

```
/*
 * tidy_peripheral_curves.c
 *
 *
 * This file provides the function
 *
 * void tidy_peripheral_curves(Triangulation *manifold);
 *
 * which is used within the kernel to clean up a set of peripheral
 * curves.
 *
 * Functions which alter triangulations maintain a set of peripheral
 * curves which is technically correct, but suffers two faults.
 * A minor fault is that the peripheral curves wind around a lot,
 * and therefore create unnecessarily complicated cusp equations.
 * The more serious fault is that the curves may evolve trivial loops;
 * that is, a single meridian or longitude will have several components,
 * one of which is the correct curve while the others are homotopically
 * trivial loops. Such trivial loops introduce erroneous multiples
 * of  $2\pi$  into the computed holonomies, and thereby foul up the
 * computation of hyperbolic structures.
 *
 * The function tidy_peripheral_curves()
 *
 * (1) makes a copy of the existing peripheral curves,
 * (2) calls peripheral_curves() to create a nice set of
 * peripheral curves, and
 * (3) expresses the original curves as linear combinations
 * of the nice curves.
 *
 * The result is an "efficient" set of curves, with no trivial
 * loops.
 */

#include "kernel.h"

/*
 * scratch_curves[0] will store the original peripheral curves.
 * scratch_curves[1] will store the nice peripheral curves.
 */

#define original_curves 0
#define nice_curves 1

static void compute_new_curves(Triangulation *manifold);

void tidy_peripheral_curves(
    Triangulation *manifold)
{
    /*
     * Copy the original peripheral curves to the
     * scratch_curve[original_curves] fields of the Tetrahedra.
     */
    copy_curves_to_scratch(manifold, original_curves, TRUE);

    /*
     * Compute a nice set of peripheral curves.
     */
    peripheral_curves(manifold);

    /*
     * Copy the nice peripheral curves to the
     * scratch_curve[nice_curves] fields of the Tetrahedra.
     */
    copy_curves_to_scratch(manifold, nice_curves, FALSE);

    /*
     * Compute the intersection numbers of the original curves
     * with the nice curves.
     */
    compute_intersection_numbers(manifold);
}
```

```

/*
 * Compute the new curves as linear combinations of the
 * nice curves.
 */
compute_new_curves(manifold);
}

static void compute_new_curves(
    Triangulation *manifold)
{
    Tetrahedron *tet;
    int h,
        i,
        j,
        k;

    for (tet = manifold->tet_list_begin.next;
         tet != &manifold->tet_list_end;
         tet = tet->next)

        for (h = 0; h < 2; h++) /* which curve */
            for (i = 0; i < 2; i++) /* which sheet */
                for (j = 0; j < 4; j++) /* which vertex */
                    for (k = 0; k < 4; k++) /* which side of that vertex */

                        tet->curve[h][i][j][k] =
                            (j == k) ?
                                0 :
                                - tet->cusp[j]->intersection_number[h][L]
                                  * tet->scratch_curve[nice_curves][M][i][j][k]
                                + tet->cusp[j]->intersection_number[h][M]
                                  * tet->scratch_curve[nice_curves][L][i][j][k];
}

```