Efficient Generation of Oriented Meshes for Solving Convection-Diffusion Problems*

Neil Madden† and Martin Stynes‡

April 29, 1996

Department of Mathematics, University College, Cork,
Ireland

Abstract

A description of Shishkin meshes for resolving boundary and interior layers is given. It is shown how PLTMG can be used to construct these special oriented meshes with minimal effort. Several types of singularly perturbed convection-diffusion problem are solved on the meshes. These solutions are compared with those obtained on uniform meshes and on meshes adaptively refined by PLTMG; it is seen that Shishkin meshes yield much more accurate solutions with little additional computational effort.

Keywords: numerical analysis, convection-diffusion, singularly perturbed problem, oriented mesh, Shishkin mesh, PLTMG.

Mathematics Subject Classification (1991): Primary 65N50, Secondary 76M99, 76R99

*Research supported by FORBAIRT Grant SC/94/215

†Email address n.madden@ucc.ie

‡Email address stynes@ucc.ie

1 Introduction

Singularly perturbed problems are difficult to solve numerically [1, 12]. To approximate their solutions accurately without excessive computational effort, two approaches have been developed:

(i). the use of specialised methods on standard meshes (e.g. [3, 9]);

(ii). the use of standard methods on highly non-uniform meshes.

Non-uniform meshes are constructed adaptively or a priori. Specially adapted meshes can be generated isotropically (e.g., Bank [2]) or anisotropically as described by Kornhuber and Roitzsch [10]. The latter approach is found to yield solutions that are more stable for less computational cost. Meshes specified a priori may be graded (an idea due to Bahkvalov [5, 15]) or piecewise uniform (a simpler approach introduced by Shishkin [13, 6]). In this paper, we examine the practical implementation of Shishkin meshes when dealing with linear elliptic convection-diffusion problems in two dimensions. Unlike some other methods, the Shishkin mesh approach is underpinned by rigorous analysis [11, 13].

While the convergence theory for upwinding schemes on Shishkin meshes is well-established [13], these meshes have been implemented only in the case of boundary layers [7, 8]. Interior layers (i.e., layers that do not lie along the boundary of the domain) pose a much more difficult problem, because the theoretical construction of the Shishkin mesh is then difficult to achieve in practice.

In the present paper, we show how to use the PLTMG package to construct approximate Shishkin meshes with minimal programming effort; our numerical results show that these meshes yield remarkably accurate solutions to convection-diffusion problems without using a large number of nodes. In particular, we show that our meshes yield much better results than uniform and adaptively refined meshes that contain a similar number of points.

2 PLTMG

All examples in this paper are solved using the software package PLTMG [2]. This package, developed by R. E. Bank et al., is designed for elliptic partial differential equations.

For a particular problem, the user must code FORTRAN subroutines and functions that specify the equation and mesh. These are compiled and linked with the main PLTMG object, producing an executable image. When executed, the package uses an exponentially upwinded finite element method from [3] to discretize the problem on this mesh. It then uses its own facilities to solve the resulting discrete system and to display the solution graphically.

PLTMG can be obtained as FORTRAN source code from any *netlib* mirror site.

When calling the PLTMG routines, the user may request that the specified mesh be refined uniformly. With one level of uniform refinement, each user-defined triangle is replaced with four triangles of the same shape. If two levels of refinement are requested, then each of these four triangles is replaced with four more similar triangles, and so on. In this way, one can generate a mesh with many elements that retains the geometry of the initial coarse mesh. We shall extensively use this facility of the package to prepare specialised fitted meshes for problems with boundary and interior layers.

PLTMG also possesses a powerful mechanism whereby, after the solution to a problem has been computed, the mesh can be altered to produce one more suited to the given problem. In the standard usage, the user specifies a (usually uniform) mesh. The package first solves the problem on this mesh. Then, using calculations based on local a posteriori error estimates, it modifies the mesh in one or both of the following ways:

(i). Adaptive refinement/unrefinement: with this approach, PLTMG endeavours to produce a suitable mesh containing approximately the number of nodes that the user requested.

(ii). Node movement: here PLTMG neither refines nor unrefines the mesh, but rather alters the (x, y) coordinates of the nodes in an attempt to improve the mesh.

