v_{m+1}^{(n)}(p) = z_m^{(n)}(p) - z_{m+1}^{(n)}, \quad p \in \overline{\theta}_m^h.$$

Using (4.14) with $\omega = 1$ and (5.13) with $m = 2$, we obtain

$$\begin{aligned}
(A_i + c^* I)\zeta_{1,i}^{(1)} &= (A_i + c^* I) Z_{1,i}^{(1)} - (A_i + c^* I) Z_{2,i}^{(1)} \\
&= -\mathcal{R}_i(V^{(0)}) - W_i Z_{2,i-1}^{(1)} - E_i Z_{2,i+1}^{(1)} + \mathcal{R}_i(V^{(0)}) \\
&= -W_i Z_{2,i-1}^{(1)} - E_i Z_{2,i+1}^{(1)}, \quad i = \widehat{i}_1 + 1, \ldots, i_1 - 1,
\end{aligned} \tag{5.19}$$

$$\zeta_1^{(1)} = Z_1^{(1)} - Z_2^{(1)}, \quad Z_m^{(1)} = \{Z_{m,i}, i = \widehat{i}_1, \ldots, i_1\}, \quad m = 1, 2,$$

where $i = \widehat{i}_1$ and $i = i_1$ correspond to the left γ_2^{hl} and right γ_1^{hr} boundaries of $\overline{\theta}_1^h$, respectively. From here, (4.12) and (5.16), we have

$$(A_i + c^* I)\zeta_{1,i}^{(1)} \geq 0, \quad i = \widehat{i}_1 + 1, \ldots, i_1 - 1,$$

and using (4.17), it follows that

$$\zeta_1(p) \geq 0, \quad p \in \theta_1^h.$$

As $\zeta_1^{(1)}(\gamma_2^{hl}) = 0$, $\zeta_1^{(1)}(\gamma_1^{hr}) = -z_2^{(1)}(\gamma_1^{hr}) \geq 0$, we conclude

$$\zeta_1(p) \geq 0, \quad p \in \overline{\theta}_1^h. \tag{5.20}$$

From (5.13) and (5.16), we have

$$(\mathcal{L} + c^*)\zeta_m^{(1)} = (\mathcal{L} + c^*)z_m^{(1)} - (\mathcal{L} + c^*)z_{m+1}^{(1)} = 0, \quad p \in \theta_m^h, \quad m = 2, \ldots, M.$$

$$\zeta_m^{(1)} = 0, \quad p \in \partial\theta_m^h \setminus \gamma_m^{hr}, \quad \zeta_m^{(1)} = -z_{m+1}^{(1)} \geq 0, \quad p \in \gamma_m^{hr}.$$

From here and by the maximum principle in Lemma 1, we get

$$\zeta_m^{(1)} \geq 0, \quad p \in \overline{\theta}_m^h, \quad m = 2, \ldots, M-1,$$

and using (5.20), it follows that

$$\zeta_m^{(1)} = v_m^{(1)} - v_{m+1}^{(1)} \geq 0, \quad p \in \overline{\theta}_m^h, \quad m = 1, \ldots, M-1. \tag{5.21}$$

Let $\tilde{i}$ correspond to γ_m^{hl}, then

$$\begin{aligned} \mathcal{R}_{\tilde{i}}(V^{(1)}) &= A_{\tilde{i}}V_{\tilde{i}}^{(1)} - W_{\tilde{i}}V_{\tilde{i}-1}^{(1)} - E_{\tilde{i}}V_{\tilde{i}+1}^{(1)} + F_{\tilde{i}}(V_{\tilde{i}}^{(1)}) + G_{\tilde{i}}^* \\ &= A_{\tilde{i}}V_{m-1,\tilde{i}}^{(1)} - W_{\tilde{i}}V_{m-1,\tilde{i}-1}^{(1)} - E_{\tilde{i}}V_{m,\tilde{i}+1}^{(1)} + F_{\tilde{i}}(V_{m-1,\tilde{i}}^{(1)}) + G_{\tilde{i}}^*. \end{aligned}$$

From here and using (5.21), we have

$$\begin{aligned} \mathcal{R}_{\tilde{i}}(V^{(1)}) &\geq A_{\tilde{i}}V_{m-1,\tilde{i}}^{(1)} - W_{\tilde{i}}V_{m-1,\tilde{i}-1}^{(1)} - E_{\tilde{i}}V_{m-1,\tilde{i}+1}^{(1)} + F_{\tilde{i}}(V_{m-1,\tilde{i}}^{(1)}) + G_{\tilde{i}}^* \\ &= \mathcal{R}_i(V_{m-1}^{(1)}) \geq 0, \quad m = 2, \ldots, M, \end{aligned}$$

and using (5.18), it follows that

$$\mathcal{R}(p, v^{(1)}) \geq 0, \quad p \in \Omega^h,$$

and, hence, $v^{(1)}$ is an upper solution to (1.15). By induction on n, we are able to conclude that $\{\overline{v}^{(n)}\}$ is a monotonically decreasing sequence of upper solutions. By (1.19), this sequence is bounded from below by $\underline{v}$, where $\underline{v}$ is any lower solution. Therefore the sequence converges. Let $v(p)$ be defined by

$$v(p) = \lim_{n\to\infty} \overline{v}^{(n)}(p), \quad p \in \overline{\Omega}^h.$$

We now prove that $v(p)$ is the solution to (1.15). From (5.12)–(5.14), we have

$$\lim_{n\to\infty} z_m^{(n)} = \lim_{n\to\infty} (v_m^{(n)} - v_m^{(n-1)}) = 0, \quad m = 1, \ldots, M,$$

and from here and (4.14),

$$\lim_{n\to\infty} \mathcal{R}_i(V^{(n)}) = -\lim_{n\to\infty} ((A_i + c^* I) Z_i^{(n+1)}) = 0, \quad i = 1, \ldots, \widehat{i}_1.$$

Thus,

$$\mathcal{R}(p, v) = \lim_{n\to\infty} \mathcal{R}(p, v^{(n)}) = \lim_{n\to\infty} \mathcal{R}(p, v_1^{(n)}) = 0, \quad p \in \Omega_1^h \setminus \theta_1^h.$$

Using (5.13) and (5.14), we have

$$\begin{aligned} \mathcal{R}(p, v) \;\; = \;\; & \lim_{n\to\infty} \mathcal{R}(p, v^{(n)}) = \lim_{n\to\infty} \mathcal{R}(p, v_m^{(n)}) = -\lim_{n\to\infty} (\mathcal{L} + c^*) z_m^{(n)}(p) = 0, \\ & p \in \Omega_m^h \setminus \theta_m^h, \quad m = 2, \ldots, M-1, \quad p \in \Omega_M^h. \end{aligned}$$

Thus,

$$\mathcal{R}(p, v) = 0, \quad p \in \Omega^h.$$

From (5.12)–(5.14), we have $v(p) = g(p)$, $p \in \partial\Omega^h$, and conclude that $v(p)$ is the solution of (1.15). □

The following theorem gives a comparison result on convergence of the monotone sequences obtained by the JAC-DD algorithm (5.1)–(5.3) and the monotone BJAC-DD algorithm (5.12)–(5.14).

Theorem 20. *Let* $\{\overline{v}_{\text{JAC-DD}}^{(n)}\}$, $\{\underline{v}_{\text{JAC-DD}}^{(n)}\}$ *and* $\{\overline{v}_{\text{BJAC-DD}}^{(n)}\}$, $\{\underline{v}_{\text{BJAC-DD}}^{(n)}\}$ *be the sequences of upper and lower solutions generated by the* JAC-DD *algorithm (5.1)–(5.3) and the* BJAC-DD *algorithm (5.12)–(5.14), respectively, where* $\overline{v}_{\text{JAC-DD}}^{(0)} = \overline{v}_{\text{BJAC-DD}}^{(0)} = \overline{v}^{(0)}$ *and* $\underline{v}_{\text{JAC-DD}}^{(0)} = \underline{v}_{\text{BJAC-DD}}^{(0)} = \underline{v}^{(0)}$. *Then*

$$\underline{v}_{\text{JAC-DD}}^{(n)}(p) \le \underline{v}_{\text{BJAC-DD}}^{(n)}(p) \le \overline{v}_{\text{BJAC-DD}}^{(n)}(p) \le \overline{v}_{\text{JAC-DD}}^{(n)}(p), \quad p \in \overline{\Omega}^h, \quad n \ge 1.$$

Proof. This theorem is proved in a similar manner as Theorem 18. The full proof can be found in Appendix B. □

The following theorem gives the convergence property of the monotone block Gauss-Seidel domain decomposition (GS-DD) algorithm.

Theorem 21. *Let $\overline{v}^{(0)}$, $\underline{v}^{(0)}$ be upper and lower solutions of (1.15). Then the upper and lower sequences $\{\overline{v}^{(n)}\}$ and $\{\underline{v}^{(n)}\}$ generated by the monotone* BGS-DD *algorithm (5.12)–(5.14), converge monotonically from above and below, respectively, to the unique solution v of (1.15):*

$$\underline{v}^{(n-1)}(p) \leq \underline{v}^{(n)}(p) \leq v(p) \leq \overline{v}^{(n)}(p) \leq \overline{v}^{(n-1)}(p), \quad p \in \overline{\Omega}^h, \quad n \geq 1.$$

Proof. This theorem is proved in a similar manner as Theorem 19. The full proof can be found in Appendix B. □

The following theorem gives a comparison result on convergence of the monotone sequences obtained by the GS-DD algorithm (5.1)–(5.3) and the monotone BGS-DD algorithm (5.12)–(5.14).

Theorem 22. *Let $\{\overline{v}^{(n)}_{\text{GS-DD}}\}$, $\{\underline{v}^{(n)}_{\text{GS-DD}}\}$ and $\{\overline{v}^{(n)}_{\text{BGS-DD}}\}$, $\{\underline{v}^{(n)}_{\text{BGS-DD}}\}$ be the sequences of upper and lower solutions generated by the* GS-DD *algorithm (5.1)–(5.3) and the* BGS-DD *algorithm (5.12)–(5.14), respectively, where $\overline{v}^{(0)}_{\text{GS-DD}} = \overline{v}^{(0)}_{\text{BGS-DD}} = \overline{v}^{(0)}$ and $\underline{v}^{(0)}_{\text{GS-DD}} = \underline{v}^{(0)}_{\text{BGS-DD}} = \underline{v}^{(0)}$. Then*

$$\underline{v}^{(n)}_{\text{GS-DD}}(p) \leq \underline{v}^{(n)}_{\text{BGS-DD}}(p) \leq \overline{v}^{(n)}_{\text{BGS-DD}}(p) \leq \overline{v}^{(n)}_{\text{GS-DD}}(p), \quad p \in \overline{\Omega}^h, \quad n \geq 1.$$

Proof. This theorem is proved in a similar manner as Theorem 18. The full proof can be found in Appendix B. □

The following theorem gives a comparison result on convergence of the monotone sequences obtained by the monotone BJAC-DD algorithm (5.12)–(5.14) and the monotone BGS-DD algorithm (5.12)–(5.14).

Theorem 23. *Let $\{\overline{v}^{(n)}_{\text{BJAC-DD}}\}$, $\{\underline{v}^{(n)}_{\text{BJAC-DD}}\}$ and $\{\overline{v}^{(n)}_{\text{BGS-DD}}\}$, $\{\underline{v}^{(n)}_{\text{BGS-DD}}\}$ be the sequences of upper and lower solutions generated by the* BJAC-DD *algorithm (5.12)–(5.14) and the* BGS-DD *algorithm (5.12)–(5.14), respectively, where $\overline{v}^{(0)}_{\text{BJAC-DD}} = \overline{v}^{(0)}_{\text{BGS-DD}} = \overline{v}^{(0)}$ and $\underline{v}^{(0)}_{\text{BJAC-DD}} = \underline{v}^{(0)}_{\text{BGS-DD}} = \underline{v}^{(0)}$. Then*

$$\underline{v}^{(n)}_{\text{BJAC-DD}}(p) \leq \underline{v}^{(n)}_{\text{BGS-DD}}(p) \leq \overline{v}^{(n)}_{\text{BGS-DD}}(p) \leq \overline{v}^{(n)}_{\text{BJAC-DD}}(p), \quad p \in \overline{\Omega}^h, \quad n \geq 1.$$

Proof. This theorem is proved in a similar manner as Theorem 18. The full proof can be found in Appendix B. □

5.4 Composite monotone two-level PDD and BDD algorithms

Here, we consider the composite montone two-level PDD and BDD algorithms, where Step 3 in the algorithms (5.1)–(5.3) and (5.12)–(5.14) are solved by the two-level monotone domain decomposition algorithm (3.10), (3.11), (3.12)–(3.14).

Theorem 24. *Let $\overline{v}^{(0)}, \underline{v}^{(0)}$ be upper and lower solutions of (1.15). Then the upper and lower sequences $\{\overline{v}^{(n)}\}$ and $\{\underline{v}^{(n)}\}$ generated by the either the monotone* PDD *method (5.1)–(5.3) or the* BDD *method (5.12)–(5.14) and where Step 3 is solved using (3.10), (3.11), (3.12)–(3.14), converge monotonically from above and below, respectively, to the unique solution v of (1.15):*

$$\underline{v}^{(n-1)}(p) \leq \underline{v}^{(n)}(p) \leq v(p) \leq \overline{v}^{(n)}(p) \leq \overline{v}^{(n-1)}(p), \quad p \in \overline{\Omega}^h, \quad n \geq 1.$$

Proof. The proof of this theorem follows directly from Theorems 6, 16, 17, 19 and 21. □

Remark 12. An advantage of using either the ω-Jacobi or the BJAC method in Step 2 is that this method is easy parallelisable as each line (block) may be solved independently from the others. By using the (3.12)–(3.14) to find the solution in Step 3, we are able to solve this step in parallel. Each of the box-subdomain problems on $\overline{\Omega}_{ms}$, $s = 1, \ldots, S_m$, can be solved on their own processor in parallel. Each of the interfacial subdomain problems on $\overline{\vartheta}^h_{ms}$, $s = 1, \ldots, S_m - 1$, can be solved on their own processors in parallel. Constructing algorithms for use on parallel computers aids in the reduction of problems caused by processor time and computer memory.

5.5 Numerical experiments

As the theoretical results indicate that the composite block Gauss-Seidel domain decomposition (BDD) algorithm results in the fastest convergence, we apply this algorithm to two test problems. We compare these results with the numerical results obtained from the monotone domain decomposition (DD) algorithm in Section 3.2.1 and the monotone

block SUR (BSUR) method in Section 4.5. The number of mesh points in the x- and y-directions are set equal to N. In our numerical experiments, the required accuracy for the stopping criteria is $\Delta = 10^{-6}$.

To solve the linear difference problems within the monotone BDD algorithm, GMRES solver with restarts is used with a diagonal preconditioner. Within GMRES, the required accuracy used is 10^{-6}, the maximum numbers of iterations and restarts are 50 and 20, respectively.

The domain is decomposed into M overlapping subdomains as described in Section 3.2.1. The overlap between the vertical strips is chosen so that for the two vertical strips either side of the boundary layer, the overlap occurs outside of the boundary layer, as described in Section 3.2.1.

We define the serial acceleration of the domain decomposition algorithm as the execution time of the undecomposed method / the minimum execution time of the domain decomposition algorithm.

5.5.1 Convection-diffusion problem with parabolic boundary layers

Consider the test problem (3.6). This problem is the convection-diffusion problem with parabolic boundary layers (1.9), which is characterised by an elliptic boundary layer close to $x = 1$ and by parabolic boundary layers close to $y = 0$ and $y = 1$. To solve this problem numerically, we use the two dimensional piecewise uniform Shishkin mesh (1.11) constructed in Section 1.4.2. We also require the constants c_*, c^* from (1.20) and an initial solution. From (3.7), we have

$$c_* = \exp(-1), \quad c^* = 1.$$

From (1.6) and (1.7), we can see that $\overline{u}^{(0)}(p) = 1$, $p \in \overline{\Omega}^h$ is an initial upper solution.

From Section 4.5.1, it is observed that the monotone BSUR method has the smallest iteration counts and fastest execution times when $\omega = 1$, leading to the conclusion that $\omega = 1$ is optimal. All of the following numerical experiments have $\omega = 1$. Using the execution times from the numerical results in Section 4.5.1, the serial acceleration for the monotone BSUR method is calculated and displayed in Figure 5.2, where the serial acceleration is the execution time of the undecomposed method ($M = 1$) / the execution time of the BSUR method. From Figure 5.2, we can see that for all values of N and ε

the serial acceleration of the BSUR method is significantly greater than one, indicating a very significant advantage in the monotone BSUR method. We can also see from Figure 5.2 that as ε decreases the serial acceleration increases.

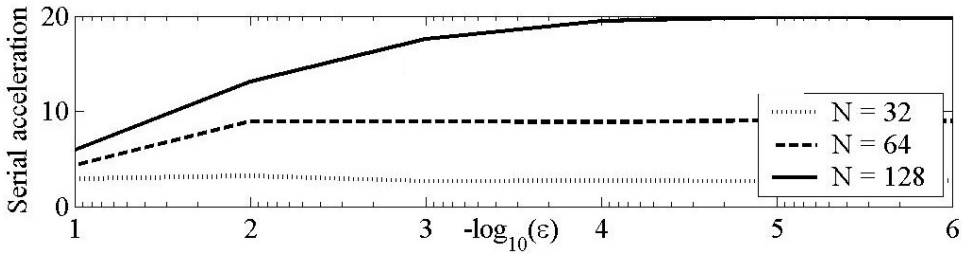

Figure 5.2: Serial acceleration of the monotone BSUR method for the test problem (3.6).

To investigate the serial acceleration of the monotone BDD algorithm, we begin by looking at the effect of varying the size of the first subdomain, which is the subdomain solved using the monotone BSUR method. Tables 5.1, 5.2 and 5.3 display the iteration counts and execution times of the monotone BDD algorithm for $N = 32$, $N = 64$ and $N = 128$, respectively. N_1 is the number of mesh points in the x-direction in the first subdomain. The minimum execution times for each value of the parameter ε are displayed in bold. From these tables, we can see that for $\varepsilon < 10^{-2}$ the execution times are minimal when the first subdomain contains a quarter of the mesh points in the x-direction. From the tables, it can also be seen that for $N = 32, 64, 128$ and for $\varepsilon < 10^{-2}$, the minimal execution times occur when the number of subdomains are $5, 9$ and 9, respectively. For fixed values of N_1 and M, the iteration counts are the same for $\varepsilon < 10^{-5}$. From this, we can conclude that the monotone BDD algorithm is parameter uniformly convergent in its iteration counts.

Figure 5.3 displays the serial acceleration of the monotone BDD algorithm. From Figure 5.3, we can see that for all values of N and ε the serial acceleration is greater than one indicating an advantage in the monotone BDD algorithm. It can also been seen that as N increases the serial acceleration increases, and as ε decreases the serial acceleration increases.

We now compare the monotone DD algorithm, the monotone BSUR method and the monotone BDD algorithm. Figure 5.4 displays the serial accelerations for all the three algorithms. From this figure, it follows that for $\varepsilon > 10^{-3}$ the monotone BSUR method results in the highest serial acceleration. For $\varepsilon \leq 10^{-3}$, the monotone BDD algorithm

N_1	$N/4$				$N/2$				$3N/4$		
ε/M	2	3	5	9	2	3	5	9	2	3	5
	Iteration counts										
10^{-1}	108	$\frac{109}{109}$	$\frac{111}{110}$	$\frac{117}{117}$	190	$\frac{190}{190}$	$\frac{193}{192}$	$\frac{195}{195}$	239	$\frac{240}{240}$	$\frac{240}{240}$
10^{-2}	17	$\frac{24}{17}$	$\frac{29}{18}$	$\frac{42}{42}$	54	$\frac{57}{55}$	$\frac{69}{64}$	$\frac{82}{82}$	200	$\frac{202}{202}$	$\frac{203}{203}$
10^{-3}	10	$\frac{25}{10}$	$\frac{30}{11}$	$\frac{44}{44}$	58	$\frac{62}{60}$	$\frac{75}{70}$	$\frac{89}{89}$	219	$\frac{221}{221}$	$\frac{221}{221}$
10^{-4}	9	$\frac{25}{9}$	$\frac{30}{11}$	$\frac{44}{44}$	59	$\frac{63}{60}$	$\frac{76}{71}$	$\frac{90}{90}$	222	$\frac{223}{223}$	$\frac{224}{224}$
10^{-5}	9	$\frac{25}{9}$	$\frac{30}{11}$	$\frac{44}{44}$	59	$\frac{63}{60}$	$\frac{76}{71}$	$\frac{90}{90}$	222	$\frac{224}{223}$	$\frac{224}{224}$
10^{-6}	9	$\frac{25}{9}$	$\frac{30}{11}$	$\frac{44}{44}$	59	$\frac{63}{60}$	$\frac{76}{71}$	$\frac{90}{90}$	222	$\frac{224}{223}$	$\frac{224}{224}$
	Execution times (seconds)										
10^{-1}	2.40	$\frac{1.10}{2.50}$	$\frac{0.67}{1.19}$	$\frac{0.55}{0.55}$	1.56	$\frac{0.87}{1.64}$	$\frac{0.64}{0.84}$	$\frac{0.66}{0.66}$	0.46	$\frac{\mathbf{0.39}}{0.48}$	$\frac{0.44}{0.44}$
10^{-2}	0.36	$\frac{0.19}{0.41}$	$\frac{\mathbf{0.14}}{0.20}$	$\frac{0.17}{0.17}$	0.37	$\frac{0.19}{0.38}$	$\frac{0.16}{0.19}$	$\frac{0.23}{0.23}$	0.30	$\frac{0.26}{0.31}$	$\frac{0.32}{0.32}$
10^{-3}	0.18	$\frac{0.14}{0.20}$	$\frac{0.11}{\mathbf{0.09}}$	$\frac{0.16}{0.16}$	0.31	$\frac{0.18}{0.34}$	$\frac{0.17}{0.19}$	$\frac{0.24}{0.24}$	0.29	$\frac{0.27}{0.31}$	$\frac{0.34}{0.34}$
10^{-4}	0.17	$\frac{0.12}{0.18}$	$\frac{0.11}{\mathbf{0.09}}$	$\frac{0.16}{0.16}$	0.24	$\frac{0.16}{0.27}$	$\frac{0.17}{0.18}$	$\frac{0.24}{0.24}$	0.27	$\frac{0.27}{0.30}$	$\frac{0.34}{0.34}$
10^{-5}	0.17	$\frac{0.10}{0.18}$	$\frac{0.10}{\mathbf{0.09}}$	$\frac{0.14}{0.14}$	0.17	$\frac{0.11}{0.18}$	$\frac{0.13}{0.15}$	$\frac{0.22}{0.22}$	0.22	$\frac{0.23}{0.26}$	$\frac{0.32}{0.32}$
10^{-6}	0.17	$\frac{0.09}{0.18}$	$\frac{0.09}{\mathbf{0.08}}$	$\frac{0.14}{0.14}$	0.15	$\frac{0.10}{0.16}$	$\frac{0.12}{0.13}$	$\frac{0.21}{0.21}$	0.19	$\frac{0.22}{0.23}$	$\frac{0.31}{0.31}$

Table 5.1: Iteration counts and execution times of the monotone BDD algorithm for the test problem (3.6), using the minimal and maximal overlap size above and below the line, respectively, for $N = 32$. N_1 is the number of mesh points in the x-direction, where the monotone BSUR method is in use.

results in the highest serial acceleration.

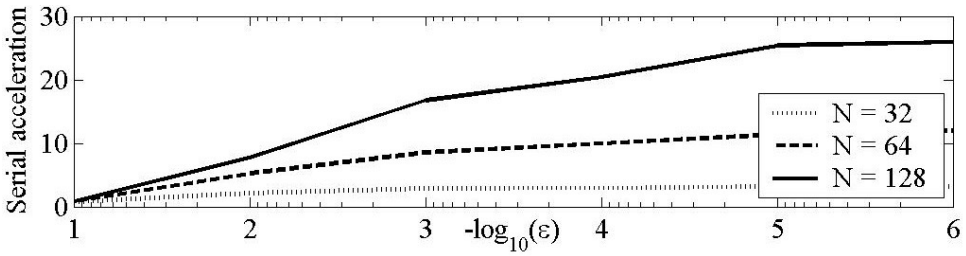

Figure 5.3: Serial acceleration of the monotone BDD algorithm for the test problem (3.6).

N_1	$N/4$					$N/2$					$3N/4$			
ε/M	2	3	5	9	17	2	3	5	9	17	2	3	5	9
	Iteration counts													
10^{-1}	329	$\frac{329}{329}$	$\frac{331}{330}$	$\frac{338}{334}$	$\frac{356}{356}$	598	$\frac{599}{599}$	$\frac{601}{600}$	$\frac{608}{605}$	$\frac{616}{616}$	767	$\frac{768}{768}$	$\frac{771}{770}$	$\frac{774}{774}$
10^{-2}	32	$\frac{38}{32}$	$\frac{42}{32}$	$\frac{70}{35}$	$\frac{115}{115}$	83	$\frac{86}{83}$	$\frac{98}{87}$	$\frac{129}{112}$	$\frac{181}{181}$	481	$\frac{482}{481}$	$\frac{486}{485}$	$\frac{490}{490}$
10^{-3}	11	$\frac{33}{11}$	$\frac{40}{11}$	$\frac{76}{27}$	$\frac{126}{126}$	89	$\frac{92}{90}$	$\frac{106}{94}$	$\frac{143}{125}$	$\frac{198}{198}$	537	$\frac{538}{537}$	$\frac{542}{541}$	$\frac{546}{546}$
10^{-4}	9	$\frac{34}{9}$	$\frac{40}{9}$	$\frac{78}{27}$	$\frac{128}{128}$	91	$\frac{94}{91}$	$\frac{108}{96}$	$\frac{146}{127}$	$\frac{201}{201}$	546	$\frac{547}{547}$	$\frac{551}{550}$	$\frac{555}{555}$
10^{-5}	9	$\frac{34}{9}$	$\frac{40}{9}$	$\frac{78}{27}$	$\frac{128}{128}$	91	$\frac{95}{92}$	$\frac{109}{97}$	$\frac{146}{128}$	$\frac{201}{201}$	547	$\frac{548}{548}$	$\frac{552}{551}$	$\frac{556}{556}$
10^{-6}	9	$\frac{34}{9}$	$\frac{40}{9}$	$\frac{78}{27}$	$\frac{128}{128}$	91	$\frac{95}{92}$	$\frac{109}{97}$	$\frac{146}{128}$	$\frac{201}{201}$	547	$\frac{548}{548}$	$\frac{552}{551}$	$\frac{556}{556}$
	Execution times (seconds)													
10^{-1}	106.0	$\frac{37.1}{112.8}$	$\frac{18.2}{51.4}$	$\frac{9.1}{19.2}$	$\frac{8.2}{8.2}$	48.9	$\frac{29.0}{58.2}$	$\frac{13.4}{33.8}$	$\frac{9.4}{12.0}$	$\frac{11.0}{11.0}$	16.7	$\frac{8.1}{17.0}$	$\frac{\mathbf{6.2}}{7.7}$	$\frac{7.4}{7.4}$
10^{-2}	12.9	$\frac{3.4}{10.9}$	$\frac{1.8}{4.1}$	$\frac{1.5}{1.6}$	$\frac{2.7}{2.7}$	6.1	$\frac{2.9}{7.1}$	$\frac{\mathbf{1.5}}{3.3}$	$\frac{1.6}{1.7}$	$\frac{3.1}{3.1}$	7.7	$\frac{3.8}{7.8}$	$\frac{3.4}{4.0}$	$\frac{4.5}{4.6}$
10^{-3}	2.4	$\frac{2.0}{2.3}$	$\frac{1.2}{1.1}$	$\frac{1.4}{\mathbf{1.0}}$	$\frac{2.6}{2.6}$	5.6	$\frac{2.6}{6.4}$	$\frac{1.5}{3.1}$	$\frac{1.7}{1.9}$	$\frac{3.3}{3.3}$	7.0	$\frac{3.9}{7.5}$	$\frac{3.6}{4.2}$	$\frac{4.9}{4.9}$
10^{-4}	2.1	$\frac{1.7}{2.1}$	$\frac{1.1}{\mathbf{0.8}}$	$\frac{1.4}{0.9}$	$\frac{2.6}{2.6}$	4.5	$\frac{2.0}{4.9}$	$\frac{1.4}{2.4}$	$\frac{1.8}{1.9}$	$\frac{3.3}{3.3}$	5.5	$\frac{3.6}{6.0}$	$\frac{3.7}{4.2}$	$\frac{5.0}{5.0}$
10^{-5}	2.1	$\frac{1.3}{2.0}$	$\frac{0.9}{0.8}$	$\frac{1.1}{\mathbf{0.8}}$	$\frac{2.4}{2.4}$	3.0	$\frac{1.3}{3.2}$	$\frac{1.0}{1.6}$	$\frac{1.4}{1.4}$	$\frac{3.1}{3.1}$	3.7	$\frac{2.8}{4.2}$	$\frac{3.1}{3.5}$	$\frac{4.7}{4.7}$
10^{-6}	2.1	$\frac{1.2}{2.0}$	$\frac{0.9}{0.8}$	$\frac{1.1}{\mathbf{0.7}}$	$\frac{2.3}{2.3}$	2.6	$\frac{1.2}{2.9}$	$\frac{0.9}{1.5}$	$\frac{1.4}{1.3}$	$\frac{3.0}{3.0}$	3.4	$\frac{2.5}{3.9}$	$\frac{3.0}{3.2}$	$\frac{4.6}{4.6}$

Table 5.2: Iteration counts and execution times of the monotone BDD algorithm for the test problem (3.6), using the minimal and maximal overlap size above and below the line, respectively, for $N = 64$. N_1 is the number of mesh points in the x-direction, where the monotone BSUR method is in use.

5.5.2 Anisotropic problem

Consider the test problem (3.8). This problem is the anisotropic convection-diffusion problem (1.12) and is characterised by an elliptic boundary layer close to $x = 1$. To solve this problem numerically, we use the two dimensional piecewise uniform Shishkin mesh (1.8) constructed in Section 1.4.3. We also require the constants c_*, c^* from (1.20) and an initial solution. From (3.9), we have

$$c_* = \exp(-1), \quad c^* = 1.$$

From (1.6) and (1.7), we can see that $\overline{u}^{(0)}(p) = 1$, $p \in \overline{\Omega}^h$ is an upper solution.

N_1	$N/4$						$N/2$						$3N/4$				
ε/M	2	3	5	9	17	33	2	3	5	9	17	33	2	3	5	9	17
	Iteration counts																
10^{-1}	1035	$\frac{1035}{1035}$	$\frac{1037}{1035}$	$\frac{1046}{1037}$	$\frac{1066}{1048}$	$\frac{1122}{1122}$	1922	$\frac{1923}{1922}$	$\frac{1926}{1923}$	$\frac{1935}{1926}$	$\frac{1954}{1944}$	$\frac{1987}{1987}$	2504	$\frac{2505}{2504}$	$\frac{2509}{2506}$	$\frac{2518}{2514}$	$\frac{2531}{2531}$
10^{-2}	77	$\frac{82}{77}$	$\frac{87}{77}$	$\frac{111}{77}$	$\frac{169}{91}$	$\frac{297}{297}$	151	$\frac{155}{151}$	$\frac{167}{152}$	$\frac{203}{162}$	$\frac{292}{235}$	$\frac{463}{463}$	1231	$\frac{1232}{1231}$	$\frac{1236}{1233}$	$\frac{1247}{1241}$	$\frac{1265}{1265}$
10^{-3}	16	$\frac{49}{16}$	$\frac{58}{16}$	$\frac{107}{21}$	$\frac{176}{69}$	$\frac{313}{313}$	139	$\frac{144}{140}$	$\frac{160}{142}$	$\frac{206}{157}$	$\frac{311}{249}$	$\frac{494}{494}$	1383	$\frac{1384}{1383}$	$\frac{1388}{1385}$	$\frac{1398}{1393}$	$\frac{1415}{1415}$
10^{-4}	9	$\frac{49}{9}$	$\frac{58}{10}$	$\frac{109}{20}$	$\frac{179}{69}$	$\frac{317}{317}$	144	$\frac{149}{145}$	$\frac{165}{147}$	$\frac{212}{163}$	$\frac{320}{257}$	$\frac{505}{505}$	1419	$\frac{1420}{1419}$	$\frac{1423}{1421}$	$\frac{1434}{1429}$	$\frac{1451}{1451}$
10^{-5}	9	$\frac{49}{9}$	$\frac{58}{9}$	$\frac{109}{20}$	$\frac{179}{69}$	$\frac{318}{318}$	145	$\frac{150}{145}$	$\frac{166}{148}$	$\frac{213}{164}$	$\frac{321}{258}$	$\frac{506}{506}$	1423	$\frac{1424}{1423}$	$\frac{1427}{1425}$	$\frac{1438}{1433}$	$\frac{1455}{1455}$
10^{-6}	9	$\frac{49}{9}$	$\frac{58}{9}$	$\frac{109}{20}$	$\frac{179}{69}$	$\frac{318}{318}$	145	$\frac{150}{145}$	$\frac{166}{148}$	$\frac{213}{164}$	$\frac{321}{258}$	$\frac{506}{506}$	1423	$\frac{1424}{1424}$	$\frac{1428}{1425}$	$\frac{1438}{1433}$	$\frac{1455}{1455}$
	Execution times (seconds)																
10^{-1}	10086	$\frac{3767}{11916}$	$\frac{762}{5949}$	$\frac{246}{1039}$	$\frac{137}{265}$	$\frac{152}{152}$	6488	$\frac{1417}{6914}$	$\frac{439}{1660}$	$\frac{192}{514}$	$\frac{163}{200}$	$\frac{234}{233}$	867	$\frac{274}{811}$	$\frac{128}{310}$	$\frac{\mathbf{110}}{134}$	$\frac{156}{156}$
10^{-2}	839	$\frac{252}{971}$	$\frac{51}{280}$	$\frac{22}{60}$	$\frac{23}{22}$	$\frac{45}{45}$	505	$\frac{96}{527}$	$\frac{26}{110}$	$\frac{\mathbf{16}}{29}$	$\frac{22}{21}$	$\frac{54}{54}$	374	$\frac{95}{336}$	$\frac{50}{108}$	$\frac{51}{58}$	$\frac{78}{78}$
10^{-3}	193	$\frac{75}{228}$	$\frac{22}{39}$	$\frac{13}{\mathbf{10}}$	$\frac{17}{11}$	$\frac{41}{41}$	453	$\frac{70}{471}$	$\frac{17}{79}$	$\frac{13}{21}$	$\frac{22}{20}$	$\frac{56}{56}$	330	$\frac{78}{290}$	$\frac{47}{93}$	$\frac{54}{60}$	$\frac{84}{84}$
10^{-4}	121	$\frac{68}{128}$	$\frac{21}{26}$	$\frac{13}{\mathbf{9}}$	$\frac{19}{12}$	$\frac{42}{42}$	468	$\frac{57}{486}$	$\frac{15}{62}$	$\frac{14}{20}$	$\frac{24}{22}$	$\frac{58}{58}$	259	$\frac{67}{230}$	$\frac{50}{87}$	$\frac{57}{65}$	$\frac{87}{87}$
10^{-5}	115	$\frac{64}{125}$	$\frac{18}{24}$	$\frac{10}{\mathbf{8}}$	$\frac{16}{10}$	$\frac{40}{40}$	472	$\frac{39}{474}$	$\frac{11}{42}$	$\frac{11}{14}$	$\frac{21}{18}$	$\frac{56}{56}$	172	$\frac{48}{154}$	$\frac{40}{62}$	$\frac{50}{54}$	$\frac{84}{85}$
10^{-6}	115	$\frac{63}{120}$	$\frac{18}{23}$	$\frac{10}{\mathbf{8}}$	$\frac{16}{10}$	$\frac{40}{40}$	473	$\frac{35}{472}$	$\frac{10}{38}$	$\frac{10}{13}$	$\frac{20}{17}$	$\frac{55}{55}$	155	$\frac{46}{139}$	$\frac{36}{58}$	$\frac{49}{52}$	$\frac{83}{83}$

Table 5.3: Iteration counts and execution times of the monotone BDD algorithm for the test problem (3.6), using the minimal and maximal overlap size above and below the line, respectively, for $N = 128$. N_1 is the number of mesh points in the x-direction, where the monotone BSUR method is in use.

