

Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

# 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  $\omega$ -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

<b>Abstract</b>	<b>ii</b>
<b>Acknowledgements</b>	<b>v</b>
<b>List of figures</b>	<b>x</b>
<b>List of tables</b>	<b>xiv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Singularly perturbed convection-diffusion problems . . . . .	1
1.2 Domain decomposition algorithms . . . . .	5
1.3 General overview of the thesis . . . . .	8
1.4 Uniform convergent numerical methods for two dimensional convection-diffusion problems . . . . .	9
1.4.1 Convection-diffusion problem with elliptic boundary layers . . . . .	9
1.4.2 Convection-diffusion problem with parabolic layers . . . . .	12
1.4.3 Anisotropic convection-diffusion problem . . . . .	14
1.5 Nonlinear difference scheme . . . . .	16
1.6 Monotone iterative method . . . . .	19
<b>2 One dimensional convection-diffusion problem</b>	<b>23</b>
2.1 Introduction . . . . .	23
2.2 Properties of the continuous problem . . . . .	25
2.3 Construction of the difference scheme . . . . .	28
2.4 Uniform convergence of the difference scheme . . . . .	30
2.5 Monotone iterative method . . . . .	34

2.6	Numerical experiments . . . . .	39
2.6.1	Numerical observations . . . . .	42
2.7	Conclusions . . . . .	42
<b>3</b>	<b>Monotone domain decomposition algorithms</b>	<b>45</b>
3.1	Introduction . . . . .	45
3.2	Monotone domain decomposition algorithm . . . . .	46
3.2.1	Numerical experiments . . . . .	52
3.2.2	Convection-diffusion problem with parabolic boundary layers . . . . .	54
3.2.3	Anisotropic convection-diffusion problem . . . . .	56
3.2.4	Numerical observations . . . . .	57
3.3	Two-level monotone domain decomposition algorithm . . . . .	57
3.3.1	The outer iterates . . . . .	58
3.3.2	The inner iterates . . . . .	60
3.3.3	Numerical stability of the outer and inner iterates . . . . .	68
3.3.4	Numerical experiments . . . . .	69
3.4	Conclusions . . . . .	76
<b>4</b>	<b>Monotone relaxation methods</b>	<b>77</b>
4.1	Introduction . . . . .	77
4.2	Point monotone relaxation methods . . . . .	78
4.3	Block monotone iterative methods . . . . .	84
4.4	Comparison of the point monotone and block monotone iterative methods . . . . .	90
4.5	Numerical experiments . . . . .	91
4.5.1	Convection-diffusion problem with parabolic layers . . . . .	92
4.5.2	Anisotropic convection-diffusion problem . . . . .	92
4.5.3	Numerical observations . . . . .	96
4.6	Conclusions . . . . .	97
<b>5</b>	<b>Composite monotone domain decomposition algorithms</b>	<b>99</b>
5.1	Introduction . . . . .	99
5.2	Composite monotone domain decomposition algorithms based on the Jacobi and Gauss-Seidel methods . . . . .	100

5.3	Composite monotone domain decomposition algorithms based on the block Jacobi and block Gauss-Seidel methods . . . . .	108
5.4	Composite monotone two-level PDD and BDD algorithms . . . . .	115
5.5	Numerical experiments . . . . .	115
5.5.1	Convection-diffusion problem with parabolic boundary layers . . .	116
5.5.2	Anisotropic problem . . . . .	119
5.5.3	Numerical observations . . . . .	122
5.6	Conclusions . . . . .	122
<b>6</b>	<b>Multigrid methods</b>	<b>127</b>
6.1	Introduction . . . . .	127
6.2	Monotone multigrid method . . . . .	129
6.2.1	Numerical experiments . . . . .	134
6.3	Block monotone multigrid method . . . . .	139
6.3.1	Numerical experiments . . . . .	141
6.4	Two-level monotone multigrid method . . . . .	145
6.4.1	Numerical experiments . . . . .	147
6.5	Conclusions . . . . .	152
<b>7</b>	<b>Conclusions</b>	<b>159</b>
<b>A</b>	<b>Theorem proofs from Chapter 4</b>	<b>165</b>
<b>B</b>	<b>Theorem proofs from Chapter 5</b>	<b>173</b>