PLTMG then computes a solution on this new mesh. In our experiments, we found that the best results are obtained when (i) and (ii) are used iteratively.

3 Test Problems

In this section we shall present a number of examples, whose solutions are computed over both uniform and adaptively refined meshes. All problems considered are of the form

$$\begin{aligned} -\varepsilon\Delta u + \vec{\beta}\cdot\nabla u = f \quad & \text{in} \quad \Omega = (0,1)^2, \\ u = g \quad & \text{on} \quad \partial\Omega_1, \\ (\nabla u + \vec{\beta}u)\cdot n = 0 \quad & \text{on} \quad \partial\Omega\setminus\partial\Omega_1, \end{aligned} \tag{1}$$

where ε is a parameter that satisfies $0 < \varepsilon \ll 1$, $\vec{\beta} \in (C(\bar{\Omega}))^2$, $f \in C^2(\Omega)$, g is defined pointwise on $\partial\Omega_1$, and $\partial\Omega_1 \subseteq \partial\Omega$. We assume that $\vec{\beta}$ is $O(1)$, so the problem is convection-dominated. In our examples, the data of the problem will be such that (1) has a unique solution.

If $\vec{\beta}$ points outwards on part of $\partial\Omega_1$, then in general the solution u of (1) will have an *exponential* boundary layer here [4]. If $\vec{\beta}$ is parallel to part of $\partial\Omega_1$, this usually causes a *parabolic* boundary layer in the solution [4].

Example 3.1 We consider (1) with

$$\varepsilon = 10^{-3}, \quad \vec{\beta} = (1 + x^2 + y^2, 0) \quad \text{and } f \equiv 0,$$

and the Dirichlet boundary conditions

$$u(x,0) = (1-x)^3, \quad u(x,1) = (1-x)^2, \quad u(0,y) = 1 \quad \text{and } u(1,y) = 0.$$

The solution to this problem has an exponential boundary layer at the side $x = 1$ of $\bar{\Omega}$ and parabolic boundary layers at $y = 0$ and $y = 1$.

We used PLTMG to solve this problem on a uniform 16×16 mesh, obtaining the solution shown in Figure 1. It is adequate on most of $\bar{\Omega}$, but spurious oscillations can be seen along the parabolic boundary layers. We needed to use a 256×256 uniform mesh to obtain a satisfactory computed solution; as we shall see later, a similar result can be computed on a Shishkin mesh with far fewer mesh points. □

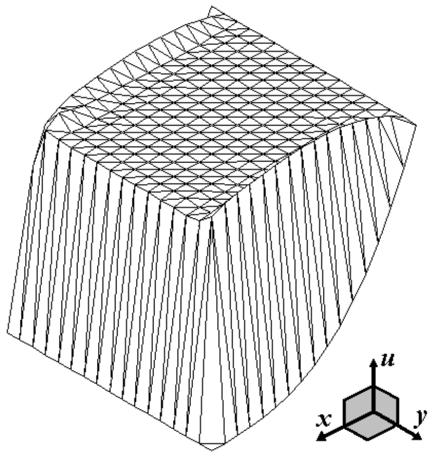

Figure 1: Solution to Example 3.1 on a uniform 16×16 mesh.

Example 3.2 We now introduce a discontinuity in the inflow boundary data. This discontinuity will be propagated across $\bar{\Omega}$, giving rise to an interior layer.

We shall take $\varepsilon = 10^{-6}$, $\vec{\beta} = (1,3)$ and $f \equiv 0$, with the boundary data

$$
\begin{aligned}
u = 1 \quad &\text{on} \quad \{(x,y) \in \partial\Omega : 0 \le x \le 0.2, y = 0\} \cup \{(x,y) \in \partial\Omega : x = 0, 0 \le y \le 1\}, \\
u = 0 \quad &\text{on} \quad \{(x,y) \in \partial\Omega : 0.2 < x \le 1, y = 0\},
\end{aligned}
$$

and homogeneous Neumann boundary conditions elsewhere. The solution to this problem has an interior layer but no boundary layers.