In Section 4.5.2, it is observed that for the monotone BSUR method $\omega = 1$ results in the minimal iteration counts and execution times, leading to the conclusion that $\omega = 1$ is optimal. All the following numerical experiments have $\omega = 1$.

Using the execution times from Section 4.5.2, we compare the monotone BSUR method with the undecomposed method ($M = 1$) by calculating the serial acceleration. The serial acceleration is the execution time of the undecomposed method / by the execution time of the monotone BSUR method. The results are displayed in Figure 5.5. Figure 5.5 shows that for all values of N and ε the serial acceleration is significantly greater than one, indicating a very significant advantage in the monotone BSUR method.

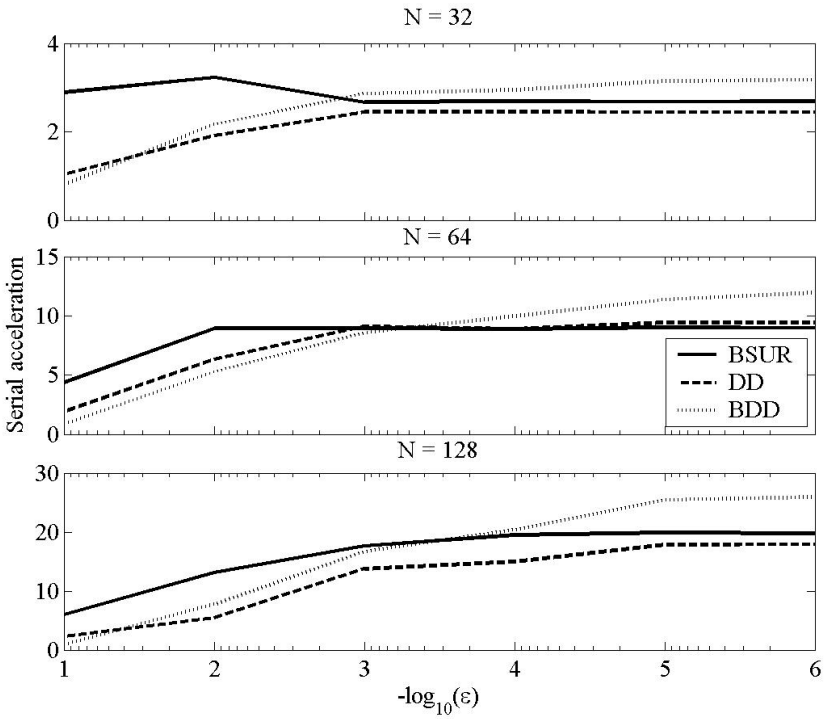

Figure 5.4: Serial acceleration of the monotone BDD algorithm, the monotone DD algorithm and the monotone BSUR method, for the test problem (3.6).

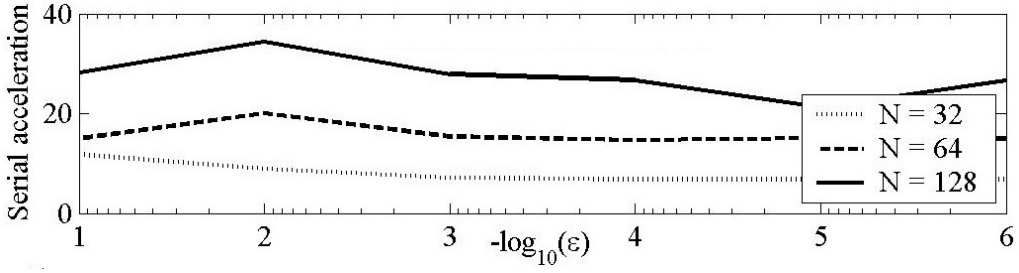

Figure 5.5: Serial acceleration of the monotone BSUR method for the test problem (3.8).

To investigate the serial acceleration of the monotone BDD algorithm, we begin by looking at the effect of varying the size of the first subdomain, which is the subdomain solved by the monotone BSUR method. Tables 5.4, 5.5 and 5.6 display the iteration counts and execution times of the monotone BDD algorithm for $N = 32$, $N = 64$ and $N = 128$, respectively. N_1 is the number of mesh points in the x-direction in the first subdomain. The minimum execution times for each value of the parameter ϵ are displayed in bold. From these tables, we can see that for $\epsilon < 10^{-1}$ the execution times are minimal when the first subdomain contains half the mesh points in the x-direction. This occurs when

the monotone BSUR method is used outside of the boundary layer and the monotone DD algorithm is used within the boundary layer. From these tables, it is also clear that for $N = 32, 64$ and 128 the execution times are minimal when the number of subdomains are $3, 5$ and 9, respectively. In all of the tables, we can also see that for each value of N_1 and M the iteration counts are the same for $\epsilon < 10^{-4}$. This indicates that the monotone BDD algorithm is parameter uniformly convergent with respect to its iteration counts.

Figure 5.6 displays the serial acceleration of the monotone BDD algorithm. From this figure, it is clear that for all values of N and ϵ, the monotone BDD algorithm gives a serial acceleration greater than one. It is also apparent that as ϵ decreases the serial acceleration of the monotone BDD algorithm increases.

Figure 5.7 displays the serial accelerations of the monotone BSUR method, the monotone DD algorithm and the monotone BDD algorithm. From this figure, it is clear that the monotone DD algorithm results in the lowest serial acceleration. For $\epsilon \geq 10^{-5}$, the monotone BSUR method results in the fastest acceleration, however, for $\epsilon < 10^{-5}$ the monotone BDD algorithm results in the highest serial acceleration.

5.5.3 Numerical observations

For both of the anisotropic convection-diffusion problem and convection-diffusion problem with parabolic layer, the monotone DD algorithm, the monotone BSUR method and the monotone BDD algorithm are parameter uniformly convergent with respect to their iteration counts. The serial acceleration for both of the test problems is greater than one for all the three of the methods. This indicates an advantage using these methods. For both of the test problems, for sufficiently large values of ε the monotone BSUR method results in the highest serial acceleration. However, for ε sufficiently small the monotone BDD algorithm results in the highest serial acceleration.

5.6 Conclusions

In this chapter, we investigate the composite monotone domain decomposition algorithms based on the Jacobi, Gauss-Seidel, block Jacobi, block Gauss-Seidel methods and the one- and two-level domain decomposition algorithms. In Theorems 16 and 17, monotone convergence of the Jacobi domain decomposition algorithm and the Gauss-Seidel domain

N_1	$N/4$				$N/2$				$3N/4$		
ε/M	2	3	5	9	2	3	5	9	2	3	5
	Iteration counts										
10^{-1}	55	$\frac{55}{55}$	$\frac{55}{55}$	$\frac{56}{56}$	75	$\frac{75}{75}$	$\frac{75}{75}$	$\frac{75}{75}$	84	$\frac{84}{84}$	$\frac{84}{84}$
10^{-2}	10	$\frac{16}{10}$	$\frac{19}{10}$	$\frac{27}{27}$	30	$\frac{31}{30}$	$\frac{37}{35}$	$\frac{45}{45}$	115	$\frac{116}{115}$	$\frac{116}{116}$
10^{-3}	7	$\frac{18}{7}$	$\frac{22}{9}$	$\frac{33}{33}$	35	$\frac{37}{35}$	$\frac{46}{42}$	$\frac{57}{57}$	144	$\frac{146}{145}$	$\frac{146}{146}$
10^{-4}	6	$\frac{18}{6}$	$\frac{23}{9}$	$\frac{33}{33}$	35	$\frac{38}{36}$	$\frac{47}{43}$	$\frac{59}{59}$	148	$\frac{150}{149}$	$\frac{150}{150}$
10^{-5}	6	$\frac{18}{6}$	$\frac{23}{9}$	$\frac{34}{34}$	35	$\frac{38}{36}$	$\frac{47}{43}$	$\frac{59}{59}$	149	$\frac{150}{150}$	$\frac{151}{151}$
10^{-6}	6	$\frac{18}{6}$	$\frac{23}{9}$	$\frac{34}{34}$	35	$\frac{38}{36}$	$\frac{47}{43}$	$\frac{59}{59}$	149	$\frac{150}{150}$	$\frac{151}{151}$
	Execution times (seconds)										
10^{-1}	2.75	$\frac{0.91}{2.81}$	$\frac{0.50}{0.91}$	$\frac{0.40}{0.39}$	1.07	$\frac{0.51}{1.09}$	$\frac{0.34}{0.43}$	$\frac{0.37}{0.36}$	0.24	$\frac{\mathbf{0.18}}{0.22}$	$\frac{0.22}{0.21}$
10^{-2}	0.59	$\frac{0.29}{0.73}$	$\frac{0.24}{0.28}$	$\frac{0.26}{0.26}$	0.20	$\frac{0.12}{0.21}$	$\frac{\mathbf{0.11}}{0.13}$	$\frac{0.14}{0.14}$	0.21	$\frac{0.18}{0.20}$	$\frac{0.20}{0.19}$
10^{-3}	0.42	$\frac{0.21}{0.53}$	$\frac{0.22}{0.29}$	$\frac{0.34}{0.33}$	0.15	$\frac{\mathbf{0.09}}{0.15}$	$\frac{0.09}{0.10}$	$\frac{0.15}{0.14}$	0.21	$\frac{0.19}{0.21}$	$\frac{0.22}{0.21}$
10^{-4}	0.35	$\frac{0.17}{0.47}$	$\frac{0.21}{0.28}$	$\frac{0.32}{0.31}$	0.10	$\frac{\mathbf{0.07}}{0.11}$	$\frac{0.08}{0.08}$	$\frac{0.14}{0.14}$	0.18	$\frac{0.17}{0.19}$	$\frac{0.22}{0.21}$
10^{-5}	0.35	$\frac{0.15}{0.44}$	$\frac{0.21}{0.27}$	$\frac{0.33}{0.32}$	0.09	$\frac{\mathbf{0.06}}{0.10}$	$\frac{0.08}{0.08}$	$\frac{0.13}{0.13}$	0.15	$\frac{0.16}{0.16}$	$\frac{0.21}{0.20}$
10^{-6}	0.35	$\frac{0.13}{0.44}$	$\frac{0.20}{0.26}$	$\frac{0.30}{0.30}$	0.08	$\frac{\mathbf{0.06}}{0.09}$	$\frac{0.07}{0.07}$	$\frac{0.13}{0.13}$	0.12	$\frac{0.14}{0.15}$	$\frac{0.19}{0.19}$

Table 5.4: Iteration counts and execution times of the monotone BDD algorithm for the test problem (3.8), using the minimal and maximal overlap size above and below the line, respectively, for $N = 32$. N_1 is the number of mesh points in the x-direction, where the monotone BSUR method is in use.

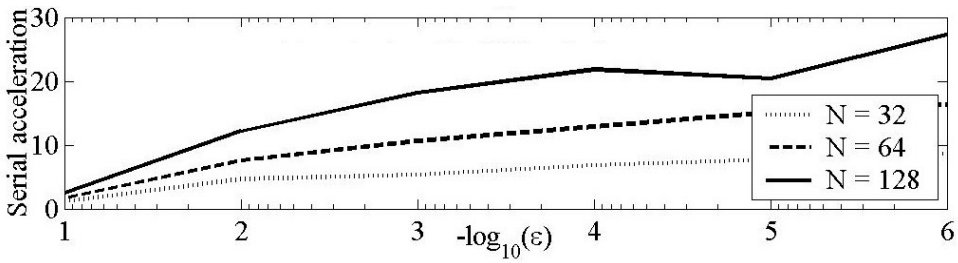

Figure 5.6: Serial acceleration of the monotone BDD algorithm for the test problem (3.8).

N_1	$N/4$					$N/2$					$3N/4$			
ε/M	2	3	5	9	17	2	3	5	9	17	2	3	5	9
	Iteration counts													
10^{-1}	165	$\frac{165}{165}$	$\frac{166}{165}$	$\frac{167}{166}$	$\frac{170}{170}$	234	$\frac{234}{234}$	$\frac{234}{234}$	$\frac{235}{235}$	$\frac{236}{236}$	266	$\frac{266}{266}$	$\frac{267}{267}$	$\frac{267}{267}$
10^{-2}	18	$\frac{22}{18}$	$\frac{26}{18}$	$\frac{43}{18}$	$\frac{71}{71}$	48	$\frac{49}{48}$	$\frac{55}{50}$	$\frac{74}{63}$	$\frac{107}{107}$	287	$\frac{288}{288}$	$\frac{290}{289}$	$\frac{292}{292}$
10^{-3}	8	$\frac{26}{8}$	$\frac{32}{8}$	$\frac{56}{22}$	$\frac{94}{94}$	59	$\frac{62}{60}$	$\frac{72}{63}$	$\frac{99}{85}$	$\frac{143}{143}$	380	$\frac{381}{380}$	$\frac{384}{383}$	$\frac{387}{387}$
10^{-4}	6	$\frac{27}{6}$	$\frac{33}{7}$	$\frac{59}{22}$	$\frac{97}{97}$	61	$\frac{64}{62}$	$\frac{75}{65}$	$\frac{103}{88}$	$\frac{148}{148}$	394	$\frac{395}{395}$	$\frac{399}{397}$	$\frac{402}{402}$
10^{-5}	6	$\frac{27}{6}$	$\frac{33}{7}$	$\frac{59}{22}$	$\frac{98}{98}$	62	$\frac{65}{62}$	$\frac{76}{65}$	$\frac{104}{89}$	$\frac{149}{149}$	396	$\frac{397}{396}$	$\frac{400}{399}$	$\frac{404}{404}$
10^{-6}	6	$\frac{27}{6}$	$\frac{33}{7}$	$\frac{59}{22}$	$\frac{98}{98}$	62	$\frac{65}{62}$	$\frac{76}{65}$	$\frac{104}{89}$	$\frac{149}{149}$	396	$\frac{397}{396}$	$\frac{400}{399}$	$\frac{404}{404}$
	Execution times (seconds)													
10^{-1}	82.8	$\frac{59.1}{124.4}$	$\frac{14.5}{76.0}$	$\frac{8.8}{16.9}$	$\frac{6.3}{6.3}$	61.7	$\frac{21.0}{66.4}$	$\frac{9.8}{21.5}$	$\frac{6.3}{8.9}$	$\frac{5.9}{5.9}$	8.6	$\frac{5.2}{9.7}$	$\frac{3.5}{4.8}$	$\frac{\mathbf{3.4}}{\mathbf{3.4}}$
10^{-2}	15.0	$\frac{5.7}{22.9}$	$\frac{5.6}{8.8}$	$\frac{4.7}{4.8}$	$\frac{5.0}{5.0}$	9.6	$\frac{2.4}{9.5}$	$\frac{1.3}{2.9}$	$\frac{\mathbf{1.2}}{1.4}$	$\frac{1.9}{1.9}$	6.6	$\frac{3.3}{6.8}$	$\frac{2.4}{3.1}$	$\frac{2.8}{2.8}$
10^{-3}	5.6	$\frac{3.3}{8.2}$	$\frac{4.2}{3.6}$	$\frac{4.7}{4.6}$	$\frac{8.8}{8.7}$	3.3	$\frac{1.4}{3.4}$	$\frac{\mathbf{0.9}}{1.6}$	$\frac{1.1}{1.1}$	$\frac{2.2}{2.2}$	5.9	$\frac{3.0}{6.0}$	$\frac{2.5}{3.0}$	$\frac{3.3}{3.3}$
10^{-4}	4.1	$\frac{2.7}{6.5}$	$\frac{3.2}{3.5}$	$\frac{4.4}{4.4}$	$\frac{7.7}{7.7}$	2.4	$\frac{1.0}{2.5}$	$\frac{\mathbf{0.7}}{1.2}$	$\frac{1.0}{1.0}$	$\frac{2.2}{2.2}$	4.5	$\frac{2.5}{4.7}$	$\frac{2.4}{2.8}$	$\frac{3.3}{3.3}$
10^{-5}	4.3	$\frac{2.2}{6.5}$	$\frac{3.5}{3.5}$	$\frac{4.4}{4.0}$	$\frac{8.2}{8.2}$	2.0	$\frac{0.8}{2.1}$	$\frac{\mathbf{0.6}}{1.0}$	$\frac{0.9}{0.9}$	$\frac{2.1}{2.1}$	3.0	$\frac{1.9}{3.2}$	$\frac{2.1}{2.3}$	$\frac{3.1}{3.1}$
10^{-6}	4.4	$\frac{2.0}{6.4}$	$\frac{3.4}{3.5}$	$\frac{4.7}{3.9}$	$\frac{8.1}{8.1}$	1.8	$\frac{0.7}{1.9}$	$\frac{\mathbf{0.6}}{0.9}$	$\frac{0.9}{0.9}$	$\frac{2.0}{2.0}$	2.3	$\frac{1.7}{2.6}$	$\frac{2.0}{2.2}$	$\frac{3.0}{3.0}$

Table 5.5: Iteration counts and execution times of the monotone BDD algorithm for the test problem (3.8), using the minimal and maximal overlap size above and below the line, respectively, for $N = 64$. N_1 is the number of mesh points in the x-direction, where the monotone BSUR method is in use.

decomposition algorithm, respectively, is proven. In Theorem 18, the convergence of these two algorithms is compared and proven that the Gauss-Seidel domain decomposition algorithm is the fastest.

In Theorems 19 and 21, monotone convergence of the block Jacobi domain decomposition algorithm and the block Gauss-Seidel domain decomposition algorithm, respectively, is proven. In Theorem 20, the point and block Jacobi domain decomposition algorithms are compared, and it is shown that the block Jacobi domain decomposition algorithm converges faster. In Theorem 22, the point and block Gauss-Seidel domain decomposition algorithms are compared, and it is shown that the block Gauss-Seidel domain decompo-

N_1	$N/4$						$N/2$						$3N/4$				
ε/M	2	3	5	9	17	33	2	3	5	9	17	33	2	3	5	9	17
	Iteration counts																
10^{-1}	520	$\frac{520}{520}$	$\frac{520}{520}$	$\frac{521}{520}$	$\frac{525}{522}$	$\frac{534}{534}$	752	$\frac{752}{752}$	$\frac{752}{752}$	$\frac{753}{752}$	$\frac{756}{755}$	$\frac{759}{759}$	866	$\frac{866}{866}$	$\frac{867}{866}$	$\frac{868}{867}$	$\frac{870}{870}$
10^{-2}	44	$\frac{44}{44}$	$\frac{44}{44}$	$\frac{63}{44}$	$\frac{104}{44}$	$\frac{185}{185}$	82	$\frac{82}{82}$	$\frac{88}{82}$	$\frac{112}{87}$	$\frac{170}{132}$	$\frac{279}{279}$	745	$\frac{745}{745}$	$\frac{747}{746}$	$\frac{752}{750}$	$\frac{759}{759}$
10^{-3}	11	$\frac{39}{11}$	$\frac{48}{11}$	$\frac{84}{16}$	$\frac{138}{55}$	$\frac{245}{245}$	100	$\frac{104}{100}$	$\frac{116}{102}$	$\frac{152}{112}$	$\frac{234}{184}$	$\frac{377}{377}$	1009	$\frac{1010}{1010}$	$\frac{1013}{1011}$	$\frac{1021}{1017}$	$\frac{1036}{1036}$
10^{-4}	7	$\frac{40}{7}$	$\frac{49}{7}$	$\frac{88}{17}$	$\frac{144}{57}$	$\frac{254}{254}$	106	$\frac{110}{106}$	$\frac{123}{108}$	$\frac{161}{120}$	$\frac{247}{195}$	$\frac{394}{394}$	1057	$\frac{1058}{1057}$	$\frac{1061}{1058}$	$\frac{1070}{1065}$	$\frac{1085}{1085}$
10^{-5}	6	$\frac{40}{6}$	$\frac{49}{7}$	$\frac{88}{17}$	$\frac{145}{57}$	$\frac{255}{255}$	106	$\frac{111}{107}$	$\frac{124}{108}$	$\frac{162}{120}$	$\frac{248}{196}$	$\frac{396}{396}$	1062	$\frac{1063}{1062}$	$\frac{1066}{1064}$	$\frac{1075}{1070}$	$\frac{1091}{1091}$
10^{-6}	6	$\frac{40}{6}$	$\frac{49}{7}$	$\frac{88}{17}$	$\frac{145}{57}$	$\frac{256}{256}$	106	$\frac{111}{107}$	$\frac{124}{109}$	$\frac{162}{121}$	$\frac{248}{196}$	$\frac{396}{396}$	1063	$\frac{1063}{1063}$	$\frac{1066}{1064}$	$\frac{1076}{1071}$	$\frac{1091}{1091}$
10^{-6}	6	$\frac{40}{6}$	$\frac{49}{7}$	$\frac{88}{17}$	$\frac{145}{57}$	$\frac{256}{256}$	106	$\frac{111}{107}$	$\frac{124}{109}$	$\frac{162}{121}$	$\frac{248}{196}$	$\frac{396}{729}$	1063	$\frac{1063}{1063}$	$\frac{1066}{1064}$	$\frac{1076}{1071}$	$\frac{1091}{1151}$
	Execution times (seconds)																
10^{-1}	8851	$\frac{5143}{10941}$	$\frac{1109}{5831}$	$\frac{231}{1829}$	$\frac{127}{280}$	$\frac{98}{97}$	6692	$\frac{1903}{7027}$	$\frac{402}{2104}$	$\frac{158}{350}$	$\frac{96}{130}$	$\frac{102}{102}$	961	$\frac{206}{893}$	$\frac{87}{187}$	$\frac{\mathbf{55}}{74}$	$\frac{60}{60}$
10^{-2}	985	$\frac{456}{1251}$	$\frac{105}{837}$	$\frac{72}{163}$	$\frac{89}{75}$	$\frac{79}{79}$	507	$\frac{132}{562}$	$\frac{23}{170}$	$\frac{\mathbf{14}}{28}$	$\frac{15}{15}$	$\frac{32}{32}$	586	$\frac{99}{550}$	$\frac{47}{119}$	$\frac{35}{45}$	$\frac{46}{46}$
10^{-3}	273	$\frac{157}{361}$	$\frac{72}{182}$	$\frac{58}{57}$	$\frac{103}{84}$	$\frac{163}{163}$	404	$\frac{40}{348}$	$\frac{13}{50}$	$\frac{\mathbf{10}}{16}$	$\frac{16}{14}$	$\frac{40}{40}$	242	$\frac{79}{258}$	$\frac{42}{94}$	$\frac{39}{46}$	$\frac{57}{58}$
10^{-4}	164	$\frac{94}{213}$	$\frac{41}{94}$	$\frac{45}{46}$	$\frac{94}{76}$	$\frac{166}{166}$	365	$\frac{30}{362}$	$\frac{10}{35}$	$\frac{\mathbf{8}}{12}$	$\frac{15}{13}$	$\frac{40}{40}$	200	$\frac{58}{204}$	$\frac{35}{70}$	$\frac{38}{42}$	$\frac{58}{59}$
10^{-5}	137	$\frac{76}{177}$	$\frac{42}{89}$	$\frac{42}{43}$	$\frac{89}{70}$	$\frac{158}{158}$	355	$\frac{22}{355}$	$\frac{7}{26}$	$\frac{\mathbf{7}}{9}$	$\frac{14}{12}$	$\frac{39}{39}$	124	$\frac{36}{125}$	$\frac{27}{45}$	$\frac{34}{37}$	$\frac{56}{57}$
10^{-6}	137	$\frac{72}{179}$	$\frac{45}{90}$	$\frac{42}{42}$	$\frac{84}{73}$	$\frac{160}{160}$	356	$\frac{20}{350}$	$\frac{\mathbf{7}}{24}$	$\frac{7}{9}$	$\frac{14}{12}$	$\frac{39}{39}$	90	$\frac{31}{93}$	$\frac{25}{40}$	$\frac{34}{35}$	$\frac{56}{56}$

Table 5.6: Iteration counts and execution times of the monotone BDD algorithm for the test problem (3.8), using the minimal and maximal overlap size above and below the line, respectively, for $N = 128$. N_1 is the number of mesh points in the x-direction, where the monotone BSUR method is in use.

sition algorithm converges faster. In Theorem 23, the convergence of the Gauss-Seidel and block Gauss-Seidel domain decomposition algorithms is compared and proven that the Gauss-Seidel domain decomposition algorithm converges faster.

The two-level composite monotone domain decomposition algorithms are constructed. The parrallisable nature of these algorithms are discussed.

As the theoretical results indicate that the block Gauss-Seidel domain decomposition algorithm has the fastest convergence, we apply this algorithm to the two test problems: the convection-diffusion problem with parabolic boundary layers and the anisotropic

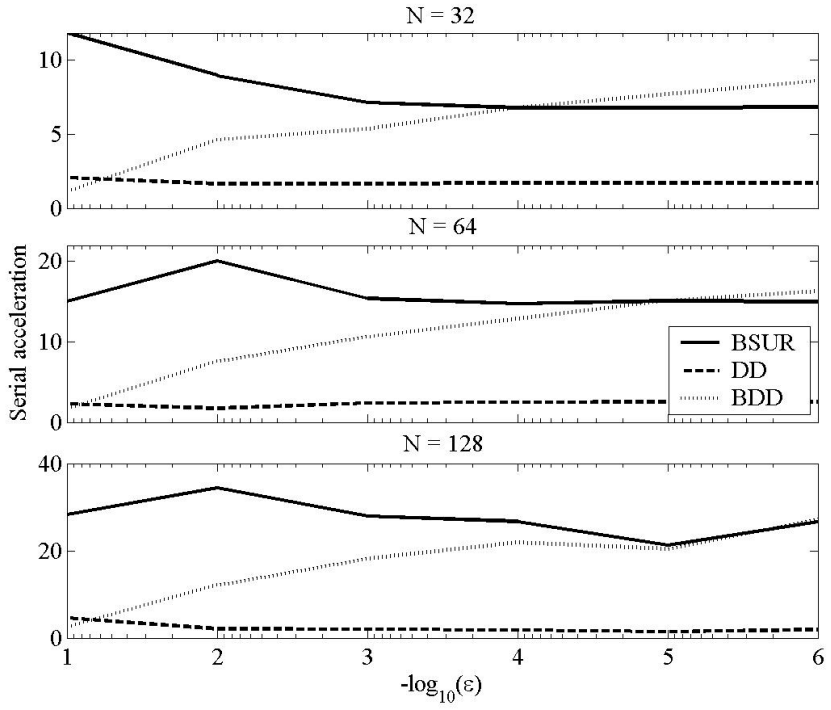

Figure 5.7: Serial acceleration of the monotone BDD algorithm, the monotone DD algorithm and the monotone BSUR method for the test problem (3.8).

convection-diffusion problem. For both of the test problems, the block Gauss-Seidel domain decomposition algorithm converges ε-uniformly with respect to its iteration counts. The numerical experiments are compared to the numerical experiments in Chapters 3 and 4. The numerical experiments on the test problems show that for sufficiently large values of ε, the monotone BSUR method from Chapter 4 results in the highest serial acceleration. However, for ε sufficiently small the composite monotone block Gauss-Seidel domain decomposition algorithm results in the highest serial acceleration.

Chapter 6

Multigrid methods

In this chapter, we construct three monotone multigrid methods: the monotone multigrid method, the block monotone multigrid method and the two-level monotone multigrid algorithm. The monotone multigrid method, the block monotone multigrid method and the two-level monotone multigrid algorithm use the monotone successive underrelaxation method, the monotone block successive relaxation method and the monotone domain decomposition algorithm, respectively, for the smoothing part. All three methods use the full approximation scheme (FAS) from [27] for the course correction part. Monotone convergence of these algorithms is proven and numerical experiments are presented.

6.1 Introduction

Multigrid methods are accepted as fast efficient solvers, especially for elliptic problems. They consist of the two parts: the smoother which reduces the high frequency components in the error between numerical and exact solutions and the coarse correction based on the fact that the smooth error can be well represented on coarser meshes. The standard multigrid methods have shown to be unsatisfactory when applied to singularly perturbed problems [38].

A modified multigrid method is applied to linear singularly perturbed convection-diffusion equations in [38]. In [41], the monotone multigrid method is applied to nonlinear elliptic boundary value problems, and its monotone convergence is proven. In [3] and [71], a monotone multigrid method for nonlinear elliptic problems where the prolongation parameter is determined adaptively, is presented. The prolongation parameter $\rho_k^{(n)}$ is

chosen as large as possible from the solution of

$$[A_k + F_k' v_k]z(p) = f_k - \mathcal{L}_k v_k, \quad \rho_k^{(n)}(p) = \frac{z(p)}{e_k(p)}, \quad p \in \Omega_k \tag{6.1}$$

where A_k and $F_k v_k$ are the linear and nonlinear terms, respectively, of $\mathcal{L}v_k$ and $e_k(p)$ is the prolongated error of the approximation on the mesh Ω_{k-1}. The course grid correction to the approximation is given by

$$w_k^{(n)} = v_k^{(n,t_1)} + \rho_k^{(n)} e_k^{(n)}.$$

From (6.1), we have

$$w_k^{(n)}(p) = v_k^{(n,t_1)}(p) + z(p).$$

From here and (6.1), we can see that the course grid correction $w_k^{(n)}(p)$ does not receive any information from the approximation on the meshes Ω_{k-1} and hence, there is no reason to find approximations on different meshes and, hence, this method is not a multigrid method.

In this chapter, we construct three monotone multigrid methods: the monotone multigrid method uses the monotone successive under-relaxation (SUR) method for the smoothing part; the block monotone multigrid method uses the block monotone successive under-relaxation (BSUR) method for the smoothing part; the monotone multigrid algorithm uses the monotone domain decomposition algorithm from Chapter 3 for the smoothing part. All the three monotone multigrid methods use the full approximation scheme (FAS) from [27] for the course correction part. The advantages of the monotone multigrid methods are that they solve only linear discrete systems at each iterative step and converge globally. Numerical results show that the monotone multigrid methods converge uniformly in the perturbation parameter.

In Section 6.2, we construct the monotone multigrid method and prove monotone convergence of the method. Numerical experiments are presented in Section 6.2.1 for the convection-diffusion problem with parabolic layers and the anisotropic convection-diffusion problem. In Section 6.3, we construct the monotone multigrid method and prove monotone convergence of the method. Numerical experiments are presented in Section 6.3.1. In Section 6.4, the two-level monotone multigrid algorithm is constructed, and monotone convergence of the algorithm is proven. Numerical experiments for two test problems are presented in Section 6.4.1.

6.2 Monotone multigrid method

In this section, we investigate the monotone multigrid method for solving nonlinear singularly perturbed convection-diffusion problems. Let $\{\Omega_k^h\}$ be the sequence of meshes

$$\Omega_0^h \subset \Omega_1^h \subset \ldots \subset \Omega_K^h, \tag{6.2}$$

where Ω_K^h is the finest mesh. On each mesh Ω_k^h, $k = 0, \ldots, K$, the nonlinear difference scheme (1.15) can be represented in the form

$$\mathcal{L}_k v_k(p) + f(p, v_k) = 0, \quad p \in \Omega_k^h, \quad v_k(p) = g(p), \quad p \in \partial\Omega_k^h, \tag{6.3}$$

$$\mathcal{L}_k v_k(p) = d_k(p) v_k(p) - \sum_{p' \in \sigma_k'(p)} e_k(p, p') v_k(p').$$

On the finest mesh Ω_K^h, choose an initial function $v_K^{(0)}(p)$, $p \in \Omega_K^h$, satisfying the boundary condition $v_K^{(0)}(p) = g(p)$, $p \in \partial\Omega_K^h$, and which is an upper or lower solution to (6.3).