# List of Figures

1.1	Illustration of the domain decomposition with overlapping subdomains. . . . .	6
1.2	The domain decomposition with nonoverlapping subdomains. . . . .	7
1.3	Solution of the convection-diffusion example with elliptic boundary layers. . . . .	10
1.4	Solution of the convection-diffusion example with parabolic boundary layers. . . . .	13
1.5	Solution of the anisotropic convection-diffusion example. . . . .	15
2.1	$E_{N,\varepsilon}(x)$ with $N=128$ and $\varepsilon = 10^{-2}, 10^{-3}$ for the test problem 1. . . . .	41
2.2	$E_{N,\varepsilon}$ with $N=128$ and $\varepsilon = 10^{-2}, 10^{-3}$ for the test problem 2. . . . .	43
3.1	Fragment of the domain decomposition with overlapping subdomains $\Omega_{m-1}, \Omega_m, \Omega_{m+1}$ and overlaps $\theta_{m-1}, \theta_m$ . . . . .	47
3.2	(a) Location of the overlap on the left. (b) Location of the overlap on the right. . . . .	53
3.3	Serial acceleration of the monotone DD algorithm for the test problem (3.6). . . . .	56
3.4	Serial acceleration of the monotone DD algorithm for the test problem (3.8). . . . .	57
3.5	Fragment of the box-domain decomposition. . . . .	61
3.6	Serial acceleration of the two-level domain decomposition algorithm for the test problem (3.8). . . . .	74
3.7	Parallel speedup of the two-level domain decomposition algorithm with $\varepsilon = 10^{-2}$ for the test problem (3.8). . . . .	75
4.1	Execution times of the monotone SUR and monotone BSUR methods, for the test problem (3.6), over varying values of $\omega$ , $N = 128$ and $\varepsilon = 0.001$ . . . . .	93

4.2	Execution times of the monotone SUR and monotone BSUR method for the test problem (3.6). . . . .	94
4.3	Execution times of the monotone SUR and monotone BSUR methods, for the test problem (3.8), over varying values of $\omega$ , $N = 128$ and $\varepsilon = 0.001$ . . . . .	95
4.4	Execution times of the monotone SUR and monotone BSUR method for the test problem (3.8). . . . .	96
5.1	Fragment of the domain decomposition with overlapping subdomains $\Omega_{m-1}$ , $\Omega_m$ , $\Omega_{m+1}$ and overlaps $\theta_{m-1}$ , $\theta_m$ . . . . .	101
5.2	Serial acceleration of the monotone BSUR method for the test problem (3.6). . . . .	117
5.3	Serial acceleration of the monotone BDD algorithm for the test problem (3.6). . . . .	118
5.4	Serial acceleration of the monotone BDD algorithm, the monotone DD algorithm and the monotone BSUR method, for the test problem (3.6). . . . .	121
5.5	Serial acceleration of the monotone BSUR method for the test problem (3.8). . . . .	121
5.6	Serial acceleration of the monotone BDD algorithm for the test problem (3.8). . . . .	123
5.7	Serial acceleration of the monotone BDD algorithm, the monotone DD algorithm and the monotone BSUR method for the test problem (3.8). . . . .	126
6.1	Cycle counts of the MMG method for $N = 64$ and varying values of $\varepsilon$ for the test problem (6.17). . . . .	136
6.2	Execution times of the MMG and SUR methods for $N = 64$ and varying values of $\varepsilon$ for the test problem (6.17). . . . .	137
6.3	Cycle counts of the MMG method for $N = 64$ and varying values of $\varepsilon$ for the test problem (6.19). . . . .	138
6.4	Execution times of the MMG and SUR methods for $N = 64$ and varying values of $\varepsilon$ for the test problem (6.19). . . . .	139
6.5	Cycle counts of the BMMG method for $N = 64$ and varying values of $\varepsilon$ for the test problem (6.17). . . . .	142
6.6	Execution times of the BMMG and BSUR methods for varying values of $\varepsilon$ for the test problem (6.17). . . . .	143
6.7	Cycle counts of the BMMG method for $N = 64$ and varying values of $\varepsilon$ for the test problem (6.19). . . . .	144

6.8 Execution times of the BMMG and BSUR methods for varying values of $\varepsilon$ for the test problem (6.19). . . . .	145
--	-----



# List of Tables

2.1	Maximal approximate error $\overline{E}_{N,\varepsilon}$ for the monotone iterative method (2.19) applied to the test problem 1. . . . .	40
2.2	Iteration counts for the monotone iterative method (2.19) applied to the test problem 1. . . . .	41
2.3	Maximal approximate error $\overline{E}_{N,\varepsilon}$ for the monotone iterative method (2.19) applied to the test problem 2. . . . .	42
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. . . . .	54
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). . . . .	55
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). . . . .	58
3.4	Outer iteration counts using the minimum and maximum overlap size, above and below the line, respectively, for the test problem (3.6). . . . .	71
3.5	Execution times using the minimum and maximum overlap size, above and below the line, respectively, for the test problem (3.6). . . . .	72
3.6	Outer iteration counts using the minimum and maximum overlap size, above and below the line, respectively, for the test problem (3.8). . . . .	73
3.7	Parallel execution times using the minimum and maximum overlap size, above and below the line, respectively, for the test problem (3.8). . . . .	73

4.1	Iteration counts of the monotone SUR and monotone BSUR methods for the test problem (3.6). . . . .	93
4.2	Iteration counts of the monotone SUR and monotone BSUR methods for the test problem (3.8). . . . .	95
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. . .	118
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. . .	119
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. . .	120
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. . .	123
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. . .	124
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. . .	125

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). . . . .	149
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). . . . .	150
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). . . . .	151
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). . . . .	152
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). . . . .	153
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). . . . .	154
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). . . . .	155
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). . . . .	156

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). . . . .	157
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). . . . .	158