Supplying PLTMG with a uniform mesh having 16 intervals along the x and y-axes, we computed the approximate solution shown in Figure 2. While the interior layer is clearly visible, this solution is nevertheless unsatisfactory because of the non-physical oscillations that it exhibits. We also computed the approximate solution on uniform meshes with up to 256 intervals along each axis, but although the interior layer became sharper, overshoots of between 10 and 20 percent could still be observed.

To study how adaptive refinement copes with this problem, we solved it starting from a uniform mesh with 289 nodes. Next we requested PLTMG to move the nodes within the region to provide a more appropriate mesh. A new solution was computed using this mesh. We now requested PLTMG to adaptively refine this mesh based on the most recently obtained solution before computing a third solution. This process of moving the

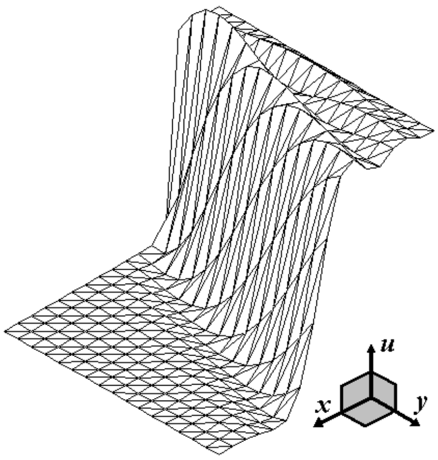

Figure 2: Solution to Example 3.2 on a uniform 16×16 mesh.

mesh, computing an approximation, adaptive refinement and recomputation was carried out several more times until the solution in Figure 3 was reached. It contains 496 nodes. Notice the irregular spacing of the node points in the area away from the layer and the density of the elements in the region of the layer. The solution in Figure 3 is clearly better than that of Figure 2; the layer is much sharper and the magnitude of the oscillations near the layer is smaller. □

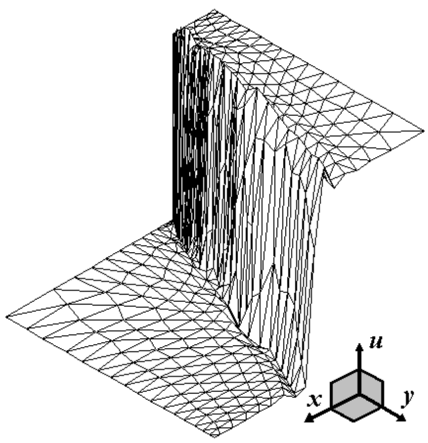

Figure 3: Solution to Example 3.2 using adaptive refinement.

Example 3.3 Finally, we consider a problem with a *curved* interior layer. This is done by making $\vec{\beta}$ variable in Example 3.2; we took $\vec{\beta} = (1, 3x)$. The rest of the data in Example 3.2 is unchanged.

In Figure 4, we give the solution computed by PLTMG over a uniform 16×16 mesh. It has shortcomings similar to Figure 2. As is Example 3.2, increasing the number of nodes in the uniform mesh did not greatly reduce the magnitude of the oscillations.

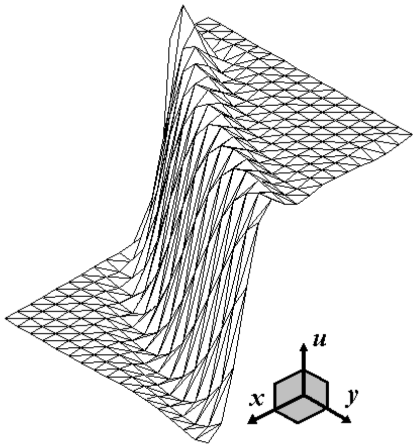

Figure 4: Solution to Example 3.3 on a uniform 16×16 mesh.