As (1.15) is the nonlinear problem, the full approximation scheme (FAS) multigird cycle [27] is used. Both the linear multigrid cycle and the FAS multigrid cycle transfer the defect $r_k^{(n)}$ to the course grid. The FAS multigrid cycle also transfers the restricted approximation $v_k^{(n)}$ to the course grid. When going from a course grid level to a fine grid level, the linear mulitgrid cycle interpolates the solution. However, for nonlinear problems, in general, this does not converge to a solution [68]. Instead, the FAS multigrid cycle prolongs the error $e_k^{(n)}$ on the course mesh. This error is then used to correct the approximation on the finer mesh. The algorithm for the FAS monotone multigrid method is as follows:

FAS monotone multigrid cycle $v_k^{(n+1)} = MMG\{k, v_k^{(n)}, \mathcal{L}_k, \phi_k^{(n)}, t_1, t_2\}$
$\phi_K^{(n)} = 0$ on the finest mesh Ω_K^h.

1. Presmoothing:

$$v_k^{(n,t)} = S_k(v_k^{(n,t-1)}, \mathcal{L}_k^{\text{SUR}}, \phi_k^{(n)}, t_1), \quad t = 1, \ldots, t_1, \tag{6.4}$$

where $v_k^{(n,0)} = v_k^{(n)}$, the mapping S_k defines the monotone SUR method (4.3) on Ω_k^h, and t_1 is the number of presmoothing steps.

2. Coarse-grid correction:

(a) On Ω_k^h, compute the residual

$$\mathcal{R}_k(v_k^{(n,t_1)}) = \mathcal{L}_k v_k^{(n,t_1)} + f(v_k^{(n,t_1)}).$$

(b) Restrict $v_k^{(n,t_1)}$ and $\mathcal{R}_k(v_k^{(n,t_1)})$ on the coarse mesh Ω_{k-1}^h

$$v_{k-1}^{(n)} = I_k^{k-1} v_k^{(n,t_1)}, \quad r_{k-1}^{(n)} = I_k^{k-1}\mathcal{R}_k(v_k^{(n,t_1)}), \tag{6.5}$$

where I_k^{k-1} is a restriction operator.

(c) On Ω_{k-1}^h, solve the nonlinear difference scheme

$$\mathcal{L}_{k-1} w_{k-1}^{(n)} + f(w_{k-1}^{(n)}) = \phi_{k-1}^{(n)}, \quad \phi_{k-1}^{(n)} = \mathcal{R}_{k-1}(v_{k-1}^{(n)}) - r_{k-1}^{(n)}, \tag{6.6}$$

$$w_{k-1}^{(n)}(p) = g(p), \quad p \in \partial\Omega_{k-1}^h.$$

If $k = 1$ then solve (6.6) as exactly as necessary, by using the monotone SUR method (4.3) on Ω_0^h. If $k > 1$ then solve (6.6) by the algorithm

$$w_{k-1}^{(n)} = MMG\{k-1, w_{k-1}^{(n,0)}, \mathcal{L}_{k-1}, \phi_{k-1}^{(n)}, t_1, t_2\},$$

with the initial iterate $w_{k-1}^{(n,0)} = v_{k-1}^{(n)}$, where $v_{k-1}^{(n)}$ is defined in (6.5).

(d) Prolong the error $e_{k-1}^{(n)} = w_{k-1}^{(n)} - v_{k-1}^{(n)}$ on the fine mesh Ω_k^h

$$e_k^{(n)} = I_{k-1}^k (w_{k-1}^{(n)} - v_{k-1}^{(n)}),$$

where I_{k-1}^k is a prolongation operator.

(e) Correct the approximation $v_k^{(n,t_1)}$ on the fine mesh Ω_k^h

$$w_k^{(n)} = v_k^{(n,t_1)} + \rho_k^{(n)} \widetilde{e}_k^{(n)}, \tag{6.7}$$

$$\widetilde{e}_k^{(n)}(p) = \begin{cases} 0, & p \in \widehat{\Omega}_k^h = \{p : \mathcal{R}_k(p, v_k^{(n,t_1)}) = 0\}, \\ e_k^{(n)}(p), & p \in \widetilde{\Omega}_k^h = \{p : \mathcal{R}_k(p, v_k^{(n,t_1)}) \neq 0\}, \end{cases}$$

where a prolongation parameter $\rho_k^{(n)} > 0$ is chosen in (6.9).

3. Postsmoothing:

$$v_k^{(n,t)} = S_k\{v_k^{(n,t-1)}, \mathcal{L}_k^{\mathrm{SUR}}, \phi_k^{(n)}, t_2\}, \quad t = t_1 + 2, \ldots, t_1 + t_2 + 1, \tag{6.8}$$

with the initial iterate $v_k^{(n,t_1+1)} = w_k^{(n)}$, where $w_k^{(n)}$ from (6.7), S_k replaces the monotone SUR method (4.3) on Ω_k^h, and t_2 is the number of postsmoothing steps. On Ω_k^h, set up $v_k^{(n+1)} = v_k^{(n,t_1+t_2+1)}$.

The prolongation parameter $\rho_k^{(n)}$ in (6.7) is given by

$$\rho_k^{(n)} = \min_{p\in\Omega_k^{h*}} -[\mathcal{R}_k(p, v_k^{(n,t_1)})/(\mathcal{L}_k + c^*)\widetilde{e}_k^{(n)}(p)], \tag{6.9}$$

where in the case of upper (lower) solutions, Ω_k^{h*} is given by

$$\Omega_k^{h*} = \{p : (\mathcal{L}_k + c^*)\widetilde{e}_k^{(n)}(p) < 0\ (>0)\}, \quad \Omega_k^{h*} \subseteq \widetilde{\Omega}_k^h.$$

We say that Q is a monotone operator if $v \geq w$ implies $Q\{v\} \geq Q\{w\}$, and assume that I_k^{k-1} and I_{k-1}^k are linear and monotone.

Remark 13. In the two dimensional case, the full weighted restriction operator ($I_k^{k-1}v_k = v_{k-1}$) and the standard prolongation operator ($I_{k-1}^k v_{k-1} = v_k$) are given by

$$v_{k-1_{i,j}} = \tfrac{1}{16}[v_{k_{2i-1,2j-1}} + v_{k_{2i-1,2j+1}} + v_{k_{2i+1,2j-1}} + v_{k_{2i+1,2j+1}}$$

$$+2(v_{k_{2i,2j-1}} + v_{k_{2i,2j+1}} + v_{k_{2i-1,2j}} + v_{k_{2i+1,2j}}) + 4v_{k_{2i,2j}}],$$

$$v_{k_{2i,2j}} = v_{k-1_{i,j}}, \quad v_{k_{2i,2j+1}} = \tfrac{1}{4}[v_{k-1_{i,j}} + v_{k-1_{2i,j+1}}v_{k-1_{i+1,j}} + v_{k-1_{2i+1,j+1}}],$$

$$v_{k_{2i,2j+1}} = \tfrac{1}{2}[v_{k-1_{i,j}} + v_{k-1_{2i,j+1}}], \quad v_{k_{2i+1,2j}} = \tfrac{1}{2}[v_{k-1_{i,j}} + v_{k-1_{2i+1,j}}],$$

$$0 \leq i, j \leq N/2 - 1.$$

It is easy to check that both the full weighted restriction operator and the standard prolongation operator are linear and monotone.

Theorem 25. *Let the coefficients of the difference operators from (6.3) satisfy (1.16). If $\overline{v}_K^{(0)}$ and $\underline{v}_K^{(0)}$ are upper and lower solutions of the difference scheme (6.3), then the upper sequence $\{\overline{v}_K^{(n)}\}$ obtained by the monotone multigrid method converges monotonically from above to the unique solution v_K of (6.3), the lower sequence $\{\underline{v}_K^{(n)}\}$ converges monotonically from below to v_K:*

$$\underline{v}_K^{(n-1)}(p) \leq \underline{v}_K^{(n)}(p) \leq v_K(p) \leq \overline{v}_K^{(n)}(p) \leq \overline{v}_K^{(n-1)}(p), \quad p \in \overline{\Omega}_K^h, \quad n \geq 1.$$

Proof. Let $\overline{v}_k^{(0)}$ be an upper solution. Using $\overline{v}_k^{(0,0)}(p) = \overline{v}_k^{(0)}$ and Theorem 9, from (6.4) we have

$$\overline{v}_k^{(0,t)}(p) \leq \overline{v}_k^{(0,t-1)}(p), \quad p \in \overline{\Omega}_k^h,$$

$$\mathcal{R}_k(p, \overline{v}^{(0,t)}) \geq 0, \quad p \in \Omega_k^h, \quad t = 1, \ldots, t_1. \tag{6.10}$$

From here and using the monotone property of I_k^{k-1}, we conclude

$$r_{k-1}^{(0)} = I_k^{k-1}\mathcal{R}_k(\overline{v}_k^{(0,t_1)}) \geq 0, \quad p \in \Omega_{k-1}^h. \tag{6.11}$$

The residual of the difference scheme (6.6) may be written as

$$\begin{aligned}\mathcal{R}_{k-1}^*(p, w_{k-1}^{(0)}) &= \mathcal{L}_{k-1}w_{k-1}^{(0)}(p) + f(p, w_{k-1}^{(0)}) - \mathcal{R}_{k-1}(p, v_{k-1}^{(0)}) + r_{k-1}^{(0)} \\ &= \mathcal{R}_{k-1}(p, w_{k-1}^{(0)}) - \mathcal{R}_{k-1}(p, v_{k-1}^{(0)}) + r_{k-1}^{(0)}, \quad p \in \Omega_{k-1}^h.\end{aligned}$$

From here, (6.11) and $w_{k-1}^{(0,0)}(p) = v_{k-1}^{(0)}(p)$, we have

$$\mathcal{R}_{k-1}^*(p, w_{k-1}^{(0,0)}) = r_{k-1}^{(0)} \geq 0, \quad p \in \Omega_{k-1}^h.$$

Therefore, $w_{k-1}^{(0,0)}(p)$ is an upper solution to (6.6). Using (4.9), from Theorem 9 we have

$$\overline{w}_{k-1}^{(0,t)}(p) \leq \overline{w}_{k-1}^{(0,t-1)}(p) \leq \overline{w}_{k-1}^{(0,0)}(p) = v_{k-1}^{(0)}, \quad p \in \overline{\Omega}_{k-1}^h, \quad t = 1, \ldots, t_1.$$

From here, $w_{k-1}^{(0)}(p) = \overline{w}_{k-1}^{(0,t_1)}(p)$ and using the monotone property of I_{k-1}^k, we have

$$e_k^{(0)}(p) = I_{k-1}^k(w_{k-1}^{(0)}(p) - v_{k-1}^{(0)}(p)) \leq 0, \quad p \in \overline{\Omega}_k^h. \tag{6.12}$$

From here and (6.7), we have $\widetilde{e}_k^{(0)}(p) \leq 0$, $p \in \Omega_k^h$, and, hence,

$$w_k^{(0)}(p) \leq v_k^{(0,t_1)}(p), \quad p \in \overline{\Omega}_k^h, \tag{6.13}$$

for all positive values of $\rho_k^{(0)}$. We now prove that $w_k^{(0)}(p)$ defined in (6.7) is an upper solution, that is,

$$\mathcal{R}_k(p, w_k^{(0)}) \geq 0, \quad p \in \Omega_k^h. \tag{6.14}$$

We have $w_k^{(0)}(p) = v_k^{(0,t_1)}(p)$, $p \in \widehat{\Omega}_k^h$. From here, (1.16) and (6.13), we have

$$\begin{aligned}\mathcal{R}_k(p, w_k^{(0)}) &= \mathcal{L}_k w_k^{(0)} + f(p, w_k^{(0)}) \\ &= d(p)w_k^{(0)} - \sum_{p' \in \sigma'(p)} e(p,p')w_k^{(0)} + f(p, w_k^{(0)}) \\ &= d(p)v_k^{(0,t_1)} - \sum_{p' \in \sigma'(p)} e(p,p')w_k^{(0)} + f(p, v_k^{(0,t_1)}) \\ &\geq d(p)v_k^{(0,t_1)} - \sum_{p' \in \sigma'(p)} e(p,p')v_k^{(0,t_1)} + f(p, v_k^{(0,t_1)}) \\ &\geq \mathcal{R}_k(p, v_k^{(0,t_1)}) \geq 0, \quad p \in \widehat{\Omega}_k^h.\end{aligned} \tag{6.15}$$

For $p \in \widetilde{\Omega}_k^h$, using the mean value theorem, we have

$$\begin{aligned}
\mathcal{R}_k(p, w_k^{(0)}) &= \mathcal{L}_k w_k^{(0)} + f(p, w_k^{(0)}) \\
&= \mathcal{L}_k v_k^{(0,t_1)} + \mathcal{L}_k(\rho_k^{(0)} \widetilde{e}_k^{(0)}) + f(p, v_k^{(0,t_1)} + \rho_k^{(0)} \widetilde{e}_k^{(0)}) \\
&= \mathcal{L}_k v_k^{(0,t_1)} + \mathcal{L}_k(\rho_k^{(0)} \widetilde{e}_k^{(0)}) \pm f(p, v_k^{(0,t_1)}) + f(p, v_k^{(0,t_1)} + \rho_k^{(0)} \widetilde{e}_k^{(0)}) \\
&= \mathcal{L}_k v_k^{(0,t_1)} + f(p, v_k^{(0,t_1)}) + \mathcal{L}_k(\rho_k^{(0)} \widetilde{e}_k^{(0)}) + f_v^{(0)}(p) \rho_k^{(0)} \widetilde{e}_k^{(0)} \\
&= \mathcal{R}_k(p, v_k^{(0,t_1)}) + (\mathcal{L}_k + f_v^{(0)}) \rho_k^{(0)} \widetilde{e}_k^{(0)},
\end{aligned}$$

where $f_v^{(0)}(p) = (p, v_k^{(0,t_1)}(p) + \Theta_k^{(0)}(p) \rho_k^{(0)} \widetilde{e}_k^{(0)}(p))$, $0 < \Theta_k^{(0)}(p) < 1$. From here, (1.20), (6.7), (6.12) and $\rho_k^{(0)} > 0$, we have

$$\mathcal{R}_k(p, w_k^{(0)}) \geq \mathcal{R}_k(p, v_k^{(0,t_1)}) + (\mathcal{L}_k + c^*) \rho_k^{(0)} \widetilde{e}_k^{(0)}(p), \quad p \in \Omega_k^h.$$

For $p \in \widetilde{\Omega}_k^h \setminus \widetilde{\Omega}_k^{h*}$, $(\mathcal{L}_k + c^*) \widetilde{e}_k^{(0)}(p) \geq 0$. From here, (1.16) and (6.10), we conclude

$$\mathcal{R}_k(p, w_k^{(0)}) \geq \mathcal{R}_k(p, v_k^{(0,t_1)}) > 0, \quad p \in \widetilde{\Omega}_k^h \setminus \widetilde{\Omega}_k^{h*}. \tag{6.16}$$

For $p \in \widetilde{\Omega}_k^{h*}$, $(\mathcal{L}_k + c^*) \widetilde{e}_k^{(0)} < 0$. From here, we have

$$\begin{aligned}
\mathcal{R}_k(p, w_k^{(0)}) &= \mathcal{R}_k(p, v_k^{(0,t_1)}) + (\mathcal{L}_k + c^*) \rho_k^{(0)} \widetilde{e}_k^{(0)}(p), \quad p \in \widetilde{\Omega}_k^{h*} \\
&= \mathcal{R}_k(p, v_k^{(0,t_1)}) + \left[\min_{p \in \widetilde{\Omega}_k^{h*}} \frac{-\mathcal{R}_k(p, v_k^{(0,t_1)})}{(\mathcal{L}_k + c^*) \widetilde{e}_k^{(0)}(p)} \right] (\mathcal{L}_k + c^*) \widetilde{e}_k^{(0)}(p),
\end{aligned}$$

where

$$\begin{aligned}
\left[\min_{p \in \widetilde{\Omega}_k^{h*}} \frac{-\mathcal{R}_k(p, v_k^{(0,t_1)})}{(\mathcal{L}_k + c^*) \widetilde{e}_k^{(0)}(p)} \right] (\mathcal{L}_k + c^*) \widetilde{e}_k^{(0)}(p) &\geq \left[\frac{-\mathcal{R}_k(p, v_k^{(0,t_1)})}{(\mathcal{L}_k + c^*) \widetilde{e}_k^{(0)}(p)} \right] (\mathcal{L}_k + c^*) \widetilde{e}_k^{(0)}(p) \\
&= -\mathcal{R}_k(p, v_k^{(0,t_1)}), \quad p \in \widetilde{\Omega}_k^{h*}.
\end{aligned}$$

From here, we obtain

$$\mathcal{R}_k(p, w_k^{(0)}) \geq \mathcal{R}_k(p, v_k^{(0,t_1)}) - \mathcal{R}_k(p, v_k^{(0,t_1)}) = 0, \quad p \in \widetilde{\Omega}_k^{h*}.$$

From here, (6.15) and (6.16), we prove (6.14). Since $v_k^{(0,t_1+1)} = w_k^{(0)}$, $p \in \overline{\Omega}_k^h$, using (4.9), from Theorem 9 we have

$$\overline{v}_k^{(0,t)}(p) \leq \overline{v}_k^{(0,t-1)}(p), \quad p \in \overline{\Omega}_k^h,$$

$$\mathcal{R}_k(p, v^{(0,t)}) \geq 0, \quad p \in \Omega_k^h, \quad t = t_1 + 2, \ldots, t_1 + t_2 + 1.$$

As $v_k^{(1)}(p) = v_k^{(0,t_1+t_2+1)}(p)$, $p \in \overline{\Omega}_k^h$, we have

$$\mathcal{R}_k(p, v_k^{(1)}(p)) \geq 0,$$

and conclude that $v_k^{(1)}(p)$ is an upper solution, where

$$v_k^{(1)}(p) \leq v_k^{(0,t_1+2)}(p) \leq w_k^{(0)}(p) \leq v_k^{(0,t_1)}(p) \leq v_k^{(0)}(p), \quad p \in \overline{\Omega}_k^h.$$

By induction on n, we can prove that $\{v_k^{(n)}(p)\}$, $p \in \overline{\Omega}_k^h$ is a monotonically decreasing sequence of upper solutions. This sequence is bounded below by $\underline{v}_K$, where $\underline{v}_K$ is any lower solution (1.19), and, hence, the mesh function defined by

$$\overline{v}_K(p) = \lim_{n\to\infty} \overline{v}_K^{(n)}(p), \quad p \in \overline{\Omega}_K^h,$$

exists. Since

$$\lim_{n\to\infty} \overline{v}_K^{(n)}(p) = \lim_{n\to\infty} \overline{v}_K^{(n,t)}(p), \quad p \in \overline{\Omega}_K^h,$$

and from Theorem 9, it follows that

$$\mathcal{R}_K(p, \overline{v}_K) = \lim_{n\to\infty} \mathcal{R}_K(p, v_K^{(n)}) = \lim_{n\to\infty} \mathcal{R}_K(p, v_K^{(n,t)}) = 0, \quad p \in \overline{\Omega}_K^h,$$

$$\overline{v}_K(p) = g(p), \quad p \in \partial\Omega_K^h.$$

Thus, $\overline{v}_K(p)$ is the solution to the difference scheme (6.3). □

6.2.1 Numerical experiments

We apply the monotone multigrid method (MMG) to the two test problems: the convection-diffusion problem with parabolic layers and the anisotropic convection-diffusion problem. For the following numerical experiments, the stopping criteria for the iterates is

$$\max_{p\in\overline{\Omega}^h} \|v^{(n)}(p) - v^{(n-1)}(p)\| \leq 10^{-6}.$$

The number of mesh points in the x- and y-directions are set equal to N. For the restriction and prolongation operators, we use the full weighting and bilinear interpolation operators, respectively.

Convection-diffusion problem with parabolic boundary layers

Consider the test problem

$$-\varepsilon(u_{xx}+u_{yy})+u_x+f(x,y,u)=0, \quad (x,y)\in\Omega=(0,1)\times(0,1), \tag{6.17}$$

$$f(x,y,u)=\exp(-1)-\exp(-u), \quad u=0 \text{ on } \partial\Omega.$$

This problem is the convection-diffusion problem with parabolic boundary layers (1.4.2), which is characterised by an elliptic boundary layer close to $x=1$ and parabolic boundary layers close to $y=0$ and $y=1$ (see Proposition 2 from Section 1.4.2 for details). To solve this problem numerically, we use the two dimensional piecewise uniform Shishkin mesh (1.11). We also require the constants c_*, c^* from (1.20). For this test problem, we have $f_u=\exp(-u)$ and the reduced equation

$$u_x+\exp(-1)-\exp(-u)=0, \quad (x,y)\in\Omega.$$

Thus, $u_r=1$ is the solution to the reduced equation. By writing the difference scheme at an interior mesh point as in (1.6), where the coefficients satisfy the inequalities (1.7), we can see that $\overline{u}(p)=1$, $p\in\Omega^h$, $\overline{u}(p)=0$, $p\in\partial\Omega^h$ and $\underline{u}(p)=0$, $p\in\overline{\Omega}^h$, are upper and lower solutions, respectively. Therefore, $0\le u(p)\le 1$, $\exp(-1)\le f_u(u)\le 1$, $p\in\Omega^h$, and

$$c_*=\exp(-1), \quad c^*=1. \tag{6.18}$$

We also require an initial solution. For the test problem (6.17), $\overline{u}^{(0)}(p)=1$, $p\in\Omega^h$, $\overline{u}^{(0)}(p)=0$, $p\in\partial\Omega^h$ is an initial upper solution.

Figure 6.1 displays the four graphs for the cycle counts of the MMG method over varying values of ω and ε. Each graph corresponds to different numbers t_1, t_2 of presmoothing and postsmoothing steps, respectively. All the four graphs indicate that for $\epsilon<10^{-4}$ the cycle counts are independent of ε. This leads to the conclusion that the MMG method converges parameter uniformly in its cycle counts. The graphs also show that the relaxation parameter $\omega=1$ is optimal for the MMG method when applied to the convection-diffusion problem with parabolic boundary layers.

Figure 6.2 displays the execution times of the MMG and SUR methods for the test problem (6.17). From Figure 6.2, it is apparent that for $\varepsilon<10^{-1}$ the SUR method results in the fastest execution times.

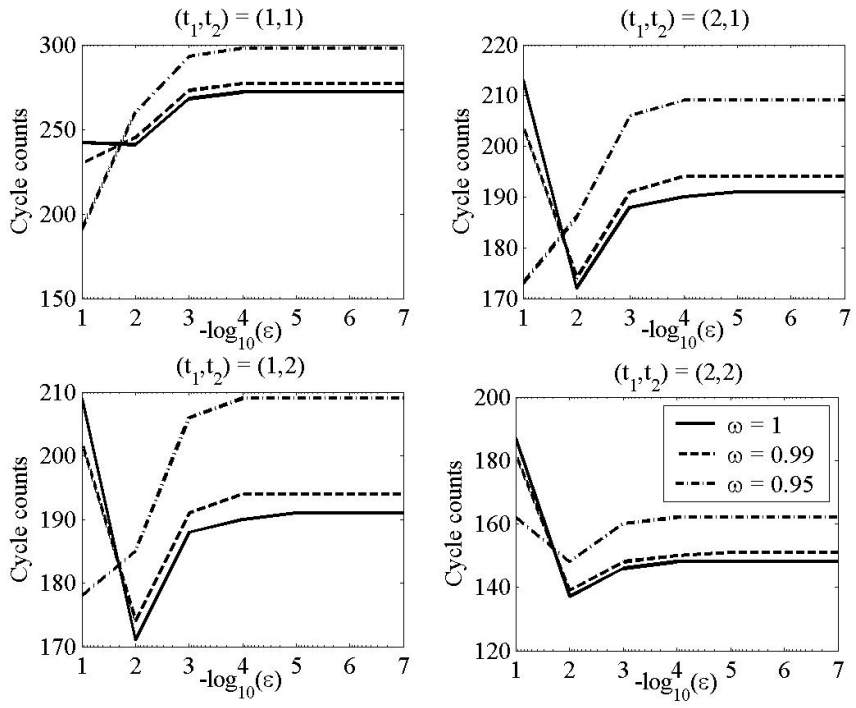

Figure 6.1: Cycle counts of the MMG method for $N = 64$ and varying values of ε for the test problem (6.17).

Anisotropic convection-diffusion problem

Consider the test problem

$$-\varepsilon u_{xx} - u_{yy} + u_x + f(x, y, u) = 0, \quad (x, y) \in \Omega = (0, 1) \times (0, 1), \tag{6.19}$$

$$f(x, y, u) = \exp(-1) - \exp(-u), \quad u = 0 \text{ on } \partial\Omega.$$

This problem is the anisotropic convection-diffusion problem (1.12) which is characterised by an elliptic boundary layer close to $x = 1$ (see Proposition 3 from Section 1.4.3 for details). To solve this problem numerically, we use the two dimensional piecewise uniform Shishkin mesh (1.8). We also require the constants c_*, c^* from (1.20). For this test problem, we have $f_u = \exp(-u)$ and the reduced equation

$$u_{yy} + u_x + \exp(-1) - \exp(-u) = 0, \quad (x, y) \in \Omega.$$

Thus, $u_r = 1$ is a solution to the reduced equation. By writing the difference scheme at an interior mesh point as in (1.6), where the coefficients satisfy the inequalities (1.7), we

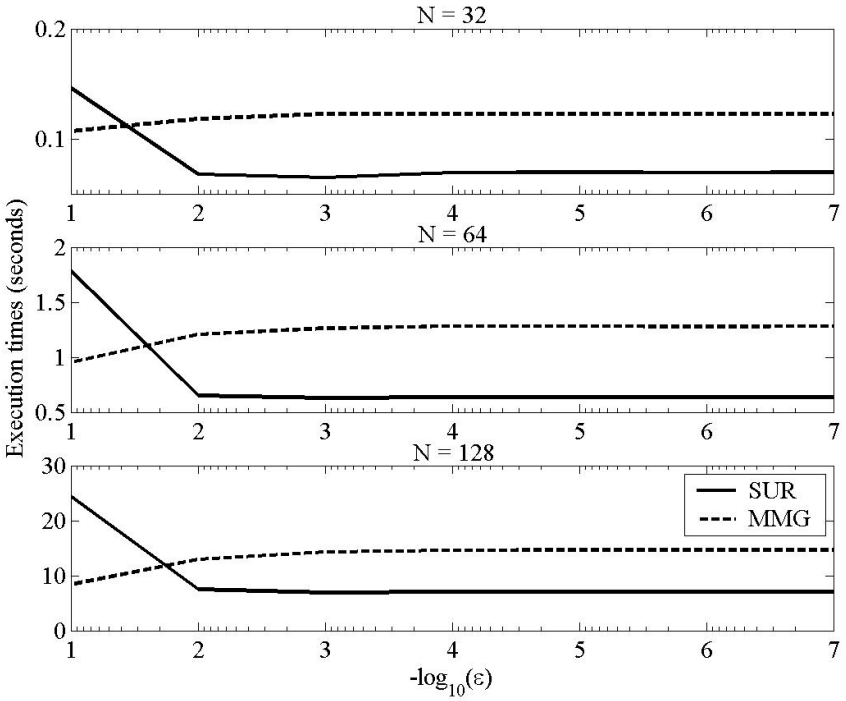

Figure 6.2: Execution times of the MMG and SUR methods for $N = 64$ and varying values of ε for the test problem (6.17).

can see that $\overline{u}(p) = 1$, $p \in \Omega^h$ $\overline{u}(p) = 0$, $p \in \partial\Omega^h$ and $\underline{u}(p) = 0$, $p \in \overline{\Omega}^h$, are upper and lower solutions, respectively. Therefore, $0 \leq u(p) \leq 1$, $\exp(-1) \leq f_u(u) \leq 1$, $p \in \Omega^h$ and

$$c_* = \exp(-1), \quad c^* = 1. \tag{6.20}$$

We also require an initial solution. For the test problem (6.19), $\overline{u}^{(0)}(p) = 1$, $p \in \Omega^h$, $\overline{u}^{(0)}(p) = 0$, $p \in \partial\Omega^h$ is an initial upper solution.

Figure 6.3 displays the four graphs with the cycle counts of the MMG method over varying values of ω and ε. Each graph corresponds to different numbers t_1, t_2 of presmoothing and postsmoothing steps, respectively. All the four graphs indicate that for $\epsilon < 10^{-4}$ the cycle counts are independent of ε. We therefore conclude that the MMG method converges parameter uniformly in its cycle counts. These graphs also show that the relaxation parameter $\omega = 1$ is not optimal for the MMG method when applied to the anisotropic convection-diffusion problem.

Figure 6.4 displays the execution times of the MMG and SUR methods for the test problem (6.19). From Figure 6.4, it is apparent that for $\varepsilon < 10^{-1}$ the SUR method results

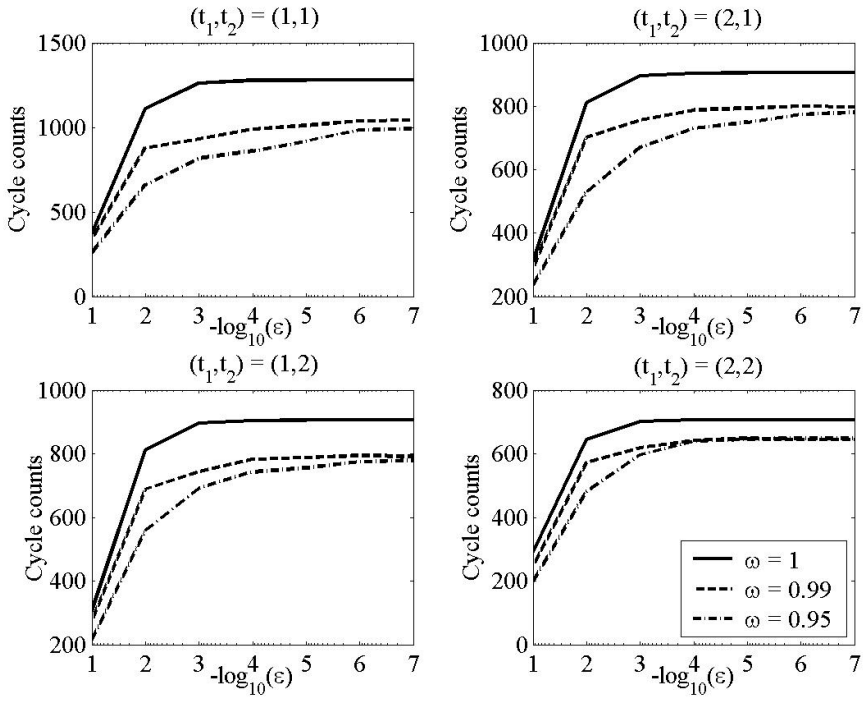

Figure 6.3: Cycle counts of the MMG method for $N = 64$ and varying values of ε for the test problem (6.19).

in the fastest execution times.

Numerical observations

From these numerical experiments we observe that the MMG method is uniformly convergent in ε in its cycle counts. For the convection-diffusion problem with parabolic boundary layers, the relaxation parameter $\omega = 1$ is optimal. However, $\omega = 1$ is not optimal for the anisotropic convection-diffusion problem. For both the convection-diffusion problem with parabolic boundary layers and the anisotropic convection-diffusion problem, the SUR method results in the fastest execution times than the MMG method. For both of the test problems, through the numerical experiments, it is observed that the prolongation parameter $\rho_k^{(n)}$ as given in (6.9) is extremely small. This is the reason why the SUR method results in the fastest execution times and why there is no advantage in using the MMG method.

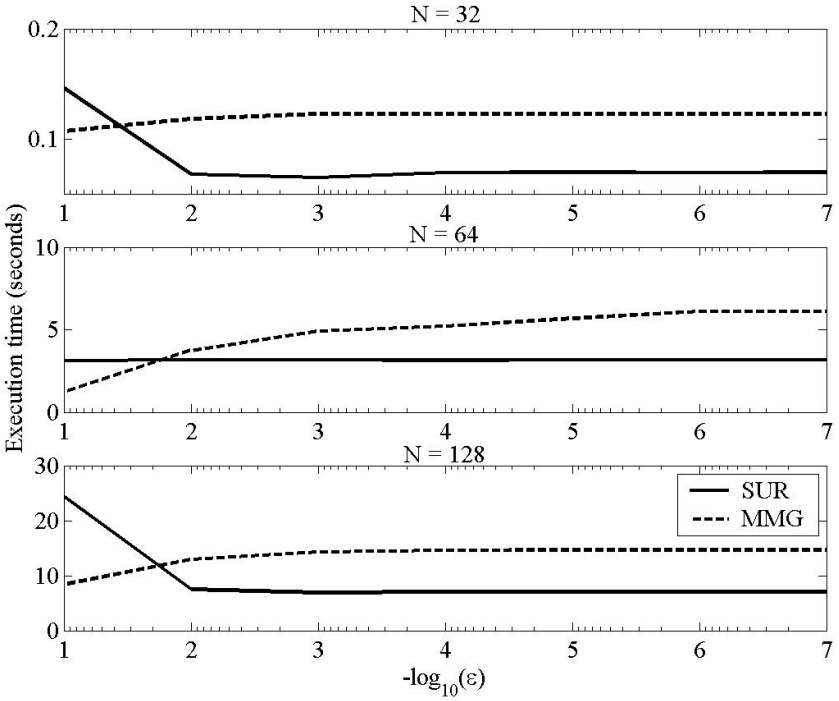

Figure 6.4: Execution times of the MMG and SUR methods for $N = 64$ and varying values of ε for the test problem (6.19).