We applied the adaptive process of Example 3.2 to this problem. The final mesh contained 501 nodes and was used to compute the solution shown in Figure 5. Again the result is a notable improvement on the solution computed on the uniform mesh. Though oscillations are still present near the layer, they are visibly smaller than in Figure 4. □

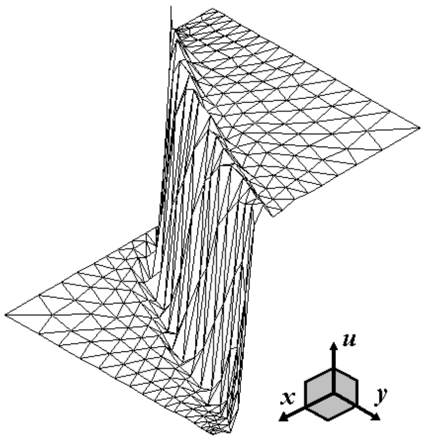

Figure 5: Solution to Example 3.3 using adaptive refinement.

4 Shishkin Meshes

In [13] and earlier papers, Shishkin introduced a special mesh that is designed to yield convergence inside layers for reasonable discretizations of (1). This piecewise uniform mesh is very fine inside "most" of the layer and coarse otherwise. No attempt is made to change the mesh size smoothly; the fine and coarse meshes are contiguous. Shishkin uses a simple first-order upwinding scheme on his mesh. Under strong a priori assumptions on the behaviour of u, he is able to prove a global convergence result that is uniform in ε [13].

We now describe the construction of Shishkin meshes. Consider a problem whose solution has a boundary layer at the side $x = 1$ of $\bar{\Omega}$. In this case, a suitable Shishkin mesh is the following.

Assume that N is even and that $N \geq 4$. If the boundary layer is of exponential type, we set

$$\lambda = C\varepsilon \ln N, \tag{2}$$

while if it is of parabolic type we set

$$\lambda = \sqrt{\varepsilon} \ln N. \tag{3}$$

We shall refer to λ as the *fine mesh thickness*; it is the width of the region where the mesh will be very fine. The optimal choice of C is not precisely determined by current analyses, so we used numerical experiments to find a good choice of C in each case. Note that λ will be small when $\varepsilon \ll N^{-1}$.

Along the x-axis, we divide both $[0, 1-\lambda]$ and $[1-\lambda, 1]$ into $N/2$ equal subintervals. That is, our mesh points are

$$x_i = \begin{cases} 2(1-\lambda)N^{-1}i, & \text{for } i = 0, \ldots, N/2, \\ \\ 1 - \lambda + 2\lambda N^{-1}(i - N/2), & \text{for } i = N/2+1, \ldots, N. \end{cases}$$

We take a uniform mesh with N subintervals on the y-axis. Form the tensor product of these two meshes. Then divide each mesh rectangle into two triangles by drawing a diagonal from north-west to south-east, say.

For $x \geq 1 - \lambda$, the mesh-width in the x-direction is $O(N^{-1}\lambda)$, which is very small. Thus the triangles of our mesh are tall and thin in the region $[1-\lambda, 1] \times [0, 1]$ (see Figure 6).

Note that as ε gets smaller, the aspect ratio of these triangles increases (it is of order $1/(\varepsilon \ln N)$).

The choice of C in (2) is motivated by an examination of interpolation error on this mesh [6]. Note that a typical exponential boundary layer has width $O(\varepsilon \ln(1/\varepsilon))$, so for $\varepsilon << N^{-1}$, the mesh does *not* resolve all of the boundary layer.

Next, consider a problem with an interior layer that runs, say, from a point on the edge $y = 0$ of $\bar{\Omega}$ to a point on the edge $y = 1$. Interior layers are asymptotically similar to parabolic boundary layers [4], so we shall take λ as in (3). Shishkin's analysis of this type of problem makes local changes of variable to map patches of the interior layer onto layers parallel to the coordinate axes. While this is theoretically possible, it seems impractical to implement; we know of no numerical results that have been obtained through following this procedure. For ease of computation, we propose to use instead what we call *Shishkin-type* meshes. These differ from Shishkin meshes in that the short sides of our triangles are not exactly perpendicular to the direction of propagation of the layer.

Our mesh points are found by first dividing the y-axis into N equal intervals by the points $y_j = j/N$, for $j = 0, ..., N$. Then each of the $N + 1$ horizontal lines $y = y_j$ is divided into N intervals by the points whose x-coordinates are

$$x_i = \begin{cases} 4(v - \lambda/2)N^{-1}i, & \text{for } i = 0, \ldots, N/4, \\ \\ v - \lambda/2 + 2\lambda N^{-1}(i - N/4), & \text{for } i = N/4 + 1, \ldots, 3N/4, \\ \\ v + \lambda/2 + 4(1 - v - \lambda/2)N^{-1}(i - 3N/4), & \text{for } i = 3N/4 + 1, \ldots, N, \end{cases}$$

where (v, y_j) is the point where the layer intersects the horizontal line.

The domain is then triangulated with the $(N + 1)^2$ points that this process yields, as shown for Examples 3.2 and 3.3 in Figures 10 and 12 respectively. As in Figure 6, the mesh triangles are long and thin inside the layer, but unlike Figure 6, the shortest side of each triangle is not necessarily perpendicular to the direction of propagation of the layer.

It should be noted that, though only linear problems are examined in this paper, the application of Shishkin meshes to quasilinear problems has also been considered [14]. Nevertheless, we do not know of any numerical implementation of Shishkin meshes for such a problem.

5 Shishkin Meshes for Boundary Layers

Recall Example 3.1, where we considered (1) with

$$\varepsilon = 10^{-3}, \quad \vec{\beta} = (1 + x^2 + y^2, 0) \quad \text{and } f \equiv 0,$$

with the Dirichlet boundary conditions

$$u(x, 0) = (1 - x)^3, \quad u(x, 1) = (1 - x)^2, \quad u(0, y) = 1 \quad \text{and } u(1, y) = 0.$$

We saw that the solution of this example has an exponential layer along the side $x = 1$ and parabolic layers along the sides $y = 0$ and $y = 1$. As described in the last section, a Shishkin mesh deals with the exponential layer by dividing each of the x-axis intervals $[0, 1 - \lambda_x]$ and $[1 - \lambda_x, 1]$ into $N/2$ equal subintervals. The parabolic layers are handled by dividing each of $[0, \lambda_y]$ and $[1 - \lambda_y, 1]$ into $N/4$ equal subintervals, and $[\lambda_y, 1 - \lambda_y]$ into $N/2$ subintervals. The tensor product of the two meshes is taken, yielding N^2 rectangles. These are then bisected to produce the final triangulation.

We now show how PLTMG's uniform refinement facilities can be used to lessen significantly the labour involved in constructing such a mesh.

We divide the interval $[0, 1]$ of the x-axis into four subintervals by the points 0, $(1 - \lambda_x)/2$, $1 - \lambda_x$, $1 - \lambda_x/2$, and 1. We divide the interval $[0, 1]$ of the y-axis into four subintervals by the points 0, λ_y, 0.5, $1 - \lambda_y$ and 1. Taking the tensor product of these one-dimensional grids yields a division of $\bar{\Omega}$ into 16 rectangles. We bisect each of these into two triangles to obtain a grid like that of Figure 6.

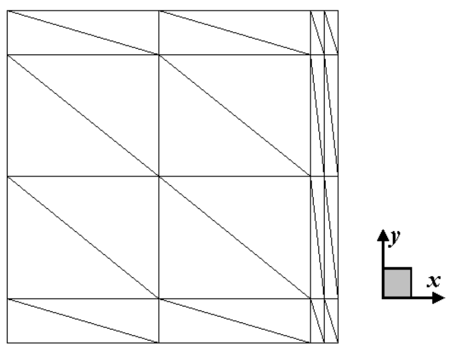

Figure 6: Initial mesh for Example 3.1.

As described in (2) and (3), we take $\lambda_x = C\varepsilon \ln N$ and $\lambda_y = \sqrt{\varepsilon} \ln N$. We choose the constant C to be 0.65 — a choice motivated by experimental work. (For $C \leq 0.60$, the outflow layer is visibly smeared, while for $C \geq 0.70$, overshoots are evident in the solution). It should be noted that in equation (4.6) of [13], Shishkin assumes a condition on C that for our problem becomes "$C \geq 1$". This condition is sufficient but not necessary for this analysis, so it does not contradict our choice of $C = 0.65$. It would be of some interest to analytically determine precisely the optimal choice of C, but this seems quite difficult.