6.3 Block monotone multigrid method

In this section, we investigate the block monotone multigrid method for solving nonlinear singularly perturbed convection-diffusion problems. Similar to the previous section, we represent the nonlinear difference scheme (1.15) on Ω_k^h, $k = 0, \ldots, K$, in the form (6.3). On the finest mesh Ω_K^h, choose an initial function $v_K^{(0)}(p)$, $p \in \Omega_K^h$, satisfying the boundary condition $v_K^{(0)}(p) = g(p)$, $p \in \partial\Omega_K^h$, and which is an upper or lower solution to (6.3).

As (1.15) is the nonlinear problem, the full approximation scheme (FAS) multigird cycle from [27] is used. The algorithm for the block monotone multigrid method (BMMG) is as follows:

FAS block monotone multigrid cycle

$$v_k^{(n+1)} = BMMG\{k, v_k^{(n)}, \mathcal{L}_k, \phi_k^{(n)}, t_1, t_2\}$$

$\phi_K^{(n)} = 0$ on the finest mesh Ω_K^h.

1. Presmoothing:

$$v_k^{(n,t)} = BS_k(v_k^{(n,t-1)}, \mathcal{L}_k^{\text{BSUR}}, \phi_k^{(n)}, t_1), \quad t = 1, \ldots, t_1,$$

where $v_k^{(n,0)} = v_k^{(n)}$, the mapping BS_k defines the monotone BSUR method (4.15) on Ω_k^h, and t_1 is the number of presmoothing steps.

2. Coarse-grid correction:

(a) On Ω_k^h, compute the residual

$$\mathcal{R}_k(v_k^{(n,t_1)}) = \mathcal{L}_k v_k^{(n,t_1)} + f(v_k^{(n,t_1)}).$$

(b) Restrict $v_k^{(n,t_1)}$ and $\mathcal{R}_k(v_k^{(n,t_1)})$ on the coarse mesh Ω_{k-1}^h

$$v_{k-1}^{(n)} = I_k^{k-1} v_k^{(n,t_1)}, \quad r_{k-1}^{(n)} = I_k^{k-1} \mathcal{R}_k(v_k^{(n,t_1)}), \tag{6.21}$$

where I_k^{k-1} is a restriction operator.

(c) On Ω_{k-1}^h, solve the nonlinear difference scheme

$$\mathcal{L}_{k-1} w_{k-1}^{(n)} + f(w_{k-1}^{(n)}) = \phi_{k-1}^{(n)}, \quad \phi_{k-1}^{(n)} = \mathcal{R}_{k-1}(v_{k-1}^{(n)}) - r_{k-1}^{(n)}, \tag{6.22}$$

$$w_{k-1}^{(n)}(p) = g(p), \quad p \in \partial\Omega_{k-1}^h.$$

If $k = 1$ then solve (6.22) as exactly as necessary, by using the monotone BSUR method (4.15) on Ω_0^h. If $k > 1$ then solve (6.22) by the algorithm

$$w_{k-1}^{(n)} = BMMG\{k-1, w_{k-1}^{(n,0)}, \mathcal{L}_{k-1}, \phi_{k-1}^{(n)}, t_1, t_2\},$$

with the initial iterate $w_{k-1}^{(n,0)} = v_{k-1}^{(n)}$, where $v_{k-1}^{(n)}$ is defined in (6.21).

(d) Prolong the error $e_{k-1}^{(n)} = w_{k-1}^{(n)} - v_{k-1}^{(n)}$ on the fine mesh Ω_k^h

$$e_k^{(n)} = I_{k-1}^k (w_{k-1}^{(n)} - v_{k-1}^{(n)}),$$

where I_{k-1}^k is a prolongation operator.

(e) Correct the approximation $v_k^{(n,t_1)}$ on the fine mesh Ω_k^h

$$w_k^{(n)} = v_k^{(n,t_1)} + \rho_k^{(n)} \widetilde{e}_k^{(n)}, \tag{6.23}$$

$$\widetilde{e}_k^{(n)}(p) = \begin{cases} 0, & p \in \widehat{\Omega}_k^h = \{p : \mathcal{R}_k(p, v_k^{(n,t_1)}) = 0\}, \\ e_k^{(n)}(p), & p \in \widetilde{\Omega}_k^h = \{p : \mathcal{R}_k(p, v_k^{(n,t_1)}) \neq 0\}, \end{cases}$$

where a prolongation parameter $\rho_k^{(n)} > 0$ is chosen in (6.9).

3. Postsmoothing:

$$v_k^{(n,t)} = BS_k\{v_k^{(n,t-1)}, \mathcal{L}_k^{\text{BSUR}}, \phi_k^{(n)}, t_2\}, \quad t = t_1 + 2, \ldots, t_1 + t_2 + 1, \tag{6.24}$$

with the initial iterate $v_k^{(n,t_1+1)} = w_k^{(n)}$, where $w_k^{(n)}$ from (6.23), BS_k replaces the monotone BSUR method (4.15) on Ω_k^h, and t_2 is the number of postsmoothing steps. On Ω_k^h, set up $v_k^{(n+1)} = v_k^{(n,t_1+t_2+1)}$.

We assume that I_k^{k-1} and I_{k-1}^k are linear and monotone.

Theorem 26. *Let the coefficients of the difference operators from (6.3) satisfy (1.16). If* $\overline{v}_K^{(0)}$ *and* $\underline{v}_K^{(0)}$ *are upper and lower solutions of the difference scheme (6.3), then the upper sequence* $\{\overline{v}_K^{(n)}\}$ *obtained by the block monotone multigrid method converges monotonically from above to the unique solution* v_K *of (6.3), the lower sequence* $\{\underline{v}_K^{(n)}\}$ *converges monotonically from below to* v_K*:*

$$\underline{v}_K^{(n-1)}(p) \leq \underline{v}_K^{(n)}(p) \leq v_K(p) \leq \overline{v}_K^{(n)}(p) \leq \overline{v}_K^{(n-1)}(p), \quad p \in \overline{\Omega}_K^h, \quad n \geq 1.$$

Proof. The proof of this theorem follows directly from the proof of Theorems 12 and 25. □

6.3.1 Numerical experiments

We apply the BMMG to the two test problems: the convection-diffusion problem with parabolic layers (1.9) and the anisotropic convection-diffusion problem (1.12). For the following numerical experiments, the stopping criteria for the iterates is

$$\max_{p \in \overline{\Omega}^h} \|v^{(n)}(p) - v^{(n-1)}(p)\| \leq 10^{-6}.$$

The number of mesh points in the x- and y-directions are set equal to N. For the restriction and prolongation operators, we use the full weighting and bilinear interpolation operators, respectively.

Convection-diffusion problem with parabolic boundary layers

As a test problem, we consider the convection-diffusion problem with parabolic layers (6.17). To solve this problem numerically, we use the piecewise uniform Shishkin mesh (1.11). We require the constants c_* and c^*, where from (6.18) we have

$$c_* = \exp(-1), \quad c^* = 1.$$

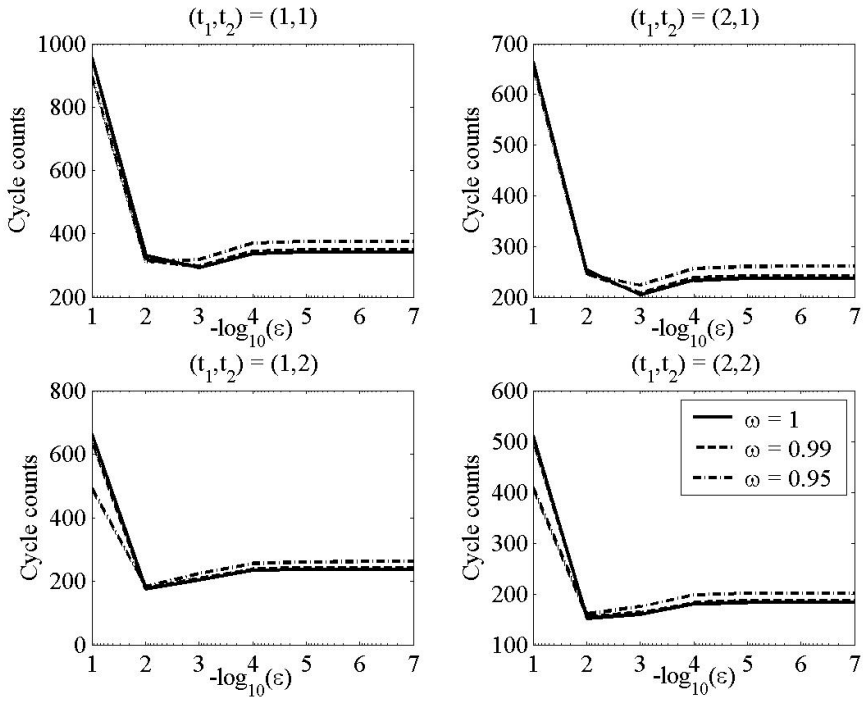

Figure 6.5: Cycle counts of the BMMG method for $N = 64$ and varying values of ε for the test problem (6.17).

Figure 6.5 displays the four graphs with the cycle counts of the BMMG method over varying values of ω and ε for the test problem (6.17). Each graph corresponds to different numbers t_1, t_2 of presmoothing and postsmoothing steps, respectively. All the graphs indicate that for $\epsilon < 10^{-4}$ the cycle counts are independent of ε, leading to the conclusion that the BMMG method converges parameter uniformly in its cycle counts. The graphs

also show that the relaxation parameter $\omega = 1$ is optimal for the BMMG method. Figure

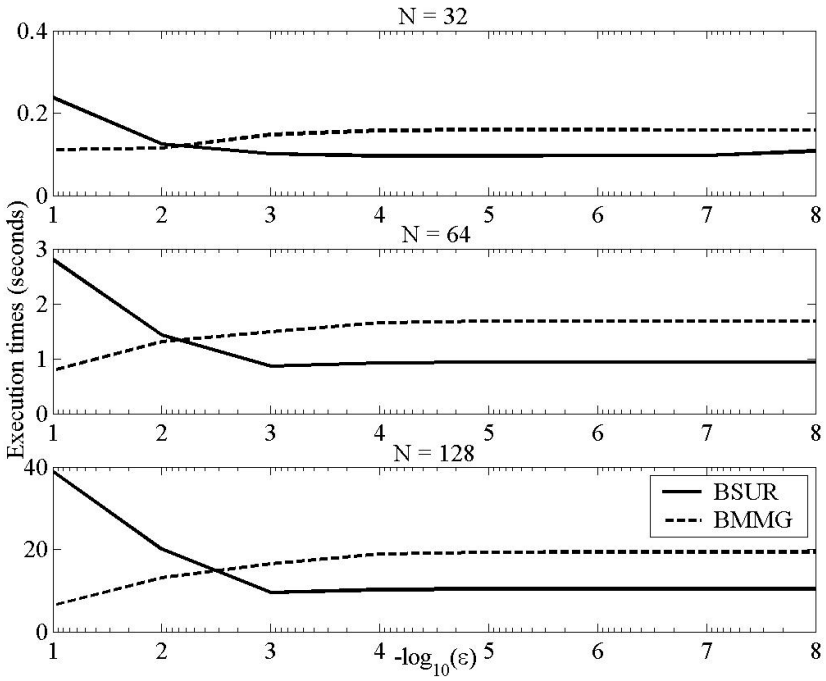

Figure 6.6: Execution times of the BMMG and BSUR methods for varying values of ε for the test problem (6.17).

6.6 displays the execution times of the BMMG and BSUR methods for varying values of ε for the test problem (6.17). From Figure 6.6, it is evident that for $\varepsilon < 10^{-2}$ the BSUR method has the fastest acceleration time, indicating there is no advantage in using the BMMG method.

Anisotropic convection-diffusion problem

The test problem here is the anisotropic convection-diffusion problem (6.19). To solve this problem numerically, we use the piecewise uniform Shishkin mesh (1.8). We require the constants c_* and c^*, where from (6.20), we have

$$c_* = \exp(-1), \quad c^* = 1.$$

Figure 6.7 displays the four graphs with the cycle counts of the BMMG method over varying values of ω and ε for the test problem (6.19). Each graph corresponds to different numbers t_1, t_2 of presmoothing and postsmoothing steps, respectively. All the four graphs

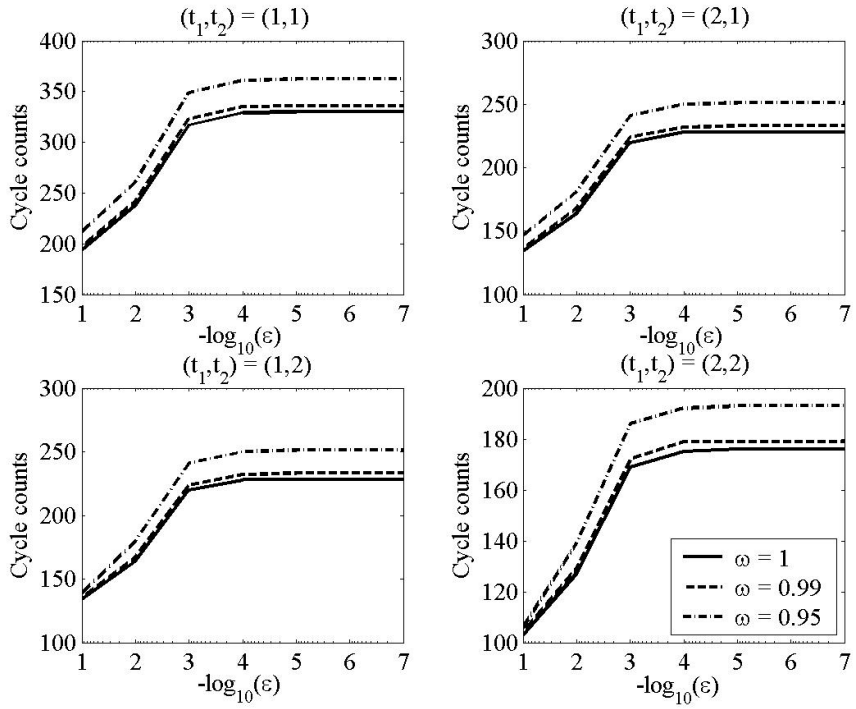

Figure 6.7: Cycle counts of the BMMG method for $N = 64$ and varying values of ε for the test problem (6.19).

indicate that for $\epsilon < 10^{-4}$ the cycle counts are independent of ε. This leads to the conclusion that the BMMG method converges parameter uniformly in its cycle counts. The graphs also show that the relaxation parameter $\omega = 1$ is optimal for the BMMG method. Figure 6.8 displays the execution times of the BMMG and BSUR methods for varying values of ε for the test problem (6.19). From Figure 6.8, it is evident that for all values of ε the BSUR method has the fastest acceleration times, indicating there is no advantage in using the BMMG method.

Numerical observations

From these numerical experiments, we observe that the BMMG method is uniformly convergent in ε in its cycle counts. The relaxation parameter $\omega = 1$ is optimal for both of the test problems. For both the convection-diffusion problem with parabolic boundary layers and the anisotropic convection-diffusion problem, the BSUR method results in the fastest execution time than the BMMG method for all values of ε. For both of the test problems, through the numerical experiments it is observed that the prolongation pa-

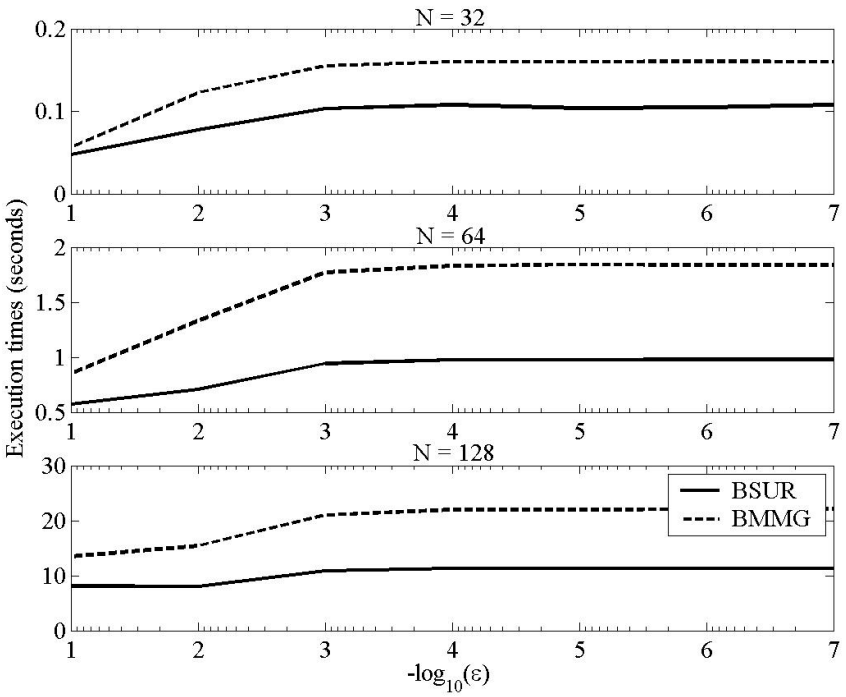

Figure 6.8: Execution times of the BMMG and BSUR methods for varying values of ε for the test problem (6.19).

rameter $\rho_k^{(n)}$ as given in (6.9) is extremely small. This provides an explanation why the BSUR method results in the fastest execution time and why there is no advantage in using the BMMG method.

6.4 Two-level monotone multigrid method

In this section, we investigate the two-level monotone multigrid algorithm for solving nonlinear singularly perturbed convection-diffusion problems. Similar to the previous sections, we represent the nonlinear difference scheme (1.15) on Ω_k^h, $k = 0, \ldots, K$, in the form (6.3). On the finest mesh Ω_K^h, choose an initial function $v_K^{(0)}(p)$, $p \in \Omega_K^h$, satisfying the boundary condition $v_K^{(0)}(p) = g(p)$, $p \in \partial\Omega_K^h$, and which is an upper or lower solution to (6.3).

As (1.15) is the nonlinear problem, the full approximation scheme (FAS) multigrid cycle [27] is used. The algorithm for the two-level monotone multigrid algorithm (TLMMG) is as follows:

FAS two-level monotone multigrid cycle

$$v_k^{(n+1)} = TLMG\{k, v_k^{(n)}, \mathcal{L}_k, \phi_k^{(n)}, t_1, t_2\}$$

$\phi_K^{(n)} = 0$ on the finest mesh Ω_K^h.

1. Presmoothing:

$$v_k^{(n,t)} = DD_k(v_k^{(n,t-1)}, \mathcal{L}_k^{\mathrm{DD}}, \phi_k^{(n)}, t_1), \quad t = 1, \ldots, t_1,$$

where $v_k^{(n,0)} = v_k^{(n)}$, the mapping DD_k defines the monotone domain decomposition algorithm on Ω_k^h, and t_1 is the number of presmoothing steps.

2. Coarse-grid correction:

(a) On Ω_k^h, compute the residual

$$\mathcal{R}_k(v_k^{(n,t_1)}) = \mathcal{L}_k v_k^{(n,t_1)} + f(v_k^{(n,t_1)}).$$

(b) Restrict $v_k^{(n,t_1)}$ and $\mathcal{R}_k(v_k^{(n,t_1)})$ on the coarse mesh Ω_{k-1}^h

$$v_{k-1}^{(n)} = I_k^{k-1} v_k^{(n,t_1)}, \quad r_{k-1}^{(n)} = I_k^{k-1} \mathcal{R}_k(v_k^{(n,t_1)}), \tag{6.25}$$

where I_k^{k-1} is a restriction operator.

(c) On Ω_{k-1}^h, solve the nonlinear difference scheme

$$\mathcal{L}_{k-1} w_{k-1}^{(n)} + f(w_{k-1}^{(n)}) = \phi_{k-1}^{(n)}, \quad \phi_{k-1}^{(n)} = \mathcal{R}_{k-1}(v_{k-1}^{(n)}) - r_{k-1}^{(n)}, \tag{6.26}$$

$$w_{k-1}^{(n)}(p) = g(p), \quad p \in \partial\Omega_{k-1}^h.$$

If $k = 1$ then solve (6.26) as exactly as necessary, by using the two-level monotone domain decomposition algorithm (3.10–5.14) on Ω_0^h. If $k > 1$ then solve (6.26) by the algorithm

$$w_{k-1}^{(n)} = TLMG\{k-1, w_{k-1}^{(n,0)}, \mathcal{L}_{k-1}, \phi_{k-1}^{(n)}, t_1, t_2\},$$

with the initial iterate $w_{k-1}^{(n,0)} = v_{k-1}^{(n)}$, where $v_{k-1}^{(n)}$ is defined in (6.25).

(d) Prolong the error $e_{k-1}^{(n)} = w_{k-1}^{(n)} - v_{k-1}^{(n)}$ on the fine mesh Ω_k^h

$$e_k^{(n)} = I_{k-1}^k (w_{k-1}^{(n)} - v_{k-1}^{(n)}),$$

where I_{k-1}^k is a prolongation operator.

(e) Correct the approximation $v_k^{(n,t_1)}$

$$w_k^{(n)} = v_k^{(n,t_1)} + \rho_k^{(n)} \widetilde{e}_k^{(n)}, \tag{6.27}$$

$$\widetilde{e}_k^{(n)}(p) = \begin{cases} 0, & p \in \widehat{\Omega}_k^h = \{p : \mathcal{R}_k(p, v_k^{(n,t_1)}) = 0\}, \\ e_k^{(n)}(p), & p \in \widetilde{\Omega}_k^h = \{p : \mathcal{R}_k(p, v_k^{(n,t_1)}) \neq 0\}. \end{cases}$$

where a prolongation parameter $\rho_k^{(n)} > 0$ is chosen in (6.9).

3. Postsmoothing:

$$v_k^{(n,t)} = DD_k\{v_k^{(n,t-1)}, \mathcal{L}_k^{\mathrm{DD}}, \phi_k^{(n)}, t_2\}, \quad t = t_1 + 2, \ldots, t_1 + t_2 + 1, \tag{6.28}$$

with the initial iterate $v_k^{(n,t_1+1)} = w_k^{(n)}$, where $w_k^{(n)}$ from (6.27), DD_k replaces the monotone domain decomposition algorithm (3.10–5.13) on Ω_k^h, and t_2 is the number of postsmoothing steps. On Ω_k^h, set up $v_k^{(n+1)} = v_k^{(n,t_1+t_2+1)}$.

Theorem 27. *Let the coefficients of the difference operators from (6.3) satisfy (1.16). If $\overline{v}_K^{(0)}$ and $\underline{v}_K^{(0)}$ are upper and lower solutions of the difference scheme (6.3), then the upper sequence $\{\overline{v}_K^{(n)}\}$ obtained by the two-level monotone multigrid algorithm converges monotonically from above to the unique solution v_K of (6.3), the lower sequence $\{\underline{v}_K^{(n)}\}$ converges monotonically from below to v_K:*

$$\underline{v}_K^{(n-1)}(p) \leq \underline{v}_K^{(n)}(p) \leq v_K(p) \leq \overline{v}_K^{(n)}(p) \leq \overline{v}_K^{(n-1)}(p), \quad p \in \overline{\Omega}_K^h, \quad n \geq 1.$$

Proof. The proof of this theorem follows directly from the proof of Theorem 6 with $S = 1$ and Theorem 25. □

6.4.1 Numerical experiments

We apply the two-level monotone multigrid algorithm to the two test problems: the convection-diffusion problem with parabolic layers (6.17) and the anisotropic convection-

diffusion problem (6.19). For the following numerical experiments, the stopping criteria for the iterates is

$$\max_{p\in\overline{\Omega}^h} \|v^{(n)}(p) - v^{(n-1)}(p)\| \leq 10^{-6}.$$

The number of mesh points in the x- and y-directions are set equal to N. For the restriction and prolongation operators, we use the full weighting and bilinear interpolation operators, respectively.

To solve the linear difference problems within the monotone domain decomposition algorithm, GMRES with restarts is used with a diagonal preconditioner. Within GMRES the required accuracy is 10^{-6}, the maximum number of iterations is 50 and a maximum of 20 restarts.

For the following numerical experiments, the balanced domain decomposition is used, where the number of subdomains is even with half of the subdomains placed within the boundary layers.

The computational domain is decomposed into M overlapping subdomains as described in Section 3.2.1, where the overlap between the vertical strips is chosen such that for the two vertical strips either side of the boundary layer, the overlap occurs outside of the boundary layer, as described in Section 3.2.1.

Convection-diffusion problem with parabolic boundary layers

The test problem here is the convection-diffusion problem with parabolic layers (6.17). To solve this problem numerically, we use piecewise uniform Shishkin mesh (1.11). We require the constants c_* and c^*, where from (6.18) we have

$$c_* = \exp(-1), \quad c^* = 1.$$

We also require an initial solution. For the test problem (6.17), $\overline{u}^{(0)}(p) = 1$, $p \in \Omega^h$, $\overline{u}^{(0)}(p) = 0$, $p \in \partial\Omega^h$ is an initial upper solution.

Tables 6.1–6.4 display the cycle counts and execution times for the TLMMG algorithm with $(t_1, t_2) = (1,1), (2,1), (1,2), (2,2)$, respectively, where t_1 and t_2 are the number of presmoothing and postsmoothing steps, respectively. From these tables, we observe that for $\varepsilon < 10^{-3}$ the cycle counts are independent of ε. Similar results are observed for numerical experiments for varying numbers of mesh points. This indicates that the TLMMG algorithm converges parameter uniformly with respect to its cycle counts. We also

ε	10^{-2}				10^{-3}				10^{-4}			
$S\backslash M$	1	2	4	8	1	2	4	8	1	2	4	8
	Cycle counts											
1	2	$\frac{5}{2}$	$\frac{25}{3}$	$\frac{38}{7}$	2	$\frac{3}{2}$	$\frac{29}{4}$	$\frac{47}{9}$	2	$\frac{3}{2}$	$\frac{29}{4}$	$\frac{48}{9}$
2	$\frac{10}{2}$	$\frac{12}{2}$	$\frac{28}{3}$	$\frac{41}{7}$	$\frac{9}{2}$	$\frac{8}{2}$	$\frac{30}{4}$	$\frac{48}{9}$	$\frac{6}{2}$	$\frac{6}{2}$	$\frac{29}{4}$	$\frac{49}{9}$
4	$\frac{15}{2}$	$\frac{16}{3}$	$\frac{29}{3}$	$\frac{42}{7}$	$\frac{14}{2}$	$\frac{11}{2}$	$\frac{30}{4}$	$\frac{49}{9}$	$\frac{8}{2}$	$\frac{7}{2}$	$\frac{29}{4}$	$\frac{49}{9}$
8	$\frac{28}{7}$	$\frac{28}{7}$	$\frac{33}{6}$	$\frac{46}{10}$	$\frac{25}{6}$	$\frac{18}{6}$	$\frac{31}{5}$	$\frac{51}{10}$	$\frac{14}{4}$	$\frac{11}{4}$	$\frac{30}{4}$	$\frac{49}{9}$
	Execution times (seconds)											
1	7	$\frac{720}{9}$	$\frac{4167}{728}$	$\frac{3476}{1105}$	7	$\frac{321}{9}$	$\frac{3555}{912}$	$\frac{2811}{941}$	4	$\frac{150}{6}$	$\frac{2751}{629}$	$\frac{2049}{723}$
2	$\frac{87}{9}$	$\frac{1157}{700}$	$\frac{2434}{1224}$	$\frac{2492}{2294}$	$\frac{77}{10}$	$\frac{678}{743}$	$\frac{2313}{1636}$	$\frac{2128}{2771}$	$\frac{46}{6}$	$\frac{486}{877}$	$\frac{1914}{1569}$	$\frac{1621}{2459}$
4	$\frac{65}{17}$	$\frac{908}{510}$	$\frac{1208}{528}$	$\frac{1350}{921}$	$\frac{72}{18}$	$\frac{636}{390}$	$\frac{1241}{774}$	$\frac{1451}{1249}$	$\frac{47}{11}$	$\frac{622}{493}$	$\frac{1161}{879}$	$\frac{1089}{1162}$
8	$\frac{55}{29}$	$\frac{653}{454}$	$\frac{701}{424}$	$\frac{1030}{600}$	$\frac{69}{35}$	$\frac{600}{580}$	$\frac{795}{463}$	$\frac{1196}{714}$	$\frac{51}{26}$	$\frac{576}{547}$	$\frac{796}{475}$	$\frac{964}{635}$

Table 6.1: Cycle counts and execution times of the two-level monotone multigrid algorithm using the minimal and maximal overlap sizes for both S and M above and below the line, respectively, with $(t_1, t_2) = (1, 1)$ and $N = 64$ for the test problem (6.17).

observe that as the number of presmoothing and/or postsmoothing iterations increase, the cycle count decreases.

Table 6.5 displays the iteration counts and execution times for the two-level domain decomposition (DD) algorithm (3.10)–(3.14). From Table 6.5 for $\varepsilon < 10^{-2}$, the iteration counts are independent of ε. This is observed for varying numbers of mesh points, indicating that the two-level domain decomposition algorithm is parameter uniform with respect to its iteration counts. By comparing the execution times in Table 6.5 with the execution times in Tables 6.1– 6.4, it is observed that the two-level domain decomposition algorithm results in the smallest execution times for all combinations of M and S and values of ε. Therefore, there is no advantage in using the two-level monotone multigrid algorithm.

Anisotropic convection-diffusion problem

The test problem here is the anisotropic convection-diffusion problem (6.19). To solve this problem numerically, we use piecewise uniform Shishkin mesh (1.8). We require the

ε	10^{-2}				10^{-3}				10^{-4}			
$S\backslash M$	1	2	4	8	1	2	4	8	1	2	4	8
	Cycle counts											
1	2	$\frac{4}{2}$	$\frac{17}{3}$	$\frac{26}{5}$	2	$\frac{3}{2}$	$\frac{20}{3}$	$\frac{33}{6}$	2	$\frac{2}{2}$	$\frac{20}{3}$	$\frac{33}{7}$
2	$\frac{10}{2}$	$\frac{11}{2}$	$\frac{21}{3}$	$\frac{30}{5}$	$\frac{10}{2}$	$\frac{7}{2}$	$\frac{21}{3}$	$\frac{34}{7}$	$\frac{6}{2}$	$\frac{5}{2}$	$\frac{21}{3}$	$\frac{34}{7}$
4	$\frac{16}{2}$	$\frac{15}{3}$	$\frac{22}{3}$	$\frac{32}{6}$	$\frac{14}{2}$	$\frac{10}{2}$	$\frac{21}{3}$	$\frac{35}{7}$	$\frac{9}{2}$	$\frac{7}{2}$	$\frac{21}{3}$	$\frac{34}{7}$
8	$\frac{30}{7}$	$\frac{27}{7}$	$\frac{27}{6}$	$\frac{39}{9}$	$\frac{26}{6}$	$\frac{17}{6}$	$\frac{23}{4}$	$\frac{37}{8}$	$\frac{15}{4}$	$\frac{11}{4}$	$\frac{21}{3}$	$\frac{35}{7}$
	Execution times (seconds)											
1	9	$\frac{729}{12}$	$\frac{5347}{678}$	$\frac{4493}{1248}$	9	$\frac{334}{13}$	$\frac{4771}{973}$	$\frac{3645}{1242}$	6	$\frac{157}{8}$	$\frac{3516}{929}$	$\frac{2685}{1098}$
2	$\frac{96}{11}$	$\frac{1061}{706}$	$\frac{3228}{2460}$	$\frac{3150}{3288}$	$\frac{85}{11}$	$\frac{475}{743}$	$\frac{2949}{2458}$	$\frac{2672}{4319}$	$\frac{47}{7}$	$\frac{256}{874}$	$\frac{2481}{2389}$	$\frac{2011}{3830}$
4	$\frac{68}{18}$	$\frac{726}{512}$	$\frac{1603}{1039}$	$\frac{1726}{1551}$	$\frac{73}{18}$	$\frac{434}{388}$	$\frac{1617}{1150}$	$\frac{1807}{1935}$	$\frac{48}{12}$	$\frac{621}{487}$	$\frac{1503}{1328}$	$\frac{1381}{1804}$
8	$\frac{56}{30}$	$\frac{468}{455}$	$\frac{965}{829}$	$\frac{1354}{1062}$	$\frac{71}{36}$	$\frac{386}{580}$	$\frac{1047}{720}$	$\frac{1445}{1140}$	$\frac{52}{27}$	$\frac{579}{549}$	$\frac{1040}{689}$	$\frac{1247}{979}$

Table 6.2: Cycle counts and execution times of the two-level monotone multigrid algorithm using the minimal and maximal overlap sizes for both S and M above and below the line, respectively, with $(t_1, t_2) = (2, 1)$ and $N = 64$ for the test problem (6.17).

constants c_* and c^*, where from (6.20) we have

$$c_* = \exp(-1), \quad c^* = 1.$$

We also require an initial solution. For the test problem (6.19), $\overline{u}^{(0)}(p) = 1$, $p \in \Omega^h$, $\overline{u}^{(0)}(p) = 0$, $p \in \partial\Omega^h$ is an initial upper solution.

Tables 6.6–6.9 display the cycle counts and execution times for the TLMMG algorithm with $(t_1, t_2) = (1, 1), (2, 1), (1, 2), (2, 2)$, respectively, where t_1 and t_2 are the number of presmoothing and postsmoothing iterative steps, respectively. From these tables, we observe that for $\varepsilon < 10^{-2}$ the cycle counts are independent of ε. Through numerical experiments, we observe similar results for varying numbers of mesh points. This indicates that the TLMMG algorithm converges parameter uniform with respect to its cycle count. As the number of presmoothing and/or postsmoothing iterations increases, the cycle counts decrease.

Table 6.10 displays the iteration counts and execution times for the two-level domain decomposition (DD) algorithm (3.10)–(3.14). From Table 6.10 for $\varepsilon < 10^{-2}$, the iteration count is independent of ε, indicating that the two-level domain decomposition is param-

ε	10^{-2}				10^{-3}				10^{-4}			
$S\backslash M$	1	2	4	8	1	2	4	8	1	2	4	8
	Cycle counts											
1	2	$\frac{4}{2}$	$\frac{17}{3}$	$\frac{26}{5}$	2	$\frac{3}{2}$	$\frac{20}{3}$	$\frac{33}{6}$	2	$\frac{2}{2}$	$\frac{20}{3}$	$\frac{33}{7}$
2	$\frac{10}{2}$	$\frac{11}{2}$	$\frac{21}{3}$	$\frac{30}{5}$	$\frac{10}{2}$	$\frac{7}{2}$	$\frac{21}{3}$	$\frac{34}{7}$	$\frac{6}{2}$	$\frac{5}{2}$	$\frac{21}{3}$	$\frac{34}{7}$
4	$\frac{16}{2}$	$\frac{15}{3}$	$\frac{22}{3}$	$\frac{32}{6}$	$\frac{14}{2}$	$\frac{10}{2}$	$\frac{21}{3}$	$\frac{35}{7}$	$\frac{9}{2}$	$\frac{7}{2}$	$\frac{21}{3}$	$\frac{34}{7}$
8	$\frac{30}{7}$	$\frac{27}{7}$	$\frac{27}{6}$	$\frac{39}{9}$	$\frac{26}{6}$	$\frac{17}{6}$	$\frac{23}{4}$	$\frac{37}{8}$	$\frac{15}{4}$	$\frac{11}{4}$	$\frac{21}{3}$	$\frac{35}{7}$
	Execution times (seconds)											
1	8	$\frac{547}{11}$	$\frac{2716}{360}$	$\frac{2337}{812}$	9	$\frac{167}{12}$	$\frac{2407}{601}$	$\frac{1882}{657}$	6	$\frac{157}{8}$	$\frac{1837}{394}$	$\frac{1408}{587}$
2	$\frac{97}{11}$	$\frac{862}{720}$	$\frac{1754}{1233}$	$\frac{1714}{1646}$	$\frac{87}{11}$	$\frac{478}{762}$	$\frac{1600}{1235}$	$\frac{1505}{2160}$	$\frac{51}{7}$	$\frac{375}{894}$	$\frac{1354}{1185}$	$\frac{1086}{1922}$
4	$\frac{69}{17}$	$\frac{674}{513}$	$\frac{837}{528}$	$\frac{930}{792}$	$\frac{74}{18}$	$\frac{436}{388}$	$\frac{856}{580}$	$\frac{983}{969}$	$\frac{47}{12}$	$\frac{624}{493}$	$\frac{816}{663}$	$\frac{751}{909}$
8	$\frac{56}{30}$	$\frac{467}{454}$	$\frac{495}{424}$	$\frac{714}{540}$	$\frac{70}{35}$	$\frac{422}{578}$	$\frac{544}{372}$	$\frac{809}{573}$	$\frac{53}{27}$	$\frac{577}{550}$	$\frac{563}{360}$	$\frac{653}{495}$

Table 6.3: Cycle counts and execution times of the two-level monotone multigrid algorithm using the minimal and maximal overlap sizes for both S and M above and below the line, respectively, with $(t_1, t_2) = (1, 2)$ and $N = 64$ for the test problem (6.17).

eter uniform with respect to its iteration counts. By comparing the execution times in Table 6.10 with the execution times in Tables 6.6–6.9, it is observed that the two-level domain decomposition algorithm (3.10)–(3.14) results in the smallest execution times for all combinations of M and S and values of ε. Therefore, there is no advantage in using the two-level monotone multigrid algorithm.

Numerical observations

From these numerical experiments, we observe that the two-level monotone multigrid algorithm is parameter uniformly convergent with respect to its cycle count. For both of the test problems, the two-level domain decomposition algorithm (3.10)–(3.14) results in the fastest execution times. This indicates no advantage in using the two-level monotone multigrid algorithm for these test problems.

ε	10^{-2}				10^{-3}				10^{-4}			
$S\backslash M$	1	2	4	8	1	2	4	8	1	2	4	8
	Cycle counts											
1	2	$\frac{3}{2}$	$\frac{14}{2}$	$\frac{20}{4}$	2	$\frac{2}{2}$	$\frac{16}{3}$	$\frac{25}{5}$	2	$\frac{2}{2}$	$\frac{16}{3}$	$\frac{26}{5}$
2	$\frac{10}{2}$	$\frac{10}{2}$	$\frac{17}{2}$	$\frac{24}{4}$	$\frac{9}{2}$	$\frac{7}{2}$	$\frac{16}{3}$	$\frac{27}{5}$	$\frac{6}{2}$	$\frac{5}{2}$	$\frac{16}{3}$	$\frac{26}{5}$
4	$\frac{15}{2}$	$\frac{14}{2}$	$\frac{18}{3}$	$\frac{27}{5}$	$\frac{13}{2}$	$\frac{9}{2}$	$\frac{17}{3}$	$\frac{27}{5}$	$\frac{8}{2}$	$\frac{6}{2}$	$\frac{16}{3}$	$\frac{26}{5}$
8	$\frac{28}{7}$	$\frac{24}{7}$	$\frac{24}{6}$	$\frac{34}{8}$	$\frac{24}{6}$	$\frac{15}{6}$	$\frac{18}{4}$	$\frac{29}{6}$	$\frac{14}{4}$	$\frac{10}{4}$	$\frac{16}{3}$	$\frac{27}{6}$
	Execution times (seconds)											
1	11	$\frac{553}{15}$	$\frac{4074}{676}$	$\frac{3437}{1020}$	9	$\frac{319}{13}$	$\frac{3533}{651}$	$\frac{2820}{943}$	7	$\frac{151}{9}$	$\frac{2742}{695}$	$\frac{2071}{869}$
2	$\frac{102}{12}$	$\frac{741}{705}$	$\frac{2534}{1648}$	$\frac{2393}{2625}$	$\frac{81}{11}$	$\frac{368}{745}$	$\frac{2316}{2458}$	$\frac{2113}{3083}$	$\frac{51}{8}$	$\frac{258}{880}$	$\frac{1888}{2397}$	$\frac{1566}{2744}$
4	$\frac{66}{17}$	$\frac{550}{345}$	$\frac{1245}{1041}$	$\frac{1332}{1296}$	$\frac{72}{18}$	$\frac{369}{388}$	$\frac{1239}{1148}$	$\frac{1375}{1380}$	$\frac{46}{12}$	$\frac{534}{488}$	$\frac{1141}{1331}$	$\frac{1073}{1297}$
8	$\frac{60}{32}$	$\frac{375}{460}$	$\frac{757}{831}$	$\frac{1031}{947}$	$\frac{72}{37}$	$\frac{316}{583}$	$\frac{796}{721}$	$\frac{1099}{859}$	$\frac{54}{28}$	$\frac{529}{551}$	$\frac{806}{691}$	$\frac{940}{842}$

Table 6.4: Cycle counts and execution times of the two-level monotone multigrid algorithm using the minimal and maximal overlap sizes for both S and M above and below the line, respectively, with $(t_1, t_2) = (2, 2)$ and $N = 64$ for the test problem (6.17).

6.5 Conclusions

In this chapter, we investigate the monotone multigrid methods for solving nonlinear singularly perturbed convection-diffusion problems. In Section 6.2, we construct the monotone multigrid method and in Theorem 25 prove the monotonic convergence. The numerical experiments in Section 6.2.1 indicate that for both of the test problems the monotone multigrid method convergence parameter uniformly with respect to its cycle count. For the convection-diffusion problem with parabolic boundary layers, the relaxation parameter $\omega = 1$ is optimal. However, $\omega = 1$ is not optimal for the anisotropic convection-diffusion problem. For both of the convection-diffusion problem with parabolic boundary layers and the anisotropic convection-diffusion problem, the successive underrelaxation method results in faster execution times than the monotone multigrid method. For both of the test problems, through the numerical experiments, it is observed that the prolongation parameter $\rho_k^{(n)}$ as given in (6.9) is extremely small. This is the reason why the successive underrelaxation method results in the fastest execution times and why there is no advantage in using the monotone multigrid method.

ε	10^{-2}				10^{-3}				10^{-4}			
$S\backslash M$	1	2	4	8	1	2	4	8	1	2	4	8
	Iteration counts											
1	2	$\frac{9}{2}$	$\frac{46}{5}$	$\frac{71}{13}$	2	$\frac{5}{2}$	$\frac{53}{6}$	$\frac{88}{16}$	2	$\frac{4}{2}$	$\frac{54}{6}$	$\frac{90}{17}$
2	$\frac{2}{2}$	$\frac{9}{2}$	$\frac{47}{5}$	$\frac{72}{13}$	$\frac{2}{2}$	$\frac{5}{2}$	$\frac{54}{6}$	$\frac{89}{16}$	$\frac{2}{2}$	$\frac{4}{2}$	$\frac{55}{6}$	$\frac{91}{17}$
4	$\frac{2}{2}$	$\frac{9}{2}$	$\frac{48}{5}$	$\frac{73}{12}$	$\frac{2}{2}$	$\frac{5}{2}$	$\frac{55}{6}$	$\frac{90}{16}$	$\frac{2}{2}$	$\frac{4}{2}$	$\frac{54}{6}$	$\frac{91}{17}$
8	$\frac{2}{2}$	$\frac{9}{2}$	$\frac{49}{5}$	$\frac{75}{13}$	$\frac{2}{2}$	$\frac{5}{2}$	$\frac{56}{6}$	$\frac{91}{16}$	$\frac{2}{2}$	$\frac{4}{2}$	$\frac{55}{6}$	$\frac{92}{17}$
	Execution times (seconds)											
1	5	$\frac{10}{7}$	$\frac{20}{7}$	$\frac{21}{10}$	5	$\frac{4}{7}$	$\frac{19}{8}$	$\frac{24}{11}$	3	$\frac{3}{4}$	$\frac{16}{6}$	$\frac{18}{9}$
2	$\frac{82}{7}$	$\frac{110}{10}$	$\frac{172}{10}$	$\frac{136}{13}$	$\frac{72}{7}$	$\frac{55}{9}$	$\frac{111}{9}$	$\frac{84}{12}$	$\frac{43}{4}$	$\frac{26}{5}$	$\frac{63}{7}$	$\frac{44}{10}$
4	$\frac{58}{15}$	$\frac{66}{18}$	$\frac{94}{13}$	$\frac{72}{13}$	$\frac{64}{16}$	$\frac{37}{18}$	$\frac{62}{11}$	$\frac{49}{10}$	$\frac{42}{9}$	$\frac{25}{11}$	$\frac{47}{9}$	$\frac{36}{9}$
8	$\frac{49}{26}$	$\frac{41}{28}$	$\frac{59}{16}$	$\frac{39}{13}$	$\frac{62}{31}$	$\frac{25}{33}$	$\frac{44}{15}$	$\frac{29}{11}$	$\frac{47}{23}$	$\frac{21}{24}$	$\frac{46}{14}$	$\frac{26}{10}$

Table 6.5: Iteration counts and execution times of the two-level domain decomposition algorithm (3.10)–(3.14) using the minimal and maximal overlap sizes for both S and M above and below the line, respectively, with $N = 64$ for the test problem (6.17).

In Section 6.3, we construct the monotone multigrid method and in Theorem 26 prove the monotonic convergence. The numerical experiments are presented in Section 6.3.1. From the numerical experiments, we observe that the block monotone multigrid method is uniformly convergent in ε in its cycle counts. The relaxation parameter $\omega = 1$ is optimal for both of the test problems. For both of the convection-diffusion problem with parabolic boundary layers and the anisotropic convection-diffusion problem, the block successive underrelaxation method results in the fastest execution times than the block monotone multigrid method for all values of ε. For both of the test problems, through the numerical experiments, it is observed that the prolongation parameter is extremely small. This provides an explanation why the block successive underrelaxation method results in the fastest execution times and why there is no advantage in using the block monotone multigrid method.

In Section 6.4, the two-level monotone multigrid algorithm is constructed. Theorem 27 proves monotone convergence of this algorithm. The numerical experiments for the two test problems are presented in Section 6.4.1. From these numerical experiments, we

ε	10^{-2}				10^{-3}				10^{-4}			
$S\backslash M$	1	2	4	8	1	2	4	8	1	2	4	8
	Cycle counts											
1	2	$\frac{5}{2}$	$\frac{25}{3}$	$\frac{38}{7}$	2	$\frac{3}{2}$	$\frac{29}{4}$	$\frac{47}{9}$	2	$\frac{3}{2}$	$\frac{29}{4}$	$\frac{48}{9}$
2	$\frac{10}{2}$	$\frac{12}{2}$	$\frac{28}{3}$	$\frac{41}{7}$	$\frac{10}{2}$	$\frac{9}{2}$	$\frac{30}{4}$	$\frac{48}{9}$	$\frac{10}{2}$	$\frac{8}{2}$	$\frac{30}{4}$	$\frac{49}{9}$
4	$\frac{15}{2}$	$\frac{16}{3}$	$\frac{29}{3}$	$\frac{42}{7}$	$\frac{16}{2}$	$\frac{12}{2}$	$\frac{30}{4}$	$\frac{49}{9}$	$\frac{16}{2}$	$\frac{11}{2}$	$\frac{30}{4}$	$\frac{50}{9}$
8	$\frac{28}{7}$	$\frac{28}{7}$	$\frac{33}{6}$	$\frac{46}{10}$	$\frac{29}{7}$	$\frac{21}{7}$	$\frac{31}{5}$	$\frac{51}{10}$	$\frac{30}{7}$	$\frac{20}{7}$	$\frac{33}{5}$	$\frac{52}{10}$
	Execution times (seconds)											
1	8	$\frac{726}{12}$	$\frac{4187}{733}$	$\frac{3492}{1108}$	9	$\frac{332}{12}$	$\frac{3737}{957}$	$\frac{2881}{974}$	8	$\frac{171}{11}$	$\frac{2901}{802}$	$\frac{2002}{763}$
2	$\frac{102}{11}$	$\frac{1160}{704}$	$\frac{2434}{1223}$	$\frac{2508}{2294}$	$\frac{103}{12}$	$\frac{795}{736}$	$\frac{2308}{1628}$	$\frac{2141}{2855}$	$\frac{100}{12}$	$\frac{719}{735}$	$\frac{1743}{1624}$	$\frac{1403}{2586}$
4	$\frac{68}{18}$	$\frac{907}{510}$	$\frac{1212}{530}$	$\frac{1350}{926}$	$\frac{82}{21}$	$\frac{721}{402}$	$\frac{1261}{776}$	$\frac{1510}{1215}$	$\frac{89}{23}$	$\frac{920}{473}$	$\frac{1301}{944}$	$\frac{1037}{1117}$
8	$\frac{55}{30}$	$\frac{658}{458}$	$\frac{708}{428}$	$\frac{1033}{603}$	$\frac{81}{41}$	$\frac{705}{657}$	$\frac{813}{441}$	$\frac{1259}{716}$	$\frac{119}{59}$	$\frac{1080}{1074}$	$\frac{1015}{667}$	$\frac{1108}{732}$

Table 6.6: Cycle counts and execution times of the two-level monotone multigrid algorithm using the minimal and maximal overlap sizes for both S and M above and below the line, respectively, with $(t_1, t_2) = (1, 1)$ and $N = 64$ for the test problem (6.19).

observe that the two-level monotone multigrid method is parameter uniformly convergent with respect to its cycle counts. For both of the test problems, the two-level domain decomposition algorithm (3.10)–(3.14) results in the fastest execution time. This indicates no advantage in using the two-level monotone multigrid method for these test problems.

ε	10^{-2}				10^{-3}				10^{-4}			
$S\backslash M$	1	2	4	8	1	2	4	8	1	2	4	8
	Cycle counts											
1	2	$\frac{4}{2}$	$\frac{17}{3}$	$\frac{26}{5}$	2	$\frac{3}{2}$	$\frac{20}{3}$	$\frac{33}{6}$	2	$\frac{2}{2}$	$\frac{20}{3}$	$\frac{33}{7}$
2	$\frac{10}{2}$	$\frac{11}{2}$	$\frac{21}{3}$	$\frac{30}{5}$	$\frac{11}{2}$	$\frac{8}{2}$	$\frac{21}{3}$	$\frac{34}{7}$	$\frac{11}{2}$	$\frac{8}{2}$	$\frac{21}{3}$	$\frac{35}{7}$
4	$\frac{16}{2}$	$\frac{15}{3}$	$\frac{22}{3}$	$\frac{32}{6}$	$\frac{16}{2}$	$\frac{11}{2}$	$\frac{22}{3}$	$\frac{35}{7}$	$\frac{16}{2}$	$\frac{11}{3}$	$\frac{22}{3}$	$\frac{35}{7}$
8	$\frac{30}{7}$	$\frac{27}{7}$	$\frac{27}{6}$	$\frac{39}{9}$	$\frac{31}{7}$	$\frac{19}{7}$	$\frac{23}{5}$	$\frac{38}{8}$	$\frac{31}{8}$	$\frac{19}{7}$	$\frac{24}{5}$	$\frac{38}{8}$
	Execution times (seconds)											
1	11	$\frac{727}{14}$	$\frac{5403}{683}$	$\frac{4534}{1249}$	11	$\frac{330}{15}$	$\frac{5044}{1055}$	$\frac{3746}{1278}$	11	$\frac{171}{15}$	$\frac{3669}{1245}$	$\frac{2589}{1180}$
2	$\frac{98}{11}$	$\frac{1047}{703}$	$\frac{3241}{2462}$	$\frac{3171}{3292}$	$\frac{101}{12}$	$\frac{586}{735}$	$\frac{2937}{2442}$	$\frac{2694}{4476}$	$\frac{106}{13}$	$\frac{503}{738}$	$\frac{2281}{2493}$	$\frac{1684}{4053}$
4	$\frac{69}{18}$	$\frac{725}{511}$	$\frac{1607}{1048}$	$\frac{1727}{1561}$	$\frac{84}{21}$	$\frac{517}{400}$	$\frac{1622}{1147}$	$\frac{1854}{1874}$	$\frac{92}{24}$	$\frac{627}{697}$	$\frac{1661}{1391}$	$\frac{1296}{1749}$
8	$\frac{56}{30}$	$\frac{469}{457}$	$\frac{969}{834}$	$\frac{1344}{1059}$	$\frac{81}{41}$	$\frac{457}{657}$	$\frac{1019}{855}$	$\frac{1483}{1137}$	$\frac{121}{61}$	$\frac{713}{1076}$	$\frac{1311}{1241}$	$\frac{1380}{1166}$

Table 6.7: Cycle counts and execution times of the two-level monotone multigrid algorithm using the minimal and maximal overlap sizes for both S and M above and below the line, respectively, with $(t_1, t_2) = (2, 1)$ and $N = 64$ for the test problem (6.19).

ε	10^{-2}				10^{-3}				10^{-4}			
$S\backslash M$	1	2	4	8	1	2	4	8	1	2	4	8
	Cycle counts											
1	2	$\frac{4}{2}$	$\frac{17}{3}$	$\frac{26}{5}$	2	$\frac{3}{2}$	$\frac{20}{3}$	$\frac{33}{6}$	2	$\frac{2}{2}$	$\frac{20}{3}$	$\frac{33}{7}$
2	$\frac{10}{2}$	$\frac{11}{2}$	$\frac{21}{3}$	$\frac{30}{5}$	$\frac{11}{2}$	$\frac{8}{2}$	$\frac{21}{3}$	$\frac{34}{7}$	$\frac{11}{2}$	$\frac{8}{2}$	$\frac{21}{3}$	$\frac{35}{7}$
4	$\frac{16}{2}$	$\frac{15}{3}$	$\frac{22}{3}$	$\frac{32}{6}$	$\frac{16}{2}$	$\frac{11}{2}$	$\frac{22}{3}$	$\frac{35}{7}$	$\frac{16}{2}$	$\frac{11}{3}$	$\frac{22}{3}$	$\frac{35}{7}$
8	$\frac{30}{7}$	$\frac{27}{7}$	$\frac{27}{6}$	$\frac{39}{9}$	$\frac{31}{7}$	$\frac{19}{7}$	$\frac{23}{5}$	$\frac{38}{8}$	$\frac{31}{8}$	$\frac{19}{7}$	$\frac{24}{5}$	$\frac{38}{8}$
	Execution times (seconds)											
1	11	$\frac{567}{15}$	$\frac{2770}{368}$	$\frac{2367}{820}$	11	$\frac{173}{16}$	$\frac{2538}{655}$	$\frac{1930}{671}$	12	$\frac{177}{15}$	$\frac{1944}{481}$	$\frac{1389}{639}$
2	$\frac{111}{12}$	$\frac{863}{717}$	$\frac{1754}{1237}$	$\frac{1725}{1644}$	$\frac{109}{13}$	$\frac{595}{753}$	$\frac{1601}{1229}$	$\frac{1475}{2230}$	$\frac{107}{13}$	$\frac{506}{746}$	$\frac{1249}{1212}$	$\frac{983}{2014}$
4	$\frac{68}{17}$	$\frac{669}{512}$	$\frac{838}{530}$	$\frac{925}{794}$	$\frac{82}{20}$	$\frac{518}{401}$	$\frac{884}{579}$	$\frac{1032}{944}$	$\frac{90}{24}$	$\frac{629}{698}$	$\frac{886}{707}$	$\frac{707}{871}$
8	$\frac{56}{30}$	$\frac{468}{457}$	$\frac{498}{428}$	$\frac{711}{542}$	$\frac{81}{41}$	$\frac{492}{657}$	$\frac{566}{441}$	$\frac{855}{572}$	$\frac{120}{61}$	$\frac{710}{1069}$	$\frac{700}{650}$	$\frac{741}{587}$

Table 6.8: Cycle counts and execution times of the two-level monotone multigrid algorithm using the minimal and maximal overlap sizes for both S and M above and below the line, respectively, with $(t_1, t_2) = (1, 2)$ and $N = 64$ for the test problem (6.19).

ε	10^{-2}				10^{-3}				10^{-4}			
$S\backslash M$	1	2	4	8	1	2	4	8	1	2	4	8
	Cycle counts											
1	2	$\frac{3}{2}$	$\frac{14}{2}$	$\frac{20}{4}$	2	$\frac{2}{2}$	$\frac{16}{3}$	$\frac{25}{5}$	2	$\frac{2}{2}$	$\frac{16}{3}$	$\frac{26}{5}$
2	$\frac{10}{2}$	$\frac{10}{2}$	$\frac{17}{2}$	$\frac{24}{4}$	$\frac{10}{2}$	$\frac{7}{2}$	$\frac{17}{3}$	$\frac{27}{5}$	$\frac{10}{2}$	$\frac{7}{2}$	$\frac{16}{3}$	$\frac{27}{5}$
4	$\frac{15}{2}$	$\frac{14}{2}$	$\frac{18}{3}$	$\frac{27}{5}$	$\frac{15}{2}$	$\frac{10}{2}$	$\frac{17}{3}$	$\frac{28}{5}$	$\frac{15}{2}$	$\frac{10}{2}$	$\frac{17}{3}$	$\frac{28}{6}$
8	$\frac{28}{7}$	$\frac{24}{7}$	$\frac{24}{6}$	$\frac{34}{8}$	$\frac{29}{7}$	$\frac{17}{7}$	$\frac{19}{4}$	$\frac{31}{7}$	$\frac{29}{7}$	$\frac{18}{7}$	$\frac{21}{4}$	$\frac{30}{7}$
	Execution times (seconds)											
1	12	$\frac{552}{15}$	$\frac{4082}{678}$	$\frac{3467}{1026}$	12	$\frac{330}{16}$	$\frac{3761}{733}$	$\frac{2898}{993}$	12	$\frac{171}{16}$	$\frac{2892}{922}$	$\frac{2013}{944}$
2	$\frac{104}{13}$	$\frac{746}{705}$	$\frac{2542}{1646}$	$\frac{2411}{2628}$	$\frac{107}{14}$	$\frac{484}{740}$	$\frac{2305}{2448}$	$\frac{2076}{3191}$	$\frac{106}{14}$	$\frac{393}{740}$	$\frac{1779}{2486}$	$\frac{1308}{2952}$
4	$\frac{72}{19}$	$\frac{559}{351}$	$\frac{1256}{1056}$	$\frac{1334}{1306}$	$\frac{89}{23}$	$\frac{386}{404}$	$\frac{1271}{1156}$	$\frac{1420}{1338}$	$\frac{94}{24}$	$\frac{531}{476}$	$\frac{1290}{1407}$	$\frac{994}{1509}$
8	$\frac{58}{31}$	$\frac{375}{460}$	$\frac{760}{841}$	$\frac{1032}{948}$	$\frac{84}{43}$	$\frac{352}{660}$	$\frac{807}{689}$	$\frac{1131}{994}$	$\frac{123}{62}$	$\frac{531}{1084}$	$\frac{1012}{1008}$	$\frac{1036}{1021}$

Table 6.9: Cycle counts and execution times of the two-level monotone multigrid algorithm using the minimal and maximal overlap sizes for both S and M above and below the line, respectively, with $(t_1, t_2) = (2, 2)$ and $N = 64$ for the test problem (6.19).

ε	10^{-2}				10^{-3}				10^{-4}			
$S\backslash M$	1	2	4	8	1	2	4	8	1	2	4	8
	Cycle counts											
1	2	$\frac{9}{2}$	$\frac{46}{5}$	$\frac{71}{13}$	2	$\frac{5}{2}$	$\frac{53}{6}$	$\frac{88}{16}$	2	$\frac{4}{2}$	$\frac{54}{6}$	$\frac{90}{17}$
2	$\frac{2}{2}$	$\frac{9}{2}$	$\frac{47}{5}$	$\frac{72}{13}$	$\frac{2}{2}$	$\frac{5}{2}$	$\frac{54}{6}$	$\frac{89}{16}$	$\frac{2}{2}$	$\frac{4}{2}$	$\frac{55}{6}$	$\frac{92}{17}$
4	$\frac{2}{2}$	$\frac{9}{2}$	$\frac{48}{5}$	$\frac{73}{12}$	$\frac{2}{2}$	$\frac{5}{2}$	$\frac{55}{6}$	$\frac{90}{16}$	$\frac{2}{2}$	$\frac{4}{2}$	$\frac{56}{6}$	$\frac{93}{17}$
8	$\frac{2}{2}$	$\frac{9}{2}$	$\frac{49}{5}$	$\frac{75}{13}$	$\frac{2}{2}$	$\frac{5}{2}$	$\frac{56}{6}$	$\frac{92}{16}$	$\frac{2}{2}$	$\frac{4}{2}$	$\frac{59}{6}$	$\frac{95}{17}$
	Execution times (seconds)											
1	6	$\frac{12}{8}$	$\frac{21}{8}$	$\frac{21}{11}$	6	$\frac{5}{9}$	$\frac{19}{9}$	$\frac{22}{12}$	7	$\frac{4}{9}$	$\frac{18}{9}$	$\frac{21}{11}$
2	$\frac{95}{9}$	$\frac{115}{11}$	$\frac{172}{11}$	$\frac{137}{13}$	$\frac{94}{10}$	$\frac{65}{12}$	$\frac{123}{10}$	$\frac{94}{12}$	$\frac{91}{9}$	$\frac{56}{12}$	$\frac{100}{10}$	$\frac{70}{12}$
4	$\frac{60}{15}$	$\frac{67}{18}$	$\frac{95}{14}$	$\frac{72}{13}$	$\frac{77}{19}$	$\frac{42}{21}$	$\frac{72}{13}$	$\frac{55}{11}$	$\frac{84}{21}$	$\frac{36}{24}$	$\frac{64}{14}$	$\frac{42}{12}$
8	$\frac{50}{27}$	$\frac{41}{29}$	$\frac{59}{16}$	$\frac{40}{13}$	$\frac{72}{37}$	$\frac{29}{38}$	$\frac{51}{18}$	$\frac{32}{12}$	$\frac{106}{52}$	$\frac{30}{54}$	$\frac{66}{23}$	$\frac{30}{13}$

Table 6.10: Iteration counts and execution times of the two-level monotone domain decomposition algorithm (3.10)–(3.14) using the minimal and maximal overlap sizes for both S and M above and below the line, respectively, with $N = 64$ for the test problem (6.19).

Chapter 7

Conclusions

In Chapter 1, we review singularly perturbed convection-diffusion problems and domain decomposition algorithms. We identify some of the issues with applying contemporary methods to singularly perturbed convection-diffusion problems. The discussion on uniformly convergent numerical methods for the three two dimensional convection-diffusion problems is presented. We construct the nonlinear difference scheme and prove the maximum principle for the linear version of the scheme. The monotone iterative method is constructed, and its monotone convergence is proven. The simple method for finding initial upper or lower solutions without any prior knowledge of the solution is provided.

In Chapter 2, we construct the nonlinear difference scheme for solving the semilinear two-point boundary-value problem with the convective dominated term and discontinuous data. The construction of the nonlinear difference scheme is based on locally exact schemes or on local Green's functions. *A priori* estimates of the exact solution are proved. We prove that the nonlinear difference scheme on arbitrary meshes converges ε-uniformly to the solution of the continuous problem. The monotone iterative method based on the upper and lower solutions is constructed to compute the nonlinear difference scheme. We prove that the upper and lower solutions generated by the monotone iterative method converge monotonically to the exact solution with a linear rate. By choosing special initial functions, we prove parameter uniform convergence of the iterative sequences to the exact solution of the continuous problem. For the test problems, the numerical experiments are implemented, and the experiments confirm the theoretical results.

In Chapter 3, we investigate the one-level monotone domain decomposition algorithm based on the multiplicative Schwarz method. Monotone convergence of this algorithm is

proven. We apply this algorithm to the two test problems: the convection-diffusion problem with parabolic layers and the anisotropic convection-diffusion problem. Through the numerical experiments, we observe that for both of the test problems the one-level monotone domain decomposition algorithm is parameter uniformly convergent in its iteration counts, and the serial acceleration for both of the test problems indicates an advantage in using the one-level monotone domain decomposition algorithm.

We construct the two-level monotone domain decomposition algorithm based on the combination of outer and inner iterates (the additive Schwarz method). We prove monotone convergence of the inner iterates. The numerical stability of the algorithm is also proven. The numerical experiments are implemented for the convection-diffusion problem with parabolic layers and the anisotropic convection-diffusion problem. The anisotropic convection-diffusion problem is solved on serial and parallel computers. From these numerical experiments, we observe that the two-level domain decomposition algorithm is parameter uniformly convergent in its outer iteration counts. It is also observed that the execution times of the two-level monotone domain decomposition algorithm decrease as the overlap size increases. The serial execution times for the two-level monotone domain decomposition algorithm show a considerable acceleration compared to the undecomposed method for certain combinations of vertical strips. It is also observed that for large number of mesh points and/or small values of ε, parallel computing using the two-level monotone domain decomposition algorithm results in a moderate speedup.

In Chapter 4, we investigate the point monotone relaxation and block monotone relaxation iterative methods. Monotone convergence of the point monotone ω-Jacobi and point successive underrelaxation iterative methods is proven. We show that the successive underrelaxation iterative method converges to the solution of the difference scheme faster that the point monotone ω-Jacobi iterative method.

Monotone convergence of the block monotone ω-Jacobi and block monotone successive underrelaxation methods is proven. The convergence of the two block monotone methods are compared, and it is shown that the block successive underrelaxation method is the fastest. We compare the convergence of the point and block monotone ω-Jacobi methods. It is proven that the block monotone ω-Jacobi method is the fastest. Similarly, we compare the point and block monotone successive underrelaxation methods, where the block monotone successive underrelaxation method is the fastest.

From our numerical experiments, we observe that the point and block monotone successive underrelaxation methods are both parameter uniformly convergent with respect to their iteration counts. For both of the methods, it is observed that $\omega = 1$ is the optimal relaxation parameter. When comparing the execution times of the point and block monotone successive underrelaxation methods, it is observed that for the anisotropic convection-diffusion problem, the block monotone successive underrelaxation method is the fastest, whereas for the convection-diffusion problem with parabolic layers, the point monotone successive underrelaxation method converges slightly faster than the block monotone successive underrelaxation method.

In Chapter 5, we investigate the composite monotone domain decomposition algorithms based on the one- and two-level domain decomposition algorithms from Chapter 3 and the Jacobi, Gauss-Seidel, block Jacobi, block Gauss-Seidel methods from Chapter 4. Monotone convergence of the composite monotone Jacobi domain decomposition and composite monotone Gauss-Seidel domain decomposition algorithms is proven. The convergence of these two algorithms is compared, and proven that the composite monotone Gauss-Seidel domain decomposition algorithm is the fastest.

Monotone convergence of the composite monotone block Jacobi domain decomposition and block Gauss-Seidel domain decomposition algorithms is proven. The composite monotone point and block Jacobi domain decomposition algorithms are compared, and it is shown that the composite monotone block Jacobi domain decomposition algorithm converges faster. The composite monotone point and block Gauss-Seidel domain decomposition algorithms are compared and it is shown that the composite monotone block Gauss-Seidel domain decomposition algorithm converges faster. The convergence of the composite monotone Gauss-Seidel and block Gauss-Seidel domain decomposition algorithms is compared and proven that the composite monotone Gauss-Seidel domain decomposition algorithm converges faster.

The two-level composite monotone domain decomposition algorithms are constructed. The parrallisable nature of these algorithms is discussed.