We now call on PLTMG to refine this mesh uniformly twice, producing a mesh with 16 intervals along each axis as displayed in Figure 7. Note that this is much more convenient than explicitly programming the same mesh; furthermore, if we wish, we can easily refine the mesh again uniformly to obtain a finer Shishkin mesh.

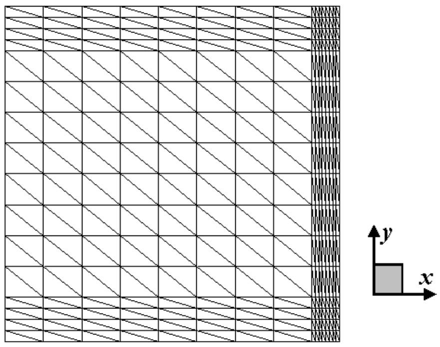

Figure 7: Shishkin mesh for Example 3.1.

The package then computes an approximate solution to the boundary value problem on this mesh. The solution, given in Figure 8, is quite an improvement on the solution shown in Figure 1, which was computed on a uniform mesh with the same number of nodes. In particular, notice how much better the layers are resolved and how the oscillations of Figure 1 have almost completely disappeared.

6 Shishkin-type Meshes for Straight Interior Layers

If the solution to our convection-diffusion problem has an interior layer that is not parallel to any edge of $\bar{\Omega}$, then it is quite awkward to construct an exact Shishkin mesh. We consider Example 3.2, whose solution contains such a layer. As we shall see, PLTMG's uniform refinement capabilities permit the task of generating a Shishkin-type mesh for this problem (as described in Section 4) to be greatly simplified.

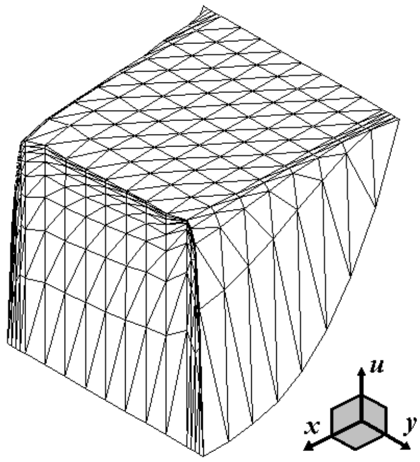

Figure 8: Solution to Example 3.1 using mesh of Figure 7.

In (1) we take $\varepsilon = 10^{-6}$, $\vec{\beta} = (1,3)$ and $f \equiv 0$. The boundary data is

$$\begin{aligned} u = 1 \quad & \text{on} \quad \{(x,y) \in \partial\Omega : 0 \leq x \leq 0.2, y = 0\} \cup \{(x,y) \in \partial\Omega : x = 0, 0 \leq y \leq 1\}, \\ u = 0 \quad & \text{on} \quad \{(x,y) \in \partial\Omega : 0.2 < x \leq 1, y = 0\}, \end{aligned}$$

and homogeneous Neumann boundary conditions elsewhere.

Our Shishkin-type mesh for this problem should have half of its elements in a region of thickness $\lambda = \sqrt{\varepsilon} \ln N$ straddling the layer, where N is still the number of subintervals along the x-axis. To do this, we divide the side $y = 0$ of $\bar{\Omega}$ into 4 subintervals by the points 0, $0.2 - \lambda/2$, 0.2, $0.2 + \lambda/2$ and 1. We divide $y = 1$ by the points 0, $v - \lambda/2$, v, $v + \lambda/2$ and 1. Here $v = 0.2 + 1/3$ is the x-coordinate where the layer meets $y = 1$. We triangulate $\bar{\Omega}$ using a division like that of Figure 9. The thin triangles in this triangulation have short sides not quite perpendicular to $\vec{\beta}$, which is why we use the term *Shishkin-type* and not *Shishkin* to describe this mesh.

PLTMG is now called, and the initial mesh is uniformly refined twice. The resulting mesh (similar to Figure 10) has 16×4 nodes, 16 in the x-direction and 4 in the y-direction. Thus we have, with reasonable ease, generated a Shishkin-type mesh containing 85 points from an initial mesh containing only 10.