As the theoretical results indicate that the composite monotone block Gauss-Seidel domain decomposition algorithm has the fastest convergence, we apply this algorithm to the two test problems: the convection-diffusion problem with parabolic boundary layers and the anisotropic convection-diffusion problem. For both of the test problems, the compos-

ite monotone block Gauss-Seidel domain decomposition algorithm converges parameter uniformly with respect to its iteration counts. The numerical experiments are compared to the numerical experiments in Chapters 3 and 4. The numerical experiments on the test problems show that for sufficiently large values of ε, the monotone BSUR method from Chapter 4 results in the highest serial acceleration. However, for ε sufficiently small the composite block monotone Gauss-Seidel domain decomposition algorithm results in the highest serial acceleration.

In Chapter 6, we investigate monotone multigrid methods for solving nonlinear singularly perturbed convection-diffusion problems. We construct the monotone multigrid method and prove its monotonic convergence. The numerical experiments indicate that for both of the test problems, the monotone multigrid method converges parameter uniformly with respect to its cycle counts. For the convection-diffusion problem with parabolic boundary layers, the relaxation parameter $\omega = 1$ is optimal. However, $\omega = 1$ is not optimal for the anisotropic convection-diffusion problem. For both of the convection-diffusion problem with parabolic boundary layers and the anisotropic convection-diffusion problem, the successive underrelaxation method results in faster execution times than the monotone multigrid method. For both of the test problems, through the numerical experiments it is observed that the prolongation parameter is extremely small. This is the reason why the successive underrelaxation method results in the fastest execution times and why there is no advantage in using the monotone multigrid method.

We construct the block monotone multigrid method and prove its monotonic convergence. The numerical experiments are presented. From these numerical experiments, we observe that the block monotone multigrid method is parameter uniformly convergent in its cycle counts. The relaxation parameter $\omega = 1$ is optimal for both of the test problems. For both of the convection-diffusion problem with parabolic boundary layers and the anisotropic convection-diffusion problem, the block successive underrelaxation method results in the fastest execution time than the block monotone multigrid method for all values of ε. For both of the test problems, through the numerical experiments, it is observed that the prolongation parameter is extremely small. This provides the explanation why the block successive underrelaxation method results in the fastest execution times and why there is no advantage in using the block monotone multigrid method.

The two-level monotone multigrid algorithm is constructed, and its monotone conver-

gence is proven. The numerical experiments for two test problems are presented. From these numerical experiments, we observe that the two-level monotone multigrid method is parameter uniformly convergent with respect to its cycle counts. For both of the test problems, the two-level domain decomposition algorithm results in the fastest execution times. This indicates no advantage in using the two-level monotone multigrid method for these test problems.

Appendix A

Theorem proofs from Chapter 4

Proof of Theorem 9. We consider only the case of the upper sequence. The case for lower solutions may be proved in a similar way. If $\overline{v}^{(0)}$ is an upper solution, then from (4.3) it follows that

$$\mathcal{L}_{\text{SUR}} z^{(1)}(p) \leq 0, \quad p \in \Omega^h, \quad z^{(1)}(p) = 0, \quad p \in \partial\Omega^h.$$

By the maximum principle in Lemma 7, we conclude

$$z^{(1)}(p) = \overline{v}^{(1)}(p) - \overline{v}^{(0)}(p) \leq 0, \quad p \in \overline{\Omega}^h. \tag{A.1}$$

Using (4.3), we represent $\omega\mathcal{R}(p, v^{(1)})$ in the following form:

$$\begin{aligned}
\omega\mathcal{R}(p, \overline{v}^{(1)}) &= \omega[\mathcal{L}\overline{v}^{(1)}(p) + f(p, \overline{v}^{(1)})] \qquad (A.2)\\
&= \omega[d(p)\overline{v}^{(1)}(p) - \sum_{p' \in \sigma'(p)} e(p,p')\overline{v}^{(1)}(p') + f(p, \overline{v}^{(1)})]\\
&= \omega[\mathcal{R}(p, \overline{v}^{(0)}) + d(p)z^{(1)}(p) - \sum_{p' \in \sigma'(p)} e(p,p')z^{(1)}(p')\\
&\quad -(f(p, \overline{v}^{(0)}) - f(p, \overline{v}^{(1)}))]\\
&= -(d(p) + \omega c^*)z^{(1)}(p) + \omega \sum_{p' \in \sigma_L(p)} e(p,p')z^{(1)}(p') + \omega d(p)z^{(1)}(p)\\
&\quad -\omega \sum_{p' \in \sigma'(p)} e(p,p')z^{(1)}(p') + \omega f_v^{(1)}(p)z^{(1)}(p)\\
&= -(1-\omega)d(p)z^{(1)}(p) - \omega \sum_{p' \in \sigma_U(p)} e(p,p')z^{(1)}(p')\\
&\quad -\omega(c^* - f_v^{(1)}(p))z^{(1)}(p), \quad p \in \Omega^h,
\end{aligned}$$

where the mean value theorem is used in the form

$$f(p, \overline{v}^{(0)}) - f(p, \overline{v}^{(1)}) = f(p, \overline{v}^{(0)}) - f(p, \overline{v}^{(0)} + z^{(1)}) = -f_v^{(1)}(p)z^{(1)},$$

$$f_v^{(1)}(p) = f_v[p, \overline{v}^{(0)}(p) + \theta^{(1)}(p) z^{(1)}(p)], \quad 0 < \theta^{(1)}(p) < 1.$$

Taking into account (1.16), (1.20) and (A.1), we conclude that $\mathcal{R}(p, \overline{v}^{(1)}) \geq 0$, $p \in \Omega^h$. By construction, $\overline{v}^{(1)}(p) = \overline{v}^{(0)}(p) = g(p)$, $p \in \partial\Omega^h$, therefore $\overline{v}^{(1)}(p)$ is an upper solution to the difference scheme (1.15).

By induction on n, we conclude that

$$z^{(n)}(p) \leq 0, \quad p \in \overline{\Omega}^h, \quad n \geq 1. \tag{A.3}$$

Thus, $\{\overline{v}^{(n)}\}$ is a monotonically decreasing sequence of upper solutions. This sequence is bounded below by $\underline{v}$, where $\underline{v}$ is any lower solution (1.19). This means that the function $v(p)$ defined as

$$v(p) = \lim_{n\to\infty} \overline{v}^{(n)}(p), \quad p \in \overline{\Omega}^h, \tag{A.4}$$

exists. We now prove that $v(p)$ is a solution to (1.15). From (4.3), we have

$$\lim_{n\to\infty} z^{(n)}(p) = \lim_{n\to\infty} (\overline{v}^{(n)}(p) - \overline{v}^{(n-1)}(p)) = 0, \quad p \in \overline{\Omega}^h.$$

Similar to (A.2), using the mean value theorem, we can show

$$\begin{aligned} \omega\mathcal{R}(p, \overline{v}^{(n)}) &= -(1-\omega)d(p)z^{(n)} - \omega \sum_{p' \in \sigma_U(p)} e(p, p') z^{(n)} \\ &\quad -\omega(c^* - f_v^{(n)}(p)) z^{(n)}, \quad p \in \Omega^h, \end{aligned}$$

where $f_v^{(n)}(p) = f_v[p, \overline{v}^{(n-1)}(p) + \theta^{(n)}(p) z^{(n)}(p)]$, $0 < \theta^{(n)}(p) < 1$. Since $\lim z^{(n)} = 0$ as $n \to \infty$, we have

$$\lim_{n\to\infty} \omega\mathcal{R}(p, \overline{v}^{(n)}) = 0.$$

From here and (A.4), it follows that $v(p)$ satisfies

$$\mathcal{L}v + f(p, v) = 0, \quad p \in \Omega^h,$$

$$v(p) = \lim_{n\to\infty} \overline{v}^{(n)}(p) = g(p) \quad p \in \partial\Omega^h.$$

Therefore $v(p)$ is a solution of (1.15), and we prove the theorem. □

Proof of Theorem 12. We consider only the case of the upper solutions. The case for the lower solutions may be proved in a similar way.

From the proof of Theorem 11, $A_i + \omega c^* I$ is an M-matrix (4.17). If $\overline{V}^{(0)}$ is an upper solution, then from (4.15) it follows that

$$(A_i + \omega c^* I) Z_i^{(1)} - \omega W_i Z_{i-1}^{(1)} \leq 0. \tag{A.5}$$

From here with $i = 1$ and using the fact that $Z_0^{(1)} = 0$, we get

$$(A_1 + \omega c^* I) Z_1^{(1)} \leq 0.$$

From (4.17) with $i = 1$, we conclude that $Z_1^{(1)} \leq 0$. From here and (A.5) with $i = 2$, we have

$$(A_2 + \omega c^* I) Z_2^{(1)} \leq \omega W_2 Z_1^{(1)} \leq 0.$$

Using (4.17) with $i = 2$, we conclude that $Z_2^{(1)} \leq 0$. By induction on i, we are able to show $Z_i^{(1)} \leq 0, \quad i = 1, \ldots, N_x - 1$, that is,

$$\overline{V}_i^{(1)} \leq \overline{V}_i^{(0)}, \quad i = 1, \ldots, N_x - 1. \tag{A.6}$$

We now need to prove that $\overline{V}^{(1)}$ is an upper solution, that is,

$$\mathcal{R}_i(\overline{V}^{(1)}) \geq 0, \quad i = 1, \ldots, N_x - 1. \tag{A.7}$$

From (4.15), we have

$$\begin{aligned}
\omega \mathcal{R}_i(\overline{V}^{(1)}) &= \omega \mathcal{R}_i(\overline{V}^{(0)} + Z^{(1)}) \\
&= \omega(A_i Z_i^{(1)} - W_i Z_{i-1}^{(1)} - E_i Z_{i+1}^{(1)} + F(\overline{V}_i^{(0)} + Z_i^{(1)}) + G_i^*) \\
&\quad + \omega(A_i \overline{V}_i^{(0)} - W_i \overline{V}_{i-1}^{(0)} - E_i \overline{V}_{i+1}^{(0)}) \\
&= \omega(A_i Z_i^{(1)} - W_i Z_{i-1}^{(1)} - E_i Z_{i+1}^{(1)} + F(\overline{V}_i^{(0)} + Z_i^{(1)}) - F(\overline{V}_i^{(0)})) \\
&\quad + \omega \mathcal{R}_i(\overline{V}^{(0)}) \\
&= \omega(A_i Z_i^{(1)} - W_i Z_{i-1}^{(1)} - E_i Z_{i+1}^{(1)} + F(\overline{V}_i^{(0)} + Z_i^{(1)}) - F(\overline{V}_i^{(0)})) \\
&\quad - (A_i + \omega c^* I) Z_i^{(1)} + \omega W_i Z_{i-1}^{(1)} \\
&= -(1 - \omega) A_i Z_i^{(1)} - \omega c^* Z_i^{(1)} - \omega E_i Z_{i+1}^{(1)} \\
&\quad + \omega(F(\overline{V}_i^{(0)} + Z_i^{(1)}) - F(\overline{V}_i^{(0)})) \\
&= -(1 - \omega) A_i Z_i^{(1)} - \omega(c^* I - F_{i,u}^{(1)}) Z_i^{(1)} - \omega E_i Z_{i+1}^{(1)},
\end{aligned}$$

where we use the mean value theorem

$$F(V_i^{(0)} + Z_i^{(1)}) - F(V_i^{(0)}) = F_{i,u}^{(1)} Z_i^{(1)}. \tag{A.8}$$

$F_{i,u}^{(1)}$ is a diagonal $(N_y - 1) \times (N_y - 1)$ matrix with diagonal entries

$$(F_{i,u}^{(1)})_{jj} = \frac{\partial F_{ij}(\overline{V}_j^{(0)} + \Theta_j^{(0)} Z_j^{(1)})}{\partial u}, \quad j = 1, \ldots, N_y - 1,$$

where $\Theta_j^{(0)}$, $j = 1, \ldots, N_y - 1$, are diagonal matrices with entries lying in $(0, 1)$. From here, (4.12), (4.17), (A.6) and $\omega \in (0, 1]$, we conclude (A.7).

By induction on n, we are able to conclude that $\{\overline{V}^{(n)}\}$ is a monotonically decreasing sequence of upper solutions. By (1.19), this sequence is bounded below by $\underline{V}$, where $\underline{V}$ is any lower solution. Therefore the sequence converges. Let V be defined by

$$V = \lim_{n\to\infty} \overline{V}^{(n)}.$$

We now prove that $V = (V_1, \ldots, V_{N_x-1})$ is the solution to (4.13). From (4.15), we have

$$\lim_{n\to\infty} Z_i^{(n)} = \lim_{n\to\infty} (\overline{V}_i^{(n)} - \overline{V}_i^{(n-1)}) = 0, \quad i = 1, \ldots, N_x - 1.$$

From here and (4.15), it follows that

$$\lim_{n\to\infty} -\omega \mathcal{R}_i(V^{(n-1)}) = \lim_{n\to\infty} ((A_i + \omega c^* I) Z_i^{(n)} - \omega W_i Z_{i-1}^{(n)}) = 0, \quad i = 1, \ldots, N_x - 1.$$

Thus,

$$\mathcal{R}_i(V) = \lim_{n\to\infty} \mathcal{R}_i(V^{(n)}) = 0, \quad i = 1, \ldots, N_x - 1,$$

and hence, $V = (V_1, \ldots, V_{N_x-1})$ is a solution to (4.13). We prove the theorem. □

Proof of Theorem 14. We prove this theorem only for the case of upper solutions. The case for lower solutions may be shown in a similar way. We introduce the following notation:

$$Z_{\text{BJAC},i}^{(n)} = \overline{V}_{\text{BJAC},i}^{(n)} - \overline{V}_{\text{BJAC},i}^{(n-1)}, \quad Z_{\omega\text{JAC},i}^{(n)} = \overline{V}_{\omega\text{JAC},i}^{(n)} - \overline{V}_{\omega\text{JAC},i}^{(n-1)},$$
$$\zeta_i^{(n)} = \overline{V}_{\text{BJAC},i}^{(n)} - \overline{V}_{\omega\text{JAC},i}^{(n)}, \quad n \geq 1.$$

From (4.14) and (4.25), we have

$$\begin{aligned}
(A_i + \omega c^* I)\zeta_i^{(n)} &= (A_i + \omega c^* I)\overline{V}_{\text{BJAC},i}^{(n)} - (A_i + \omega c^* I)\overline{V}_{\omega\text{JAC},i}^{(n)} \\
&= (A_i + \omega c^* I)\zeta_i^{(n-1)} + (A_i + \omega c^* I) Z_{\text{BJAC},i}^{(n)} \\
&\quad -(A_i + \omega c^* I) Z_{\omega\text{JAC},i}^{(n)} \\
&= (A_i + \omega c^* I)\zeta_i^{(n-1)} + (S_i + N_i) Z_{\omega\text{JAC},i}^{(n)} \\
&\quad -\omega(\mathcal{R}_i(\overline{V}_{\text{BJAC}}^{(n-1)}) - \mathcal{R}_i(\overline{V}_{\omega\text{JAC}}^{(n-1)})) \\
&= (A_i + \omega c^* I)\zeta_i^{(n-1)} + (S_i + N_i) Z_{\omega\text{JAC},i}^{(n)} \\
&\quad -\omega(A_i \zeta_i^{(n-1)} - W_i \zeta_{i-1}^{(n-1)} - E_i \zeta_{i+1}^{(n-1)} \\
&\quad +F_i(\overline{V}_{\text{BJAC},i}^{(n-1)}) - F_i(\overline{V}_{\omega\text{JAC},i}^{(n-1)})).
\end{aligned}$$

From here, (4.14) and (4.25),we have

$$\begin{aligned}(A_i+\omega c^*I)\zeta_i^{(n)} &= (1-\omega)A_i\zeta_i^{(n-1)}+\omega W_i\zeta_{i-1}^{(n-1)}+\omega E_i\zeta_{i+1}^{(n-1)}+\omega c^*\zeta_i^{(n-1)} \\ &\quad +(S_i+N_i)Z_{\omega\text{JAC},i}^{(n)}-\omega(F_i(\overline{V}_{\text{BJAC},i}^{(n-1)})-F_i(\overline{V}_{\omega\text{JAC},i}^{(n-1)}))\end{aligned} \tag{A.9}$$

For $n=1$, as $\zeta_i^{(0)}=0$, $i=1,\ldots,N_x-1$, we have

$$(A_i+\omega c^*I)\zeta_i^{(1)} = (S_i+N_i)Z_{\omega\text{JAC},i}^{(1)}.$$

From Theorem 8, $Z_{\omega\text{JAC},i}^{(1)}\leq 0$. From here and (4.12), we have

$$(A_i+\omega c^*I)\zeta_i^{(1)}\leq 0, \quad i=1,\ldots,N_x-1.$$

From here and (4.17), we conclude

$$\zeta_i^{(1)}\leq 0, \quad i=1,\ldots,N_x-1. \tag{A.10}$$

Using (A.9) with $n=2$, we have

$$\begin{aligned}(A_i+\omega c^*I)\zeta_i^{(2)} &= (1-\omega)A_i\zeta_i^{(1)}+\omega W_i\zeta_{i-1}^{(1)}+\omega E_i\zeta_{i+1}^{(1)}+\omega c^*\zeta_i^{(1)} \\ &\quad +(S_i+N_i)Z_{\omega\text{JAC},i}^{(2)}-\omega(F_i(\overline{V}_{\text{BJAC},i}^{(1)})-F_i(\overline{V}_{\omega\text{JAC},i}^{(1)})) \\ &= (1-\omega)A_i\zeta_i^{(1)}+\omega W_i\zeta_{i-1}^{(1)}+\omega E_i\zeta_{i+1}^{(1)}+(S_i+N_i)Z_{\omega\text{JAC},i}^{(2)} \\ &\quad +\omega(c^*I-F_{i,u}^{(1)})\zeta_i^{(1)},\end{aligned}$$

where we use the mean value theorem

$$F(\overline{V}_{\text{BJAC},i}^{(1)})-F(\overline{V}_{\omega\text{JAC},i}^{(1)})=F_{i,u}^{(1)}\zeta_i^{(1)},$$

and $F_{i,u}^{(1)}$ is a $(N_y-1)\times(N_y-1)$ diagonal matrix with entries

$$(F_{i,u}^{(1)})_{jj}=\frac{\partial F_{ij}(\overline{V}_{\omega\text{JAC},j}^{(1)}+\Theta_j^{(1)}\zeta_j^{(1)})}{\partial u}, \quad j=1,\ldots,N_y-1,$$

where $\Theta_j^{(1)}$, $j=1,\ldots,N_y-1$, are diagonal matrices with entries lying in $(0,1)$. From here, (1.20), (4.4), (4.12), (A.10) and $\omega\in(0,1]$, we have

$$(A_i+\omega c^*I)\zeta_i^{(2)}\leq 0.$$

From here and (4.17), we conclude

$$\zeta_i^{(2)}\leq 0, \quad i=1,\ldots,N_x-1.$$

By induction on n, we conclude that

$$\zeta_i^{(n)} = \overline{V}_{\mathrm{BJAC},i}^{(n)} - \overline{V}_{\omega\mathrm{JAC},i}^{(n)} \leq 0, \quad i = 1, \ldots, N_x - 1, \quad n \geq 1,$$

and prove the theorem. □

Proof of Theorem 15. We prove this theorem only for the case of upper solutions. The case of lower solutions may be shown in a similar way. We introduce the following notation:

$$Z_{\mathrm{BSUR},i}^{(n)} = \overline{V}_{\mathrm{BSUR},i}^{(n)} - \overline{V}_{\mathrm{BSUR},i}^{(n-1)}, \quad Z_{\mathrm{SUR},i}^{(n)} = \overline{V}_{\mathrm{SUR},i}^{(n)} - \overline{V}_{\mathrm{SUR},i}^{(n-1)},$$

$$\zeta_i^{(n)} = \overline{V}_{\mathrm{BSUR},i}^{(n)} - \overline{V}_{\mathrm{SUR},i}^{(n)}, \quad n \geq 1.$$

From (4.15), (4.26) and $Z_{\mathrm{BSUR},i}^{(n)} - Z_{\mathrm{SUR},i}^{(n)} = \zeta_i^{(n)} - \zeta_i^{(n-1)}$, $i = 1, \ldots, N_x - 1$, we have

$$\begin{aligned}
(A_i + \omega c^* I)\zeta_i^{(n)} &= (A_i + \omega c^* I)\overline{V}_{\mathrm{BSUR},i}^{(n)} - (A_i + \omega c^* I)\overline{V}_{\mathrm{SUR},i}^{(n)} \qquad \text{(A.11)}\\
&= (A_i + \omega c^* I)\zeta_i^{(n-1)} + (A_i + \omega c^* I)Z_{\mathrm{BSUR},i}^{(n)}\\
&\quad -(A_i + \omega c^* I)Z_{\mathrm{SUR},i}^{(n)}\\
&= (A_i + \omega c^* I)\zeta_i^{(n-1)} + \omega W_i Z_{\mathrm{BSUR},i-1}^{(n)} + (1-\omega)S_i Z_{\mathrm{SUR},i}^{(n)}\\
&\quad +N_i Z_{\mathrm{SUR},i}^{(n)} - \omega W_i Z_{\mathrm{SUR},i-1}^{(n)} - \omega(\mathcal{R}_i(\overline{V}_{\mathrm{BSUR}}^{(n-1)}) - \mathcal{R}_i(\overline{V}_{\mathrm{SUR}}^{(n-1)}))\\
&= (A_i + \omega c^* I)\zeta_i^{(n-1)} + (1-\omega)S_i Z_{\mathrm{SUR},i}^{(n)} + N_i Z_{\mathrm{SUR},i}^{(n)}\\
&\quad +\omega W_i(Z_{\mathrm{BSUR},i-1}^{(n)} - Z_{\mathrm{SUR},i}^{(n)}) - \omega(A_i\zeta_i^{(n-1)} - W_i\zeta_{i-1}^{(n-1)}\\
&\quad -E_i\zeta_{i+1}^{(n-1)} + F(\overline{V}_{\mathrm{BSUR},i}^{(n-1)}) - F(\overline{V}_{\mathrm{SUR},i}^{(n-1)}))\\
&= (A_i + \omega c^* I)\zeta_i^{(n-1)} + (1-\omega)S_i Z_{\mathrm{SUR},i}^{(n)} + N_i Z_{\mathrm{SUR},i}^{(n)}\\
&\quad +\omega W_i\zeta_{i-1}^{(n)} - \omega W_i\zeta_{i-1}^{(n-1)} - \omega(A_i\zeta_i^{(n-1)} - W_i\zeta_{i-1}^{(n-1)}\\
&\quad -E_i\zeta_{i+1}^{(n-1)} + F(\overline{V}_{\mathrm{BSUR},i}^{(n-1)}) - F(\overline{V}_{\mathrm{SUR},i}^{(n-1)})))\\
&= (1-\omega)A_i\zeta_i^{(n-1)} + \omega W_i\zeta_{i-1}^{(n)} + \omega E_i\zeta_{i+1}^{(n-1)}\\
&\quad +(1-\omega)S_i Z_{\mathrm{SUR},i}^{(n)} + N_i Z_{\mathrm{SUR},i}^{(n)} + \omega c^*\zeta_i^{(n-1)}\\
&\quad -\omega(F(\overline{V}_{\mathrm{BSUR},i}^{(n-1)}) - F(\overline{V}_{\mathrm{SUR},i}^{(n-1)})).
\end{aligned}$$

For $n = 1$, as $\zeta_i^{(0)} = 0$, $i = 1, \ldots, N_x - 1$, we have

$$(A_i + \omega c^* I)\zeta_i^{(1)} = \omega W_i\zeta_{i-1}^{(1)} + (1-\omega)S_i Z_{\mathrm{SUR},i}^{(1)} + N_i Z_{\mathrm{SUR},i}^{(1)}. \qquad \text{(A.12)}$$

For $i = 1$, as $\zeta_0^{(1)} = 0$,

$$(A_1 + \omega c^* I)\zeta_1^{(1)} = (1-\omega)S_1 Z_{\mathrm{SUR},1}^{(1)} + N_1 Z_{\mathrm{SUR},1}^{(1)}.$$

From (4.9) with $n = 1$, (4.12) and $\omega \in (0, 1]$ we have

$$(A_1 + \omega c^* I)\zeta_1^{(1)} \leq 0.$$

From here and (4.17), we conclude that

$$\zeta_1^{(1)} \leq 0. \tag{A.13}$$

From (A.12) with $i = 2$, we have

$$(A_2 + \omega c^* I)\zeta_2^{(1)} = \omega W_2 \zeta_1^{(1)} + (1 - \omega) S_2 Z_{\text{SUR},2}^{(1)} + N_2 Z_{\text{SUR},2}^{(1)}.$$

From (4.9), (4.12), (A.13) and $\omega \in (0, 1]$, we have

$$(A_2 + \omega c^* I)\zeta_2^{(1)} \leq 0.$$

From (4.17), we conclude that

$$\zeta_2^{(1)} \leq 0.$$

By induction on i, we conclude

$$\zeta_i^{(1)} \leq 0, \quad i = 1, \ldots, N_x - 1. \tag{A.14}$$

Thus, $\overline{V}_{\text{BSUR},i}^{(1)} \leq \overline{V}_{\text{SUR},i}^{(1)}$, $i = 1, \ldots, N_x - 1$.

Using (A.11) with $n = 2$, we have

$$\begin{aligned}
(A_i + \omega c^* I)\zeta_i^{(2)} &= (1 - \omega) A_i \zeta_i^{(1)} + \omega W_i \zeta_{i-1}^{(2)} + \omega E_i \zeta_{i+1}^{(1)} + N_i Z_{\text{SUR},i}^{(2)} \\
&\quad + (1 - \omega) S_i Z_{\text{SUR},i}^{(2)} + \omega c^* \zeta_i^{(1)} \\
&\quad - \omega (F(\overline{V}_{\text{BSUR},i}^{(1)}) - F(\overline{V}_{\text{SUR},i}^{(1)}))) \\
&= (1 - \omega) A_i \zeta_i^{(1)} + \omega W_i \zeta_{i-1}^{(2)} + \omega E_i \zeta_{i+1}^{(1)} + N_i Z_{\text{SUR},i}^{(2)} \\
&\quad + (1 - \omega) S_i Z_{\text{SUR},i}^{(2)} + \omega (c^* I - F_{i,u}^{(1)}) \zeta_i^{(1)},
\end{aligned}$$

where we use the mean value theorem

$$F(\overline{V}_{\text{BSUR},i}^{(1)}) - F(\overline{V}_{\text{SUR},i}^{(1)}) = F_{i,u}^{(1)} \zeta_i^{(1)},$$

and $F_{i,u}^{(1)}$ is a $(N_y - 1) \times (N_y - 1)$ diagonal matrix with entries

$$(F_{i,u}^{(1)})_{jj} = \frac{\partial F_{ij}(\overline{V}_{\text{SUR},j}^{(1)} + \Theta_j^{(1)} \zeta_j^{(1)})}{\partial u}, \quad j = 1, \ldots, N_y - 1,$$

where $\Theta_j^{(1)}$, $j = 1, \ldots, N_y - 1$, are diagonal matrices with entries lying in $(0, 1)$. Taking into account $\zeta_0^{(2)} = 0$, from here with $i = 1$, we have

$$\begin{aligned}(A_1 + \omega c^* I)\zeta_1^{(2)} &= (1-\omega)A_1\zeta_1^{(1)} + \omega E_1\zeta_2^{(1)} + N_1 Z_{\mathrm{SUR},1}^{(2)} \\ &\quad +(1-\omega)S_1 Z_{\mathrm{SUR},1}^{(2)} + \omega(c^* I - F_{1,u}^{(1)})\zeta_1^{(1)}.\end{aligned}$$

From (1.20), (4.9) with $n = 2$, (4.12), (A.14) and $\omega \in (0, 1]$, we have

$$(A_1 + \omega c^* I)\zeta_1^{(2)} \leq 0.$$

Using (4.17), we conclude that

$$\zeta_1^{(2)} \leq 0. \tag{A.15}$$

For $i = 2$, we have

$$\begin{aligned}(A_2 + \omega c^* I)\zeta_2^{(2)} &= (1-\omega)A_2\zeta_2^{(1)} + \omega W_2\zeta_1^{(2)} + \omega E_2\zeta_3^{(1)} + N_2 Z_{\mathrm{SUR},2}^{(2)} \\ &\quad +(1-\omega)S_2 Z_{\mathrm{SUR},2}^{(2)} + \omega(c^* I - F_{2,u}^{(1)})\zeta_2^{(1)}.\end{aligned}$$

From (1.20), (4.9) with $n = 2$, (4.12), (A.14), (A.15) and $\omega \in (0, 1]$, we have

$$\zeta_2^{(2)} \leq 0.$$

By induction on i, we conclude that

$$\zeta_i^{(2)} \leq 0, \quad i = 1, \ldots, N_x - 1.$$

Now by induction on n, we conclude that

$$\zeta_i^{(n)} = \overline{V}_{\mathrm{BSUR},i}^{(n)} - \overline{V}_{\mathrm{SUR},i}^{(n)} \leq 0, \quad i = 1, \ldots, N_x - 1, \quad n \geq 1,$$

and prove the theorem. □

Appendix B

Theorem proofs from Chapter 5

Proof of Theorem 17. We consider only the case of upper sequences, as the case of lower sequences is proved in a similar manner.

Let $\overline{v}^{(0)}(p)$, $p \in \overline{\Omega}^h$ be an upper solution of (1.15) satisfying the boundary conditions. From (5.1) with $n = 1$, $m = 1$ and using (4.9) from Theorem 9, we have

$$z_1^{(1)}(p) \leq 0, \quad p \in \overline{\Omega}_1^h. \tag{B.1}$$

From here and (5.2) with $n = 1$ and $m = 2$, we have

$$\begin{gathered}(\mathcal{L} + c^*) z_2^{(n)}(p) = -\mathcal{R}(p, v^{(0)}) \leq 0, \quad p \in \Omega_2^h, \\ z_2^{(1)}(\gamma_2^{hl}) = z_1^{(1)}(\gamma_2^{hl}) \leq 0, \quad z_2^{(1)}(\gamma_2^{hr}) = 0, \quad z_2^{(1)}(\gamma_2^{h0}) = 0,\end{gathered}$$

By the maximum principle in Lemma 1, we conclude

$$z_2^{(1)}(p) \leq 0, \quad p \in \overline{\Omega}_2^h,$$

and by induction on m, $z_m^{(1)}(p) \leq 0$, $p \in \overline{\Omega}_m^h$, $m = 2, \ldots, M$. From here and (B.1), it follows that

$$z_m^{(1)}(p) \leq 0, \quad p \in \overline{\Omega}_m^h, \quad m = 1, \ldots, M.$$

We now prove that $v^{(1)}(p)$ defined in Step 4 is an upper solution, that is,

$$\mathcal{R}(p, v^{(1)}) \geq 0, \quad p \in \Omega. \tag{B.2}$$

From (5.2) and the mean value theorem,

$$\begin{aligned}
\mathcal{R}(p, v_m^{(1)}) &= \mathcal{R}(p, \overline{v}^{(0)} + z_m^{(1)}) \\
&= \mathcal{L}\overline{v}^{(0)}(p) + \mathcal{L}z_m^{(1)}(p) + f(p, \overline{v}^{(0)} + z_m^{(1)}) \\
&= \mathcal{L}\overline{v}^{(0)}(p) + f(p, \overline{v}^{(0)}) + \mathcal{L}z_m^{(1)}(p) + f(p, \overline{v}^{(0)} + z_m^{(1)}) - f(p, \overline{v}^{(0)}) \\
&= \mathcal{R}(p, \overline{v}^{(0)}) + \mathcal{L}z_m^{(1)}(p) + f_v^{(1)}(p) z_m^{(1)}(p) \\
&= -(\mathcal{L} + c^*) z_m^{(1)}(p) + \mathcal{L}z_m^{(1)}(p) + f_v^{(1)}(p) z_m^{(1)}(p) \\
&= -(c^* - f_v^{(1)}(p)) z_m^{(1)}(p), \quad p \in \Omega_m^h, \quad m = 1, \ldots, M,
\end{aligned}$$

where $f_v^{(1)}(p) = f_v[p, \overline{v}^{(0)}(p) + \Theta_m^{(1)}(p) z_m^{(1)}(p)]$, $0 < \Theta_m^{(1)}(p) < 1$. From here and (1.20), we have

$$\mathcal{R}(p, v_m^{(1)}) = -(c^* - f_v^{(1)}(p)) z_m^{(1)} \geq 0 \quad p \in \Omega_m^h, \quad m = 1, \ldots, M.$$

From here and (5.3), we have

$$\mathcal{R}(p, v^{(1)}) \geq 0, \quad p \in \Omega^h \setminus \gamma^{hl}, \quad \gamma^{hl} = \bigcup_{m=2}^{M} \gamma_m^{hl}.$$

We therefore only need to verify (B.2) on the boundaries γ_m^{hl}, $m = 2, \ldots, M$. Let

$$\zeta_m^{(n)}(p) = v_m^{(n)}(p) - v_{m+1}^{(n)}(p) = z_m^{(n)}(p) - z_{m+1}^{(n)}, \quad p \in \overline{\theta}_m^h.$$

Using (4.3) with $\omega = 1$ and (5.2) with $m = 2$, we obtain

$$\begin{aligned}
(\mathcal{L} + c^*)\zeta_1^{(1)}(p) &= (\mathcal{L} + c^*) z_1^{(1)}(p) - (\mathcal{L} + c^*) z_2^{(1)}(p) \\
&= (\mathcal{L} + c^*) z_1^{(1)}(p) + \mathcal{R}(p, v^{(0)}) \\
&= (d(p) + c^*) z_1^{(1)}(p) - \sum_{p' \in \sigma'(p)} e(p, p') z_1^{(1)}(p') + \mathcal{R}(p, v^{(0)}) \\
&= (d(p) + c^*) z_1^{(1)}(p) - \sum_{p' \in \sigma_L(p)} e(p, p') z_1^{(1)}(p') \\
&\qquad - \sum_{p' \in \sigma_U(p)} e(p, p') z_1^{(1)}(p') + \mathcal{R}(p, v^{(0)}) \\
&= -\sum_{p' \in \sigma_U(p)} e(p, p') z_1^{(1)}(p'), \quad p \in \theta_1^h.
\end{aligned}$$

From here, (1.16) and (B.1), it follows that

$$(\mathcal{L} + c^*)\zeta_1^{(1)}(p) \geq 0, \quad p \in \theta_1^h.$$

Using (5.2), we have

$$\zeta_1^{(1)}(\gamma_1^{hr}) \geq 0, \quad \zeta_1^{(1)}(\gamma_2^{hl}) = 0, \quad \zeta_1^{(1)}(\gamma_1^{h0} \cap \gamma_2^{h0}) = 0,$$

and by the maximum principle in Lemma 1,

$$\zeta_1^{(1)}(p) \geq 0, \quad p \in \overline{\theta}_1^h. \tag{B.3}$$

For $m = 2, \ldots, M-1$, from (2.19) we have

$$\begin{gathered}(\mathcal{L} + c^*)\zeta_m^{(1)}(p) = 0, \quad p \in \theta_m^h, \\ \zeta_m^{(1)}(\gamma_m^{hr}) \geq 0, \quad \zeta_m^{(1)}(\gamma_{m+1}^{hl}) = 0, \quad \zeta_m^{(1)}(\gamma_1^{h0} \cap \gamma_2^{h0}) = 0,\end{gathered}$$

and by the maximum principle in Lemma 1, conclude $\zeta_m^{(1)} \geq 0$, $p \in \overline{\theta}_m^h$, $m = 2, \ldots, M$. From here and (B.3), it follows that

$$v_m^{(1)}(p) \geq v_{m+1}^{(1)}(p), \quad p \in \overline{\theta}_m^h, \quad m = 1, \ldots, M-1.$$

For $p \in \gamma_{m+1}^{hl}$, $m = 1, \ldots, M-1$, we have

$$\begin{aligned}\mathcal{R}(p, v^{(1)}) &= (d(p) + c^*)v^{(1)}(p) - \sum_{p' \in \sigma'(p)} e(p, p')v^{(1)}(p') \\ &= (d(p) + c^*)v^{(1)}(p) - \sum_{p' \in \sigma_L(p)} e(p, p')v^{(1)}(p') \\ &\quad - \sum_{p' \in \sigma_U(p)} e(p, p')v^{(1)}(p') \\ &= (d(p) + c^*)v_m^{(1)}(p) - \sum_{p' \in \sigma_L(p)} e(p, p')v_m^{(1)}(p') \\ &\quad - \sum_{p' \in \sigma_U(p)} e(p, p')v_{m+1}^{(1)}(p') \\ &\geq (d(p) + c^*)v_m^{(1)}(p) - \sum_{p' \in \sigma_L(p)} e(p, p')v_m^{(1)}(p') \\ &\quad - \sum_{p' \in \sigma_U(p)} e(p, p')v_m^{(1)}(p') \\ &= \mathcal{R}(p, v_m^{(1)}) \geq 0.\end{aligned}$$

Thus, $v^{(1)}$ is an upper solution to (1.15). By induction on n, $\{\overline{v}^{(n)}\}$ is a monotonically decreasing sequence of upper solutions bounded below by $\underline{v}$ (1.19), where $\underline{v}$ is any lower solution. Let $v(p)$ be defined by

$$v(p) = \lim_{n \to \infty} \overline{v}^{(n)}(p), \quad p \in \overline{\Omega}^h.$$

We now prove that $v(p)$ is the solution to (1.15). From (5.1)–(5.3),

$$\lim_{n\to\infty} z_m^{(n)} = \lim_{n\to\infty} (v_m^{(n)} - v_m^{(n-1)}) = 0, \quad m = 1, \ldots, M.$$

From here and (4.3),

$$\begin{aligned} \mathcal{R}(p, v) &= \lim_{n\to\infty} \mathcal{R}(p, v^{(n-1)}) \\ &= -[\lim_{n\to\infty} ((d(p) + c^*) z_1^{(n)}(p) - \sum_{p' \in \sigma_L(p)} e(p, p') z_1^{(n)})(p')] \\ &= 0, \quad p \in \Omega_1^h \setminus \theta_1^h. \end{aligned}$$

Using (5.2) and (5.3), we have

$$\begin{aligned} \mathcal{R}(p, v) &= \lim_{n\to\infty} \mathcal{R}(p, v^{(n)}) = \lim_{n\to\infty} \mathcal{R}(p, v_m^{(n)}) = -\lim_{n\to\infty} (\mathcal{L} + c^*) z_m^{(n)}(p) = 0, \\ & p \in \Omega_m^h \setminus \theta_m^h, \quad m = 2, \ldots, M-1, \quad p \in \Omega_M. \end{aligned}$$

Thus,

$$\mathcal{R}(p, v) = 0, \quad p \in \Omega^h.$$

Since $v(p) = g(p)$, $p \in \partial\Omega^h$, we conclude that $v(p)$ is the solution of (1.15). □

Proof of Theorem 20. We prove this theorem only for the case of upper solutions. The case for lower solutions may be shown in a similar way. We introduce the following notation

$$\zeta_m^{(n)}(p) = \overline{v}_{\text{JAC-DD},m}^{(n)}(p) - \overline{v}_{\text{BJAC-DD},m}^{(n)}(p), \quad p \in \overline{\Omega}_m, \quad m = 1, \ldots, M.$$

From (5.1), (5.12) and Theorem 14 with $m = 1$ and $n = 1$, we have

$$\zeta_1^{(1)}(p) = \overline{v}_{\text{JAC-DD},1}^{(1)}(p) - \overline{v}_{\text{BJAC-DD},1}^{(1)}(p) \geq 0, \quad p \in \overline{\Omega}_1^h. \tag{B.4}$$

From (5.2) and (5.13), for $m = 2$ and $n = 1$ it follows that

$$\begin{aligned} (\mathcal{L} + c^*)\zeta_2^{(1)}(p) &= (\mathcal{L} + c^*)\overline{v}_{\text{JAC-DD},2}^{(1)}(p) - (\mathcal{L} + c^*)\overline{v}_{\text{BJAC-DD},2}^{(1)}(p) \\ &= (\mathcal{L} + c^*) z_{\text{JAC-DD},2}^{(1)}(p) - (\mathcal{L} + c^*)\overline{v}^{(0)}(p) \\ &\quad -(\mathcal{L} + c^*) z_{\text{BJAC-DD},2}^{(1)}(p) + (\mathcal{L} + c^*)\overline{v}^{(0)}(p) \\ &= -\mathcal{R}(p, \overline{v}^{(0)}) + \mathcal{R}(p, \overline{v}^{(0)}) = 0, \quad p \in \Omega_2^h, \end{aligned}$$

Using (B.4), we get

$$\zeta_2^{(1)}(\gamma_2^{hl}) = \overline{v}_{\text{JAC-DD},1}^{(1)}(\gamma_2^{hl}) - \overline{v}_{\text{BJAC-DD},1}^{(1)}(\gamma_2^{hl}) \geq 0, \quad \zeta_2^{(1)}(\gamma_2^{hr} \cup \gamma_2^{h0}) = 0.$$

By the maximum principle in Lemma 1, we conclude

$$\zeta_2^{(1)}(p) \geq 0, \quad p \in \overline{\Omega}_2^h,$$

and, by induction on m, $\zeta_m^{(1)}(p) \geq 0$, $p \in \overline{\Omega}_m^h$, $m = 2, \ldots, M$. From here and (B.4), it follows that

$$\zeta_m^{(1)}(p) = \overline{v}_{\text{JAC-DD},m}^{(1)}(p) - \overline{v}_{\text{BJAC-DD},m}^{(1)}(p) \geq 0, \quad p \in \overline{\Omega}_m^h, \quad m = 1, \ldots, M. \tag{B.5}$$

From here, (5.3) and (5.14), we conclude that

$$\overline{v}_{\text{JAC-DD}}^{(1)}(p) \geq \overline{v}_{\text{BJAC-DD}}^{(1)}(p), \quad p \in \overline{\Omega}^h. \tag{B.6}$$

Using the vector-matrix form as in Section 4.2, denote

$$\overline{V}_i^{(n)} = (\overline{v}_{i,1}^{(n)}, \ldots, \overline{v}_{i,N_y-1}^{(n)}), \quad \mathcal{Z}_i^{(n)} = (\zeta_{i,1}^{(n)}, \ldots, \zeta_{i,N_y-1}^{(n)}).$$

Let $\widehat{i}$ correspond to γ_1^{hr}. Using (4.14) and (4.25), for $m = 1$, $n = 2$ and $i = 1, \ldots, \widehat{i} - 1$, we have

$$\begin{aligned}
(A_i + c^*I)\mathcal{Z}_i^{(2)} &= (A_i + c^*I)\overline{V}_{\text{JAC-DD},i}^{(2)} - (A_i + c^*I)\overline{V}_{\text{BJAC-DD},i}^{(2)} \\
&= (A_i + c^*I)\mathcal{Z}_i^{(1)} + (A_i + c^*I)Z_{\text{JAC-DD},i}^{(2)} \\
&\quad -(A_i + c^*I)Z_{\text{BJAC-DD},i}^{(2)} \\
&= (A_i + c^*I)\mathcal{Z}_i^{(1)} - (S_i + N_i)Z_{\text{JAC-DD},i}^{(2)} \\
&\quad -\mathcal{R}_i(\overline{V}_{\text{JAC-DD}}^{(1)}) + \mathcal{R}_i(\overline{V}_{\text{BJAC-DD}}^{(1)}) \\
&= (A_i + c^*I)\mathcal{Z}_i^{(1)} - (S_i + N_i)Z_{\text{JAC-DD},i}^{(1)} \\
&\quad -A_i\mathcal{Z}_i^{(1)} + W_i\mathcal{Z}_{i-1}^{(1)} + E_i\mathcal{Z}_{i+1}^{(1)} \\
&\quad -F_i(\overline{V}_{\text{JAC-DD},i}^{(1)}) + F_i(\overline{V}_{\text{BJAC-DD},i}^{(1)})) \\
&= W_i\mathcal{Z}_{i-1}^{(1)} + E_i\mathcal{Z}_{i+1}^{(1)} - (S_i + N_i)Z_{\text{JAC-DD},i}^{(2)} \\
&\quad +(c^*I - F_{i,u}^{(1)})\mathcal{Z}_i^{(1)}
\end{aligned}$$

where A_i, E_i and W_i are defined in (4.13) and S_i and N_i are defined in (4.24). Here we use the mean value theorem as follows

$$F(\overline{V}_{\text{BJAC-DD},i}^{(1)}) - F(\overline{V}_{\text{JAC-DD},i}^{(1)}) = -F_{i,u}^{(1)}\mathcal{Z}_i^{(1)},$$

and $F_{i,u}^{(1)}$ is $(N_y - 1) \times (N_y - 1)$ diagonal matrix with entries

$$(F_{i,u}^{(1)})_{jj} = \frac{\partial F_{ij}(\overline{V}_{\text{BJAC-DD},j}^{(1)} + \Theta_j^{(1)}\mathcal{Z}_j^{(1)})}{\partial u}, \quad j = 1, \ldots, N_y - 1,$$

where $\Theta_j^{(1)}$, $j = 1, \ldots, N_y - 1$ are diagonal matrices with entries lying in $(0, 1)$. From here, (1.20), (4.4), (4.12) and (B.5), we have

$$(A_i + c^* I)\mathcal{Z}_i^{(2)} \geq 0.$$

From here and (4.17),

$$\mathcal{Z}_i^{(2)} \geq 0, \quad i = 1, \ldots, \widehat{i} - 1,$$

that is,

$$\zeta^{(2)}(p) \geq 0, \quad p \in \Omega_1^h.$$

Using (B.6), we get

$$\zeta_1^{(2)}(\gamma_1^{hr}) = \overline{v}_{\text{JAC-DD}}^{(1)}(\gamma_1^{hr}) - \overline{v}_{\text{BJAC-DD}}^{(1)}(\gamma_1^{hr}) \geq 0, \quad \zeta_1^{(2)}(\gamma_1^{hl} \cup \gamma_1^{h0}) = 0.$$

By the maximum principle in Lemma 1, we conclude

$$\zeta_1^{(2)}(p) = \overline{v}_{\text{JAC-DD},1}^{(2)}(p) - \overline{v}_{\text{BJAC-DD},1}^{(2)}(p) \geq 0, \quad p \in \overline{\Omega}_1^h. \tag{B.7}$$

From (5.2), (5.13) and by the mean value theorem, for $n = 2$ and $m = 2$, it follows that

$$\begin{aligned}
(\mathcal{L} + c^*)\zeta_2^{(2)}(p) &= (\mathcal{L} + c^*)\overline{v}_{\text{JAC-DD},2}^{(2)}(p) - (\mathcal{L} + c^*)\overline{v}_{\text{BJAC-DD},2}^{(2)}(p) \\
&= (\mathcal{L} + c^*)z_{\text{JAC-DD},2}^{(2)}(p) - (\mathcal{L} + c^*)z_{\text{BJAC-DD},2}^{(2)}(p) \\
&\quad (\mathcal{L} + c^*)\overline{v}_{\text{JAC-DD}}^{(1)}(p) - (\mathcal{L} + c^*)\overline{v}_{\text{BJAC-DD}}^{(1)}(p) \\
&= -\mathcal{R}(p, \overline{v}_{\text{JAC-DD}}^{(1)}) + \mathcal{R}(p, \overline{v}_{\text{BJAC-DD}}^{(1)}) \\
&\quad (\mathcal{L} + c^*)\overline{v}_{\text{JAC-DD}}^{(1)}(p) - (\mathcal{L} + c^*)\overline{v}_{\text{BJAC-DD}}^{(1)}(p) \\
&= -\mathcal{L}\overline{v}_{\text{JAC-DD}}^{(1)}(p) - f(p, \overline{v}_{\text{JAC-DD}}^{(1)}) + \mathcal{L}\overline{v}_{\text{BJAC-DD}}^{(1)}(p) + f(p, \overline{v}_{\text{BJAC-DD}}^{(1)}) \\
&\quad (\mathcal{L} + c^*)\overline{v}_{\text{JAC-DD}}^{(1)}(p) - (\mathcal{L} + c^*)\overline{v}_{\text{BJAC-DD}}^{(1)}(p) \\
&= -f(p, \overline{v}_{\text{JAC-DD}}^{(1)}) + f(p, \overline{v}_{\text{BJAC-DD}}^{(1)}) \\
&\quad + c^*\overline{v}_{\text{JAC-DD}}^{(1)}(p) - c^*\overline{v}_{\text{BJAC-DD}}^{(1)}(p) \\
&= -\zeta_2^{(1)}(p) f_v^{(2)}(p) + c^*\zeta_2^{(1)}(p) \\
&= (c^* - f_v^{(2)}(p))\zeta_2^{(1)}(p), \quad p \in \Omega_2^h,
\end{aligned}$$

where $f_v^{(2)}(p) = f_v(p, \overline{v}_{\text{JAC-DD},2}^{(1)} + \Theta_2^{(2)}\zeta_2^{(2)})$, $0 < \Theta_2^{(2)}(p) < 1$. From here, (1.20), (B.5) and (B.7), we have

$$(\mathcal{L} + c^*)\zeta_2^{(2)}(p) \geq 0,$$

$$\zeta_2^{(2)}(\gamma_2^{hl}) = \overline{v}^{(2)}_{\text{JAC-DD},1}(\gamma_2^{hl}) - \overline{v}^{(2)}_{\text{BJAC-DD},1}(\gamma_2^{hl}) \geq 0, \quad \zeta_2^{(2)}(\gamma_2^{h0}) = 0,$$
$$\zeta_2^{(2)}(\gamma_2^{hr}) = \overline{v}^{(2)}_{\text{JAC-DD}}(\gamma_2^{hr}) - \overline{v}^{(2)}_{\text{BJAC-DD}}(\gamma_2^{hr}) \geq 0.$$

By the maximum principle in Lemma 1, we conclude that

$$\zeta_2^{(2)}(p) \geq 0, \quad p \in \overline{\Omega}_2^h,$$

and by induction on m, $\zeta_m^{(2)}(p) \geq 0$, $p \in \overline{\Omega}_m^h$, $m = 2, \ldots, M$. From here and (B.7), it follows that

$$\zeta_m^{(2)}(p) = \overline{v}^{(2)}_{\text{JAC-DD},m}(p) - \overline{v}^{(2)}_{\text{BJAC-DD},m}(p) \geq 0, \quad p \in \overline{\Omega}_m^h, \quad m = 1, \ldots, M.$$

Using (5.3) and (5.14), we conclude that

$$\overline{v}^{(2)}_{\text{JAC-DD}}(p) \geq \overline{v}^{(2)}_{\text{BJAC-DD}}(p), \quad p \in \overline{\Omega}^h,$$

and by induction on n, we prove the theorem. □

Proof of Theorem 21. We consider only the case of upper sequences, as the case of lower sequences is proved in a similar manner.

Let $\overline{v}^{(0)}(p)$, $p \in \overline{\Omega}^h$ be an upper solution of (1.15) satisfying the boundary condition. From (5.12) with $n = 1$, $m = 1$ and using (4.21) from Theorem 12, we have

$$z_1^{(1)}(p) \leq 0, \quad p \in \overline{\Omega}_1^h. \tag{B.8}$$

From (5.13) with $n = 1$ and $m = 2$, we have

$$(\mathcal{L} + c^*) z_2^{(n)}(p) = -\mathcal{R}(p, \overline{v}^{(0)}) \leq 0, \quad p \in \Omega_2^h,$$
$$z_2^{(1)}(\gamma_2^{hl}) = z_1^{(1)}(\gamma_2^{hl}), \quad z_2^{(1)}(\gamma_2^{hr}) = 0, \quad z_2^{(1)}(\gamma_2^{h0}) = 0.$$

From (B.8), we have $z_2^{(1)}(\gamma_2^{hl}) \leq 0$. By the maximum principle in Lemma 1, we conclude

$$z_2^{(1)}(p) \leq 0, \quad p \in \overline{\Omega}_2^h.$$

By induction on m, we conclude $z_m^{(1)}(p) \leq 0$, $p \in \overline{\Omega}_m^h$, $m = 2, \ldots, M$. From here and (B.8), it follows that

$$z_m^{(1)}(p) \leq 0, \quad p \in \overline{\Omega}_m^h, \quad m = 1, \ldots, M. \tag{B.9}$$

We now prove that $v^{(1)}(p)$ defined in Step 4 is an upper solution, that is,

$$\mathcal{R}(p, v^{(1)}) \geq 0, \quad p \in \Omega. \tag{B.10}$$

From (5.13) and the mean value theorem,

$$\begin{aligned}
\mathcal{R}(p, v_m^{(1)}) &= \mathcal{R}(p, \overline{v}^{(0)} + z_m^{(1)}) \\
&= \mathcal{L}\overline{v}^{(0)}(p) + \mathcal{L}z_m^{(1)}(p) + f(p, \overline{v}^{(0)} + z_m^{(1)})(p) \\
&= \mathcal{L}\overline{v}^{(0)}(p) + f(p, \overline{v}^{(0)}) + \mathcal{L}z_m^{(1)}(p) + f(p, \overline{v}^{(0)} + z_m^{(1)}) - f(p, \overline{v}^{(0)}) \\
&= \mathcal{R}(p, \overline{v}^{(0)}) + \mathcal{L}z_m^{(1)}(p) + f_v^{(1)}(p) z_m^{(1)}(p) \\
&= -(\mathcal{L} + c^*) z_m^{(1)}(p) + \mathcal{L}z_m^{(1)}(p) + f_v^{(1)}(p) z_m^{(1)}(p) \\
&= -(c^* - f_v^{(1)}(p)) z_m^{(1)}(p), \quad p \in \Omega_m^h, \quad m = 1, \ldots, M,
\end{aligned}$$

where $f_v^{(1)}(p) = f_v[p, \overline{v}^{(0)}(p) + \Theta_m^{(1)}(p) z_m^{(1)}(p)]$, $0 < \Theta_m^{(1)}(p) < 1$. From here, (1.20) and (B.9), we have

$$\mathcal{R}(p, v_m^{(1)}) = -(c^* - f_v^{(1)}(p)) z_m^{(1)}(p) \geq 0 \quad p \in \Omega_m^h, \quad m = 1, \ldots, M.$$

From here and (5.14), it follows that

$$\mathcal{R}(p, v^{(1)}) \geq 0, \quad p \in \Omega^h \setminus \gamma^{hl}, \quad \gamma^{hl} = \bigcup_{m=2}^{M} \gamma_m^{hl}. \tag{B.11}$$

We therefore only need to verify (B.10) on the boundaries $\gamma_m^{hl}, m = 2, \ldots, M$. Let

$$\zeta_m^{(n)}(p) = v_m^{(n)}(p) - v_{m+1}^{(n)}(p) = z_m^{(n)}(p) - z_{m+1}^{(n)}, \quad p \in \overline{\theta}_m^h.$$

Using (4.15) with $\omega = 1$ and (5.13) with $m = 2$, we obtain

$$\begin{aligned}
(A_i + c^* I)\zeta_{1,i}^{(1)} &= (A_i + c^* I) Z_{1,i}^{(1)} - (A_i + c^* I) Z_{2,i}^{(1)} \\
&= W_i Z_{1,i-1}^{(1)} - \mathcal{R}_i(V^{(0)}) - W_i Z_{2,i-1}^{(1)} - E_i Z_{2,i+1}^{(1)} + \mathcal{R}_i(V^{(0)}) \\
&= W_i (Z_{1,i-1}^{(1)} - Z_{2,i-1}^{(1)}) - E_i Z_{2,i+1}^{(1)}, \quad i = \widehat{i}_1 + 1, \ldots, i_1 - 1,
\end{aligned} \tag{B.12}$$

$$\zeta_1^{(1)} = Z_1^{(1)} - Z_2^{(1)}, \quad Z_m^{(1)} = \{Z_{m,i}, i = \widehat{i}_1, \ldots, i_1\}, \quad m = 1, 2,$$

where $i = \widehat{i}_1$ and $i = i_1$ correspond to the left γ_2^{hl} and right γ_1^{hr} boundaries of $\overline{\theta}_1^h$, respectively. Since $Z_{1,\widehat{i}_1}^{(1)} = Z_{2,\widehat{i}_1}^{(1)}$, for $i = \widehat{i}_1 + 1$, we have

$$(A_{\widehat{i}_1+1} + c^* I)\zeta_{1,\widehat{i}_1+1}^{(1)} = -E_{\widehat{i}_1+1} Z_{2,\widehat{i}_1+2}^{(1)}.$$

From (4.12) and (B.9), we conclude

$$(A_{\widehat{i}_1+1} + c^* I)\zeta_{1,\widehat{i}_1+1}^{(1)} \geq 0,$$

and using (4.17), it follows that

$$\zeta^{(1)}_{1,\widehat{i}_1+1} = Z^{(1)}_{1,\widehat{i}_1+1} - Z^{(1)}_{2,\widehat{i}_1+1} \geq 0. \tag{B.13}$$

From (B.12) with $i = \widehat{i}_1 + 2$, we get

$$(A_{\widehat{i}_1+2} + c^* I)\zeta^{(1)}_{1,\widehat{i}_1+2} = W_{\widehat{i}_1+2}(Z^{(1)}_{1,\widehat{i}_1+1} - Z^{(1)}_{2,\widehat{i}_1+1}) - E_{\widehat{i}_1+2} Z^{(1)}_{2,\widehat{i}_1+3}.$$

From here, (4.12), (B.9) and (B.13), we have

$$(A_{\widehat{i}_1+2} + c^* I)\zeta^{(1)}_{1,\widehat{i}_1+2} \geq 0,$$

and using (4.17), it follows that

$$\zeta^{(1)}_{1,\widehat{i}_1+2} \geq 0.$$

By induction on i, we get

$$\zeta^{(1)}_{1,i} \geq 0, \quad i = \widehat{i}_1 + 1, \ldots, i_1 - 1.$$

Thus, $\zeta_1(p) \geq 0$, $p \in \theta^h_1$. As $\zeta^{(1)}_1(\gamma^{hl}_2) = 0$, $\zeta^{(1)}_1(\gamma^{hr}_1) = -z^{(1)}_2(\gamma^{hr}_1) \geq 0$, we conclude

$$\zeta_1(p) \geq 0, \quad p \in \overline{\theta}^h_1. \tag{B.14}$$

From (5.13) and (B.9), we have

$$(\mathcal{L} + c^*)\zeta^{(1)}_m(p) = (\mathcal{L} + c^*)z^{(1)}_m(p) - (\mathcal{L} + c^*)z^{(1)}_{m+1}(p) = 0, \quad p \in \theta^h_m, \quad m = 2, \ldots, M.$$

$$\zeta^{(1)}_m(p) = 0, \quad p \in \partial\theta^h_m \setminus \gamma^{hr}_m, \quad \zeta^{(1)}_m(p) = -z^{(1)}_{m+1}(p) \geq 0, \quad p \in \gamma^{hr}_m.$$

From here and by the maximum principle in Lemma 1, we get

$$\zeta^{(1)}_m(p) \geq 0, \quad p \in \overline{\theta}^h_m, \quad m = 2, \ldots, M-1,$$

and using (B.14), it follows that

$$\zeta^{(1)}_m(p) = v^{(1)}_m(p) - v^{(1)}_{m+1}(p) \geq 0, \quad p \in \overline{\theta}^h_m, \quad m = 1, \ldots, M-1. \tag{B.15}$$

Let $\tilde{i}$ correspond to γ^{hl}_m, then

$$\begin{aligned} \mathcal{R}_{\tilde{i}}(V^{(1)}) &= A_{\tilde{i}} V^{(1)}_{\tilde{i}} - W_{\tilde{i}} V^{(1)}_{\tilde{i}-1} - E_{\tilde{i}} V^{(1)}_{\tilde{i}+1} + F_{\tilde{i}}(V^{(1)}_{\tilde{i}}) + G^*_i \\ &= A_{\tilde{i}} V^{(1)}_{m-1,\tilde{i}} - W_{\tilde{i}} V^{(1)}_{m-1,\tilde{i}-1} - E_{\tilde{i}} V^{(1)}_{m,\tilde{i}+1} + F_{\tilde{i}}(V^{(1)}_{m-1,\tilde{i}}) + G^*_i. \end{aligned}$$

From here and using (B.15), we have

$$\begin{aligned}\mathcal{R}_{\tilde{i}}(V^{(1)}) &\geq A_{\tilde{i}}V^{(1)}_{m-1,\tilde{i}} - W_{\tilde{i}}V^{(1)}_{m-1,\tilde{i}-1} - E_{\tilde{i}}V^{(1)}_{m-1,\tilde{i}+1} + F_{\tilde{i}}(V^{(1)}_{m-1,\tilde{i}}) + G^*_i \\ &= \mathcal{R}_i(V^{(1)}_{m-1}) \geq 0, \quad m = 2, \ldots, M.\end{aligned}$$

From here and (B.11), it follows that

$$\mathcal{R}(p, v^{(1)}) \geq 0, \quad p \in \Omega^h,$$

and, hence, $v^{(1)}$ is an upper solution to (1.15). By induction on n, we are able to conclude that $\{\overline{v}^{(n)}\}$ is a monotonically decreasing sequence of upper solutions. By (1.19), this sequence is bounded from below by $\underline{v}$, where $\underline{v}$ is any lower solution. Therefore the sequence converges. Let $v(p)$ be defined by

$$v(p) = \lim_{n\to\infty} \overline{v}^{(n)}(p), \quad p \in \overline{\Omega}^h.$$

We now prove that $v(p)$ is the solution to (1.15). From (5.12)–(5.14), we have

$$\lim_{n\to\infty} z^{(n)}_m = \lim_{n\to\infty} (v^{(n)}_m - v^{(n-1)}_m) = 0, \quad m = 1, \ldots, M,$$

and from here and (4.15),

$$\lim_{n\to\infty} \mathcal{R}_i(V^{(n)}) = -\lim_{n\to\infty} ((A_i + c^*I)Z^{(n+1)}_i - W_i Z^{(n+1)}_{i-1}) = 0, \quad i = 1, \ldots, \widehat{i}_1.$$

Thus,

$$\mathcal{R}(p, v) = \lim_{n\to\infty} \mathcal{R}(p, v^{(n)}) = \lim_{n\to\infty} \mathcal{R}(p, v^{(n)}_1) = 0, \quad p \in \Omega^h_1 \setminus \theta^h_1.$$

Using (5.13) and (5.14), we have

$$\begin{aligned}\mathcal{R}(p, v) &= \lim_{n\to\infty} \mathcal{R}(p, v^{(n)}) = \lim_{n\to\infty} \mathcal{R}(p, v^{(n)}_m) = -\lim_{n\to\infty} (\mathcal{L} + c^*) z^{(n)}_m(p) = 0, \\ &\quad p \in \Omega^h_m \setminus \theta^h_m, \quad m = 2, \ldots, M-1, \quad p \in \Omega^h_M.\end{aligned}$$

Thus,

$$\mathcal{R}(p, v) = 0, \quad p \in \Omega^h.$$

From (5.12)–(5.14), we have $v(p) = g(p)$, $p \in \partial\Omega^h$, and conclude that $v(p)$ is the solution of (1.15). □

Proof of Theorem 22. We prove this theorem only for the case of upper solutions. The case for lower solutions may be shown in a similar way. We introduce the following notation:

$$\zeta_m^{(n)}(p) = \overline{v}_{\text{GS-DD},m}^{(n)}(p) - \overline{v}_{\text{BGS-DD},m}^{(n)}(p), \quad p \in \overline{\Omega}_m, \quad m = 1, \ldots, M.$$

From (5.1), (5.12) and Theorem 15 with $m = 1$ and $n = 1$, we have

$$\zeta_1^{(1)}(p) = \overline{v}_{\text{GS-DD},1}^{(1)}(p) - \overline{v}_{\text{BGS-DD},1}^{(1)}(p) \geq 0, \quad p \in \overline{\Omega}_1^h. \tag{B.16}$$

From (5.2) and (5.13), for $m = 2$ and $n = 1$, it follows that,

$$\begin{aligned}
(\mathcal{L} + c^*)\zeta_2^{(1)}(p) &= (\mathcal{L} + c^*)\overline{v}_{\text{GS-DD},2}^{(1)}(p) - (\mathcal{L} + c^*)\overline{v}_{\text{BGS-DD},2}^{(1)}(p) \\
&= (\mathcal{L} + c^*)z_{\text{GS-DD},2}^{(1)}(p) - (\mathcal{L} + c^*)\overline{v}^{(0)}(p) \\
&\quad -(\mathcal{L} + c^*)z_{\text{BGS-DD},2}^{(1)}(p) + (\mathcal{L} + c^*)\overline{v}^{(0)}(p) \\
&= -\mathcal{R}(p, \overline{v}^{(0)}) + \mathcal{R}(p, \overline{v}^{(0)}) = 0, \quad p \in \Omega_2^h,
\end{aligned}$$

Using (B.16), we get

$$\zeta_2^{(1)}(\gamma_2^{hl}) = \overline{v}_{\text{GS-DD},1}^{(1)}(\gamma_2^{hl}) - \overline{v}_{\text{BGS-DD},1}^{(1)}(\gamma_2^{hl}) \geq 0, \quad \zeta_2^{(1)}(\gamma_2^{hr} \cup \gamma_2^{h0}) = 0.$$

By the maximum principle in Lemma 1, we conclude

$$\zeta_2^{(1)}(p) \geq 0, \quad p \in \overline{\Omega}_2^h,$$

and by induction on m, $\zeta_m^{(1)}(p) \geq 0$, $p \in \overline{\Omega}_m^h$, $m = 2, \ldots, M$. From here and (B.16), it follows that

$$\zeta_m^{(1)}(p) = \overline{v}_{\text{GS-DD},m}^{(1)}(p) - \overline{v}_{\text{BGS-DD},m}^{(1)}(p) \geq 0, \quad p \in \overline{\Omega}_m^h, \quad m = 1, \ldots, M. \tag{B.17}$$

From here, (5.3) and (5.14), we conclude that

$$\overline{v}_{\text{GS-DD}}^{(1)}(p) \geq \overline{v}_{\text{BGS-DD}}^{(1)}(p), \quad p \in \overline{\Omega}^h. \tag{B.18}$$

Using the vector-matrix notation as in Section 4.2, denote

$$\overline{V}_i^{(n)} = (\overline{v}_{i,1}^{(n)}, \ldots, \overline{v}_{i,N_y-1}^{(n)}), \quad \mathcal{Z}_i^{(n)} = (\zeta_{i,1}^{(n)}, \ldots, \zeta_{i,N_y-1}^{(n)}).$$

Let $\widehat{i}$ correspond to γ_1^{hr}. For $m = 1$, $n = 2$ using (4.15) and (4.26), for $i = 1, \ldots, \widehat{i} - 1$, we have

$$\begin{aligned}
(A_i + c^*I)\mathcal{Z}_i^{(2)} &= (A_i + c^*I)\overline{V}^{(2)}_{\text{GS-DD},i} - (A_i + c^*I)\overline{V}^{(2)}_{\text{BGS-DD},i} \qquad \text{(B.19)}\\
&= (A_i + c^*I)\mathcal{Z}_i^{(1)} + (A_i + c^*I)Z^{(2)}_{\text{GS-DD},i}\\
&\quad -(A_i + c^*I)Z^{(2)}_{\text{BGS-DD},i}\\
&= (A_i + c^*I)\mathcal{Z}_i^{(1)} + W_i Z^{(2)}_{\text{GS-DD},i-1} - N_i Z^{(2)}_{\text{GS-DD},i}\\
&\quad -\mathcal{R}_i(\overline{V}^{(1)}_{\text{GS-DD}}) - W_i Z^{(2)}_{\text{BGS-DD},i-1} + \mathcal{R}_i(\overline{V}^{(1)}_{\text{BGS-DD}})\\
&= (A_i + c^*I)\mathcal{Z}_i^{(1)} + W_i \mathcal{Z}^{(2)}_{i-1} - N_i Z^{(1)}_{\text{GS-DD},i}\\
&\quad -A_i\mathcal{Z}_i^{(1)} + W_i\mathcal{Z}_{i-1}^{(1)} + E_i\mathcal{Z}_{i+1}^{(1)}\\
&\quad -F_i(\overline{V}^{(1)}_{\text{GS-DD},i}) + F_i(\overline{V}^{(1)}_{\text{BGS-DD},i}))\\
&= W_i\mathcal{Z}^{(2)}_{i-1} + W_i\mathcal{Z}_{i-1}^{(1)} + E_i\mathcal{Z}_{i+1}^{(1)} - N_i Z^{(2)}_{\text{GS-DD},i}\\
&\quad +(c^*I - F^{(1)}_{i,u})\mathcal{Z}_i^{(1)}.
\end{aligned}$$

Here we use the mean value theorem

$$F(\overline{V}^{(1)}_{\text{BGS-DD},i}) - F(\overline{V}^{(1)}_{\text{GS-DD},i}) = -F^{(1)}_{i,u}\mathcal{Z}_i^{(1)},$$

and $F^{(1)}_{i,u}$ is $(N_y - 1) \times (N_y - 1)$ diagonal matrix with entries

$$(F^{(1)}_{i,u})_{jj} = \frac{\partial F_{ij}(\overline{V}^{(1)}_{\text{BGS-DD},j} + \Theta^{(1)}_j\mathcal{Z}_j^{(1)})}{\partial u}, \quad j = 1, \ldots, N_y - 1,$$

where $\Theta^{(1)}_j$, $j = 1, \ldots, N_y - 1$ are diagonal matrices with entries lying in $(0, 1)$. From (1.20), (4.9), (4.12), (B.17) and (B.19), for $i = 1$ as $\mathcal{Z}_0^{(2)} = 0$, we have

$$(A_1 + c^*I)\mathcal{Z}_1^{(2)} \geq 0.$$

From here and (4.17), we conclude that

$$\mathcal{Z}_1^{(2)} \geq 0.$$

From here, (1.20), (4.9), (4.12), (B.17) and (B.19), for $i = 2$ we have

$$(A_2 + c^*I)\mathcal{Z}_2^{(2)} \geq 0.$$

From (4.17), we conclude that

$$\mathcal{Z}_2^{(2)} \geq 0.$$

By induction on i, we conclude

$$\mathcal{Z}_i^{(2)} \geq 0, \quad i = 1, \ldots, \hat{i} - 1,$$

that is,

$$\zeta^{(2)}(p) \geq 0, \quad p \in \Omega_1^h.$$

Using (B.18), we get

$$\zeta_1^{(2)}(\gamma_1^{hr}) = \overline{v}_{\text{GS-DD}}^{(1)}(\gamma_1^{hr}) - \overline{v}_{\text{BGS-DD}}^{(1)}(\gamma_1^{hr}) \geq 0, \quad \zeta_1^{(2)}(\gamma_1^{hl} \cup \gamma_1^{h0}) = 0.$$

By the maximum principle in Lemma 1, we conclude

$$\zeta_1^{(2)}(p) = \overline{v}_{\text{GS-DD},1}^{(2)}(p) - \overline{v}_{\text{BGS-DD},1}^{(2)}(p) \geq 0, \quad p \in \overline{\Omega}_1^h. \tag{B.20}$$

From (5.2), (5.13) and by the mean value theorem, for $n = 2$ and $m = 2$, it follows that

$$\begin{aligned}
(\mathcal{L} + c^*)\zeta_2^{(2)}(p) &= (\mathcal{L} + c^*)\overline{v}_{\text{GS-DD},2}^{(2)}(p) - (\mathcal{L} + c^*)\overline{v}_{\text{BGS-DD},2}^{(2)}(p) \\
&= (\mathcal{L} + c^*)z_{\text{GS-DD},2}^{(2)}(p) - (\mathcal{L} + c^*)z_{\text{BGS-DD},2}^{(2)}(p) \\
&\quad (\mathcal{L} + c^*)\overline{v}_{\text{GS-DD}}^{(1)}(p) - (\mathcal{L} + c^*)\overline{v}_{\text{BGS-DD}}^{(1)}(p) \\
&= -\mathcal{R}(p, \overline{v}_{\text{GS-DD}}^{(1)}) + \mathcal{R}(p, \overline{v}_{\text{BGS-DD}}^{(1)}) \\
&\quad (\mathcal{L} + c^*)\overline{v}_{\text{GS-DD}}^{(1)}(p) - (\mathcal{L} + c^*)\overline{v}_{\text{BGS-DD}}^{(1)}(p) \\
&= -\mathcal{L}\overline{v}_{\text{GS-DD}}^{(1)}(p) - f(p, \overline{v}_{\text{GS-DD}}^{(1)}) + \mathcal{L}\overline{v}_{\text{BGS-DD}}^{(1)}(p) + f(p, \overline{v}_{\text{BGS-DD}}^{(1)}) \\
&\quad (\mathcal{L} + c^*)\overline{v}_{\text{GS-DD}}^{(1)}(p) - (\mathcal{L} + c^*)\overline{v}_{\text{BGS-DD}}^{(1)}(p) \\
&= -f(p, \overline{v}_{\text{GS-DD}}^{(1)}) + f(p, \overline{v}_{\text{BGS-DD}}^{(1)}) \\
&\quad + c^*\overline{v}_{\text{GS-DD}}^{(1)}(p) - c^*\overline{v}_{\text{BGS-DD}}^{(1)}(p) \\
&= -\zeta_2^{(1)}(p) f_v^{(2)}(p) + c^*\zeta_2^{(1)}(p) \\
&= (c^* - f_v^{(2)}(p))\zeta_2^{(1)}(p), \quad p \in \Omega_2^h,
\end{aligned}$$

where $f_v^{(2)}(p) = f_v(p, \overline{v}_{\text{GS-DD},2}^{(1)} + \Theta_2^{(2)}\zeta_2^{(2)})$, $0 < \Theta_2^{(2)}(p) < 1$. From here, (1.20), (B.17) and (B.20), we have

$$(\mathcal{L} + c^*)\zeta_2^{(2)}(p) \geq 0,$$

$$\zeta_2^{(2)}(\gamma_2^{hl}) = \overline{v}_{\text{GS-DD},1}^{(2)}(\gamma_2^{hl}) - \overline{v}_{\text{BGS-DD},1}^{(2)}(\gamma_2^{hl}) \geq 0, \quad \zeta_2^{(2)}(\gamma_2^{h0}) = 0,$$

$$\zeta_2^{(2)}(\gamma_2^{hr}) = \overline{v}_{\text{GS-DD}}^{(2)}(\gamma_2^{hr}) - \overline{v}_{\text{BGS-DD}}^{(2)}(\gamma_2^{hr}) \geq 0.$$

By the maximum principle in Lemma 1, we conclude that

$$\zeta_2^{(2)}(p) \geq 0, \quad p \in \overline{\Omega}_2^h,$$

and by induction on m, $\zeta_m^{(2)}(p) \geq 0$, $p \in \overline{\Omega}_m^h$, $m = 2, \ldots, M$. From here and (B.20), it follows that

$$\zeta_m^{(2)}(p) = \overline{v}_{\text{GS-DD},m}^{(2)}(p) - \overline{v}_{\text{BGS-DD},m}^{(2)}(p) \geq 0, \quad p \in \overline{\Omega}_m^h, \quad m = 1, \ldots, M.$$

Using (5.3) and (5.14), we conclude

$$\overline{v}_{\text{GS-DD}}^{(2)}(p) \geq \overline{v}_{\text{BGS-DD}}^{(2)}(p), \quad p \in \overline{\Omega}^h,$$

and by induction on n, we prove the theorem. □

Proof of Theorem 23. We prove this theorem only for the case of upper solutions. The case for lower solutions may be shown in a similar way. We introduce the following notation:

$$\zeta_m^{(n)}(p) = \overline{v}_{\text{BJAC-DD},m}^{(n)}(p) - \overline{v}_{\text{BGS-DD},m}^{(n)}(p), \quad p \in \overline{\Omega}_m, \quad m = 1, \ldots, M.$$

From (5.12) and Theorem 13 with $m = 1$ and $n = 1$, we have

$$\zeta_1^{(1)}(p) = \overline{v}_{\text{GS-DD},1}^{(1)}(p) - \overline{v}_{\text{BGS-DD},1}^{(1)}(p) \geq 0, \quad p \in \overline{\Omega}_1^h. \tag{B.21}$$

From (5.13), for $m = 2$ and $n = 1$, it follows that,

$$\begin{aligned}
(\mathcal{L} + c^*)\zeta_2^{(1)}(p) &= (\mathcal{L} + c^*)\overline{v}_{\text{BJAC-DD},2}^{(1)}(p) - (\mathcal{L} + c^*)\overline{v}_{\text{BGS-DD},2}^{(1)}(p) \\
&= (\mathcal{L} + c^*)z_{\text{BJAC-DD},2}^{(1)}(p) - (\mathcal{L} + c^*)\overline{v}^{(0)}(p) \\
&\quad -(\mathcal{L} + c^*)z_{\text{BGS-DD},2}^{(1)}(p) + (\mathcal{L} + c^*)\overline{v}^{(0)}(p) \\
&= -\mathcal{R}(p, \overline{v}^{(0)}) + \mathcal{R}(p, \overline{v}^{(0)}) = 0, \quad p \in \Omega_2^h,
\end{aligned}$$

Using (B.21), we get

$$\zeta_2^{(1)}(\gamma_2^{hl}) = \overline{v}_{\text{BJAC-DD},1}^{(1)}(\gamma_2^{hl}) - \overline{v}_{\text{BGS-DD},1}^{(1)}(\gamma_2^{hl}) \geq 0, \quad \zeta_2^{(1)}(\gamma_2^{hr} \cup \gamma_2^{h0}) = 0.$$

By the maximum principle in Lemma 1, we conclude

$$\zeta_2^{(1)}(p) \geq 0, \quad p \in \overline{\Omega}_2^h,$$

and by induction on m, $\zeta_m^{(1)}(p) \geq 0$, $p \in \overline{\Omega}_m^h$, $m = 2, \ldots, M$. From here and (B.21), it follows that

$$\zeta_m^{(1)}(p) = \overline{v}_{\text{BJAC-DD},m}^{(1)}(p) - \overline{v}_{\text{BGS-DD},m}^{(1)}(p) \geq 0, \quad p \in \overline{\Omega}_m^h, \quad m = 1, \ldots, M. \tag{B.22}$$

From here and (5.14), we conclude that

$$\overline{v}_{\text{BJAC-DD}}^{(1)}(p) \geq \overline{v}_{\text{BGS-DD}}^{(1)}(p), \quad p \in \overline{\Omega}^h. \tag{B.23}$$

Using the vector-matrix notation as in Section 4.2, denote

$$\overline{V}_i^{(n)} = (\overline{v}_{i,1}^{(n)}, \ldots, \overline{v}_{i,N_y-1}^{(n)}), \quad \mathcal{Z}_i^{(n)} = (\zeta_{i,1}^{(n)}, \ldots, \zeta_{i,N_y-1}^{(n)}).$$

Let $\widehat{i}$ correspond to γ_1^{hr}. For $m = 1$, $n = 2$ using (4.14) and (4.15), for $i = 1, \ldots, \widehat{i} - 1$, we have

$$\begin{aligned}
(A_i + c^*I)\mathcal{Z}_i^{(2)} &= (A_i + c^*I)\overline{V}_{\text{BJAC-DD},i}^{(2)} - (A_i + c^*I)\overline{V}_{\text{BGS-DD},i}^{(2)} \\
&= (A_i + c^*I)\mathcal{Z}_i^{(1)} + (A_i + c^*I)Z_{\text{BJAC-DD},i}^{(2)} \\
&\quad -(A_i + c^*I)Z_{\text{BGS-DD},i}^{(2)} \\
&= (A_i + c^*I)\mathcal{Z}_i^{(1)} - \mathcal{R}_i(\overline{V}_{\text{BJAC-DD}}^{(1)}) - W_i Z_{\text{BGS-DD},i-1}^{(2)} \\
&\quad +\mathcal{R}_i(\overline{V}_{\text{BGS-DD}}^{(1)}) \\
&= (A_i + c^*I)\mathcal{Z}_i^{(1)} - W_i Z_{\text{BGS-DD},i-1}^{(2)} - A_i\mathcal{Z}_i^{(1)} + W_i\mathcal{Z}_{i-1}^{(1)} \\
&\quad +E_i\mathcal{Z}_{i+1}^{(1)} - F_i(\overline{V}_{\text{BJAC-DD}}^{(1)}) + F_i(\overline{V}_{\text{BGS-DD}}^{(1)}) \\
&= -W_i Z_{\text{BGS-DD},i-1}^{(2)} + W_i\mathcal{Z}_{i-1}^{(1)} + E_i\mathcal{Z}_{i+1}^{(1)} \\
&\quad +(c^*I - F_{i,u}^{(1)})\mathcal{Z}_i^{(1)}.
\end{aligned} \tag{B.24}$$

Here we use the mean value theorem

$$F(\overline{V}_{\text{BGS-DD},i}^{(1)}) - F(\overline{V}_{\text{BJAC-DD},i}^{(1)}) = -F_{i,u}^{(1)}\mathcal{Z}_i^{(1)},$$

and $F_{i,u}^{(1)}$ is $(N_y - 1) \times (N_y - 1)$ diagonal matrix with entries

$$(F_{i,u}^{(1)})_{jj} = \frac{\partial F_{ij}(\overline{V}_{\text{BGS-DD},j}^{(1)} + \Theta_j^{(1)}\mathcal{Z}_j^{(1)})}{\partial u}, \quad j = 1, \ldots, N_y - 1,$$

where $\Theta_j^{(1)}$, $j = 1, \ldots, N_y - 1$ are diagonal matrices with entries lying in $(0, 1)$. From (1.20), (4.12), (4.21), (B.22) and (B.24), we have

$$(A_i + c^*I)\mathcal{Z}_i^{(2)} \geq 0.$$

From here and (4.17), we conclude that

$$\mathcal{Z}_i^{(2)} \geq 0, \quad i = 1, \ldots, \widehat{i} - 1,$$

that is,

$$\zeta^{(2)}(p) \geq 0, \quad p \in \Omega_1^h.$$

Using (B.23), we get

$$\zeta_1^{(2)}(\gamma_1^{hr}) = \overline{v}_{\text{BJAC-DD}}^{(1)}(\gamma_1^{hr}) - \overline{v}_{\text{BGS-DD}}^{(1)}(\gamma_1^{hr}) \geq 0, \quad \zeta_1^{(2)}(\gamma_1^{hl} \cup \gamma_1^{h0}) = 0.$$

By the maximum principle in Lemma 1, we conclude

$$\zeta_1^{(2)}(p) = \overline{v}_{\text{BJAC-DD},1}^{(2)}(p) - \overline{v}_{\text{BGS-DD},1}^{(2)}(p) \geq 0, \quad p \in \overline{\Omega}_1^h. \tag{B.25}$$

From (5.13) and by the mean value theorem, for $n = 2$ and $m = 2$, it follows that

$$\begin{aligned}
(\mathcal{L} + c^*)\zeta_2^{(2)}(p) &= (\mathcal{L} + c^*)\overline{v}_{\text{BJAC-DD},2}^{(2)}(p) - (\mathcal{L} + c^*)\overline{v}_{\text{BGS-DD},2}^{(2)}(p) \\
&= (\mathcal{L} + c^*)z_{\text{BJAC-DD},2}^{(2)}(p) - (\mathcal{L} + c^*)z_{\text{BGS-DD},2}^{(2)}(p) \\
&\quad (\mathcal{L} + c^*)\overline{v}_{\text{BJAC-DD}}^{(1)}(p) - (\mathcal{L} + c^*)\overline{v}_{\text{BGS-DD}}^{(1)}(p) \\
&= -\mathcal{R}(p, \overline{v}_{\text{BJAC-DD}}^{(1)}) + \mathcal{R}(p, \overline{v}_{\text{BGS-DD}}^{(1)}) \\
&\quad (\mathcal{L} + c^*)\overline{v}_{\text{BJAC-DD}}^{(1)}(p) - (\mathcal{L} + c^*)\overline{v}_{\text{BGS-DD}}^{(1)}(p) \\
&= -\mathcal{L}\overline{v}_{\text{BJAC-DD}}^{(1)}(p) - f(p, \overline{v}_{\text{BJAC-DD}}^{(1)}) + \mathcal{L}\overline{v}_{\text{BGS-DD}}^{(1)}(p) \\
&\quad + f(p, \overline{v}_{\text{BGS-DD}}^{(1)}) + (\mathcal{L} + c^*)\overline{v}_{\text{BJAC-DD}}^{(1)}(p) - (\mathcal{L} + c^*)\overline{v}_{\text{BGS-DD}}^{(1)}(p) \\
&= -f(p, \overline{v}_{\text{BJAC-DD}}^{(1)}) + f(p, \overline{v}_{\text{BGS-DD}}^{(1)}) \\
&\quad + c^*\overline{v}_{\text{BJAC-DD}}^{(1)}(p) - c^*\overline{v}_{\text{BGS-DD}}^{(1)}(p) \\
&= -\zeta_2^{(1)}(p) f_v^{(2)}(p) + c^*\zeta_2^{(1)}(p) \\
&= (c^* - f_v^{(2)}(p))\zeta_2^{(1)}(p), \quad p \in \Omega_2^h,
\end{aligned}$$

where $f_v^{(2)}(p) = f_v(p, \overline{v}_{\text{BJAC-DD},2}^{(1)} + \Theta_2^{(2)}\zeta_2^{(2)})$, $0 < \Theta_2^{(2)}(p) < 1$. From here, (1.20), (B.22) and (B.25), we have

$$(\mathcal{L} + c^*)\zeta_2^{(2)}(p) \geq 0,$$

$$\zeta_2^{(2)}(\gamma_2^{hl}) = \overline{v}_{\text{BJAC-DD},1}^{(2)}(\gamma_2^{hl}) - \overline{v}_{\text{BGS-DD},1}^{(2)}(\gamma_2^{hl}) \geq 0, \quad \zeta_2^{(2)}(\gamma_2^{h0}) = 0,$$

$$\zeta_2^{(2)}(\gamma_2^{hr}) = \overline{v}_{\text{BJAC-DD}}^{(2)}(\gamma_2^{hr}) - \overline{v}_{\text{BGS-DD}}^{(2)}(\gamma_2^{hr}) \geq 0.$$

By the maximum principle in Lemma 1, we conclude that

$$\zeta_2^{(2)}(p) \geq 0, \quad p \in \overline{\Omega}_2^h,$$

and by induction on m, $\zeta_m^{(2)}(p) \geq 0$, $p \in \overline{\Omega}_m^h$, $m = 2, \ldots, M$. From here and (B.25), it follows that

$$\zeta_m^{(2)}(p) = \overline{v}_{\text{BJAC-DD},m}^{(2)}(p) - \overline{v}_{\text{BGS-DD},m}^{(2)}(p) \geq 0, \quad p \in \overline{\Omega}_m^h, \quad m = 1, \ldots, M.$$

Using (5.14), we conclude

$$\overline{v}_{\text{BJAC-DD}}^{(2)}(p) \geq \overline{v}_{\text{BGS-DD}}^{(2)}(p), \quad p \in \overline{\Omega}^h,$$

and by induction on n, we prove the theorem. □

Bibliography

[1] J.A.D. Ackroyd, B.P. Axcell and A.J. Ruban, Early Developments of Modern Aerodynamics, Butterworth-Heinemann, Oxford, 2001.

[2] N.S. Bakhvalov, On the optimisation of methods for boundary-value problems with boundary layers, USSR Comput. Maths. Math. Phys., 9, 841–859, (1969) (in Russian).

[3] C. Bäsler and W. Törnig, On monotone including nonlinear multigrid methods and applications, Computing, 50, 51–67, (1993).

[4] M.G. Beckett, The robust and efficient numerical solution of a singularly perturbed boundary value problems using grid adaptivity, PhD Thesis, University of Strathclyde, 1998.

[5] M.G. Beckett and J.A. Mackenzie, Convergence analysis of finite differences approximations on equidistributed grids to a singularly perturbed boundary value problem, Apppl. Numer. Math., 35, 87–109, (2000).

[6] M.G. Beckett and J.A. Mackenzie, On a uniformly accurate finite difference approximation of a singularly perturbed reaction-diffusion problem using grid equidistribution, J. Comput. Appl. Math., 131, 381–405, (2001).

[7] A.E. Berger, J.M. Solomon, M. Ciment, S.H. Leventhal and B.C. Weinberg, Generalized operator compact implicit schemes for boundary layer problems, Math. Comp., 35, 695–731, (1980).

[8] A.E. Berger, J.M. Solomon and M. Ciment, An analysis of a uniformly accurate difference method for a singular perturbation problem, Math. Comp., 37, 79–94, (1981).

[9] I.P. Boglaev, Numerical solution of a singularly perturbed boundary value problem by using a nonuniform mesh, Numerical Methods in Continuum Mechanics, 15, 48–58, (1984).

[10] I.P. Boglaev, Approximte solution of a nonlinear boundary value problem with a small parameter for the highest-order derivative, USSR Comput. Maths. Math. Phys., 24, 30–35, (1984).

[11] I.P. Boglaev, On numerical solution of some singular perturbation problems for quasi-linear equations of elliptic parabolic types, USSR Acad. Sci., Institute of Problems of Microelectronics Technology and Superpure Materials, Preprint, Chernogolovka, 1987 (in Russian).

[12] I.P. Boglaev, A numerical method for a quasi-linear singular perturbation problem of elliptic type, USSR Comput. Maths. Math. Phys., 28, 492–502, (1988).

[13] I.P. Boglaev and V.V. Sirotkin, On numerical solution of some quasi-linear singularly perturbed heat equations using nonuniform grids, USSR Comput. Maths. Math. Phys., 30, 680–696, (1990).

[14] I.P. Boglaev and V.V. Sirotkin, Application of domain decomposition to semiconductor device modeling and simulation in diagnostic of semiconductor structures. In: Computational Methods for Boundary and Interior Layers in Several Dimensions, (J.J.H. Miller, Ed.), Boole Press, Dublin, 1–32, 1993.

[15] I.P. Boglaev, The solution of a singualry perturbed convection-diffusion problem by an iterative domain decomposition method, Numer. Algor., 31, 27–46, (2002).

[16] I.P. Boglaev and V. Duoba, Domain decomposition for an advection-diffusion problem with parabolic layers, Appl. Math. Comput., 146, 27–53, (2003).

[17] I.P. Boglaev, A monotone Schwarz algorithm for a semilinear convection-diffusion problem, J. Numer. Math., 12, 169–191, (2004).

[18] I.P. Boglaev, On monotone iterative methods for a nonlinear singularly perturbed reaction-diffusion problem, J. Comput. Appl. Math., 162, 445–466, (2004).

[19] I.P. Boglaev, Uniform numerical methods on arbitrary meshes for singularly perturbed problems with discontinuous data, Appl. Math. Comput., 154, 815–833, (2004).

[20] I.P. Boglaev and V. Duoba, On an uniform multidomain decomposition method applied to a singularly perturbed problem with regular boundary layers, J. Comput. Appl. Math., 166, 13–29, (2004).

[21] I.P. Boglaev and V. Duoba, Iterative domain decomposition algorithms for a convection-diffusion problem, Comput. Math. Appl., 47, 501–518, (2004).

[22] I.P. Boglaev and S. Pack, An iterative domain decompistion algorithm for a nonlinear convection-diffusion problem, ANZIAM J. (E), 48, C494–C508, (2007).

[23] I.P. Boglaev and S. Pack, Parallel two-level Schwarz algorithm for a convection-diffusion problem. In: Proc. Fourth International Conference on Finite Difference Methods: Theory and Applications, The Rousse University, Rousse, Bulgaria, 149–154, 2007.

[24] I.P. Boglaev, Monotone ω-Jacobi and SUR Iterates and Applications to Semiliner Singularly Perturbed Problems, Reports in Mathematics 18, IFS, Massey University, Palmerston North, 2008.

[25] I.P. Boglaev and S. Pack, A uniformly cnovergent method on arbitrary meshes for a semilinear convection-diffusion problem with discontinuous data, Int. J. Numer. Anal. Model, 5, 24–39, (2008).

[26] I.P. Boglaev and S. Pack, Block mnotone domain decomposition methods for a quasi-linear anisotropic convection-diffusion equation, ANZIAM J. (E), 49, 493–512, (2008).

[27] A. Brandt, A Multi-level adaptive solutions to boundary-value problems, Math. Comput., (31), 333–390, (1977).

[28] T. Chan and T. Mathew, Domain decomposition algorithms, Acta Numerica, 61–143, (1994).

[29] D.N. de G Allen and R.V. Southwell, Relaxation methods applied to determine the motion, in two dimensions, of a viscous fluid past a fixed cylinder, Quart. J. Mech. Appl. Math, 8, 129–145, (1955).

[30] E.P. Doolan, J.J.H. Miller and W.H.A Schildrers, Uniform Numerical Methods for Problems with Initial and Boundary Layers, Boole Press, Dublin, 1980.

[31] P.A. Farrell, I.P. Boglaev and V.V. Sirotkin, Parallel domain decomposition methods for semi-linear singularly perturbed differential equations, Comput. Fluid Dyn. J., 2, 423–433, (1994).

[32] P.A. Farrell, A.F. Hegarty, J.J.H. Miller, E.O'Riordan and G.I. Shishkin, Robust Computational Techniques for Boundary Layers, Chapman & Hall/CRC, Florida, 2000.

[33] P.A. Farrell, A.F. Hegarty, J.J.H. Miller, E. O'Riordan and G.I. Shishkin, Singularly perturbed convection-diffusion problems with boundary and weak layers, J. Comput. Appl. Math., 166, 133–151, (2004).

[34] K.O. Friedrichs and W.R. Wasow, Singular perturbations of nonlinear oscillations, Duke Math. J., 13, 367–381, (1946).

[35] M. Garbey, A Schwarz alternatiing procedure for singular perturbation problems, SIAM J. Sci. Comput., 17, 1175–1201, (1996).

[36] M. Garbey and H.G. Kaper, Heterogeneous domain decomposition for singularly perturbed elliptic boundary value problems, SIAM J. Numer. Anal., 34, 1513–1544, (1997).

[37] M. Garbey, Yu.A. Kuznetsov and Yu.V. Vassilevski, A parallel Schwarz method for a convection-diffusion problem, SIAM J. Sci. Comput., 22, 891–916, (2000).

[38] F.G. Gaspar, C. Claverno, F. Lisbona, Some numerical experiments with multigrid methods on Shishkin meshes, J. Comput. Appl. Math., 138, 21–35, (2002).

[39] A.F. Hegarty, J.J.H. Miller, E. O'Riordan and G.I. Shishkin, Special meshes for finite difference approximations to an advection-diffusion equation with parabolic layers, J. Comput. Phys., 117, 47–54, (1995).

[40] V. John, A numerical study of a posteriori error estimates for convection-diffusion equations, Comput. Meth. Appl. Mech. Eng., 190, 757–781, (2000).

[41] Z. Jun, A new fast solver - Monotone MG method (MMG), J. Comp. Maths., 5, 325–335, (1987).

[42] M.K. Ladalbajoo and K.C.Patidar, A survey of numerical techniques for solving singularly perturbed ordinary differential equations, Appl. Math. Comput., 130, 457–510, (2002).

[43] M.K. Ladalbajoo and K.C.Patidar, Singularly perturbed problems in partial differential equations: a survey, Appl. Math. Comput., 134, 371–429, (2003).

[44] T. Linss, Uniform pointwise convergence of an upwind finite volume method on layer-adapted meshes, ZAMM, 82, 247–254, (2002).

[45] P.L. Lions, On the Schwarz alternating method. In: International Conference on Domain Decomposition, SIAM, Philedelphia, 47–70, 1989.

[46] S.H. Lui, On linear monotone iteration and Schwarz methods for nonlinear elliptic PDEs, Numer. Math, 93, 109–129, (2002).

[47] J. Mackenzie, Uniform convergence analysis of an upwind finite difference approximation of a convection-diffusion boundary value problem on an adaptive grid, IMA J. Numer. Anal., 19, 233–249, (1999).

[48] H. MacMullen, J.J.H. Miller, E. O'Riordan and G.I. Shishkin, Schwarz iterative method for convection-diffusion problems with boundary layers. In: Analytical and Numerical Methods for Convection-Dominated and Singularly Perturbed Problems (J.J.H. Miller, G.I. Shishkin, L.G. Vulkov, Eds), Nova Science Publishers, New York, 213–218, 2000.

[49] T.P. Mathew, Uniform convergence of the Schwarz alternating method for solving singularly perturbed advection-diffusion equations, SIAM J. Numer. Anal., 35, 1663–1683, (1998).

[50] J.J.H. Miller, E. O'Riordan and G.I. Shishkin, Fitted Numerical Methods for Singular Perturbation Problems, World Sciencific, Singapore, 1996.

[51] K.W. Morton, Numerical Solution of Convection-Diffusion Problems, Chapman & Hall, London, 1996.

[52] J.M. Ortega and W.C. Rheinbolt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York-London, 1970.

[53] C.V. Pao, Block monotone iterative methods for numerical solutions of nonlinear elliptic equations, Numer. Math., 72, 239–262, (1995).

[54] V. Pereyra and E.G. Sewell, Mesh selection for discrete solution of boundary problems in ordinary differential equations, Numer. Math., 23, 261–268, (1975).

[55] L. Prandtl, Über Flüssigkeitsbewegung bei sehr kleiner Reibung. In: Verhandlungen, III Inter. Math. Kongresses, Tuebner, Leipzig, 484–491, 1905.

[56] A. Quarteroni and V. Valli, Domain Decomposition Methods for Partial Differential Equations, Oxford University Press, Oxford, 1999.

[57] H.G Roos, M. Stynes and L. Tobiska, Numerical Methods for Singularly Perturbed Differential Equations, Springer-Verlag, Berlin, 1996.

[58] H.G. Roos, M. Stynes and L.Tobiska, Robust Numerical Methods for Singularly Perturbed Differential Equations, Second Edition, Springer-Verlag, Berlin-Heidelberg, 2008.

[59] R.D. Russell and J. Christiansen, Adaptive mesh selection strategies for boundary-value problems, SIAM J. Numer. Anal., 15, 59–80, (1978).

[60] Y. Saad and M Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput., 7, 856–869, (1986).

[61] A. Samarskii, The Theory of Difference Schemes, Marcel Dekker Inc., New York-Basel, 2001.

[62] G.I. Shishkin, Approximation of solutions of singularly perturbed boundary value problmes with a parabolic boundary layer, USSR Comput. Maths. Math. Phys., 29, 1–10, (1989).

[63] G.I. Shishkin, On finite difference fitted schemes for singularly perturbed boundary value problems with a parabolic boundary layer, INCA Preprint 4, INCA, Dublin, 1995.

[64] B.F. Smith, P.E. Bjorstad and W.D. Gropp, Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations, Cambridge University Press, New York, 1996.

[65] M. Stynes and E. O'Riordan, A finite element method for a singularly perturbed boundary-value problem, Numer. Math., 50, 1-15, (1986).

[66] M. Stynes, Numerical methods for convection-diffusion problems or the 30 years war. In: 20th Biennial Conference on Numerical Analysis, Numerical Analysis Report NA/217, University of Dundee, 95–103, 2003.

[67] M. Stynes, Steady-state convection-diffusion problems, Acta Numerica, 14, 445–508, 2005.

[68] U. Trottenberg, C.W. Oosterlee and A. Schuller, Multigrid, Academic Press, London, 2001.

[69] R.S. Varga, Matrix Iterative Analysis, Second Edition, Springer-Verlag, Berlin-Heidelberg, 2000.

[70] N.I. Vasil'ev and Yu.A. Klokov, Fundamentals of Boundary Value Problems for Ordinary Differential Equations, Zinatne, Riga, 1978 (in Russian).

[71] R.L. Voller, A monotone including multigrid-method, Computing, 45, 377–382, (1990).

[72] R. Vulanović, Mesh Construction for Discretization of Singularly Perturbed Boundary Value Problems, PhD Thesis, University of Novi Sad, 1986.

[73] R. Vulanović, A uniform method for quasilinear singular perturbation problems without turning points, Computing, 41, 97–106, (1989).

[74] R. Vulanović, A second order numerical method for nonlinear singular perturbation problems without turning points, USSR Comput. Maths. Math. Phys., 31, 522–532, (1991).

[75] R. Vulanović, Non-equidustant finite difference methods for elliptic singular preturbation problems. In: Computational Methods for Boundary and Interior Layers in Several Dimensions, (J.J.H. Miller, ed.), Boole Press, Dublin, 203–223, 1993.

[76] R. Vulanović, Fourth order algorithms for a semilinear singular perturbation problem, Numer. Algor., 16, 117–128, (1997).

[77] R. Vulanović, A priori meshes for singularly perturbed quasilinear two-point boundary value problems, IMA J. Numer. Anal., 21, 349–366, (2001).

[78] A.B. White, On selection of equidistributing meshes for two-point boundary-value probelms, SIAM J. Numer. Anal., 16, 472–502, (1979).

[79] D. Wollstein, T. Linss and H.-G. Roos, A uniformly accurate finite volume discretization for a convection-diffusion problem, ETNA, 13, 1–11, (2002).