Finally, we use PLTMG to compute the solution on this mesh. The result is displayed in Figure 11. We see that the interior layer has been computed with remarkable sharpness

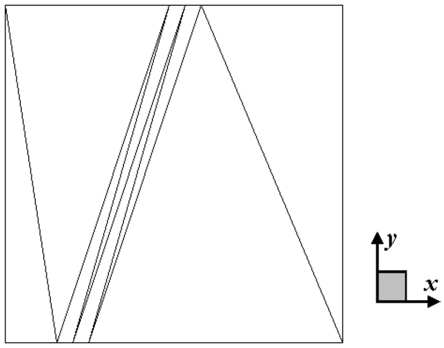

Figure 9: Initial mesh for Example 3.2.

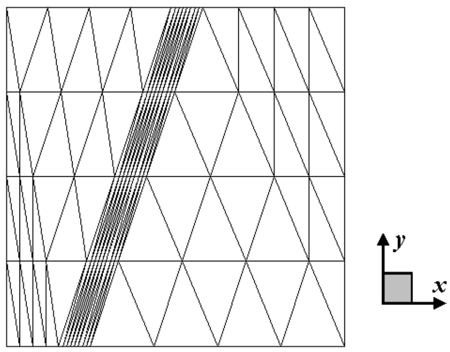

Figure 10: Shishkin-type mesh for Example 3.2.

and no oscillations are visible; contrast this with the imprecision exhibited in Figures 2 and 3.

7 Shishkin-type Meshes for Curved Interior Layers

We now extend the constructive approach of the previous section to the case of Example 3.3, whose solution has a curved interior layer.

For a fitted mesh to be effective for this type of problem, the thin triangles near the layer must follow the curve of the layer as closely as possible. Uniform refinement, however, divides a triangle into smaller ones of the same shape. Thus the new smaller triangles in any refined mesh will not follow the layer any more closely than their crude predecessors; that is, uniform refinement of an initial coarse mesh for a curved interior layer does not yield a mesh that is nearer to a Shishkin mesh. Consequently, we take a different approach.

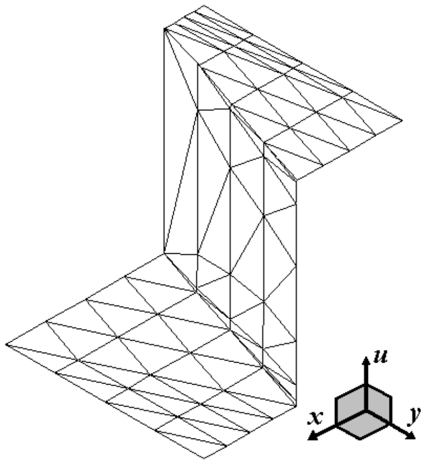

Figure 11: Solution of Example 3.2 on the Shishkin-type mesh.

We first determine precisely the location of the interior layer. Now the convective flow through the region at any point (x, y) is determined by the vector $\vec{\beta} = (1, 3x)$. Consequently, the layer through $\bar{\Omega}$ that propagates from the point $(0.2, 0)$ is given by $y = 3x^2/2 - 0.06$. We rewrite this as

$$x = \sqrt{\frac{2}{3}(y + 0.06)}. \tag{4}$$

Let $N = 4i$, where i is a positive integer. We first divide the interval $[0, 1]$ on the y-axis into N subintervals by the points $y_j = j/N$, where $j = 0, ..., N$. Next we divide each of the lines $y = y_j$ into four subintervals by the points 0, $v_x - \lambda/2$, v_x, $v_x + \lambda/2$, 1. Here $\lambda = \sqrt{\varepsilon} \ln N$ and v_x is the value of x computed from (4) with $y = y_j$. Now we divide each of these four subintervals into $N/4$ smaller subintervals, yielding a total of N intervals on each horizontal line. The mesh is then triangulated using the $(N + 1)^2$ points in our construction. We show a mesh generated in this fashion with $N = 16$ in Figure 12.

We solved the problem of Example 3.3 on this mesh. The result is shown in Figure 13. We see that our Shishkin-type mesh yields a sharp curved layer in the solution. No oscillations are visible. The number of mesh points is 289, which is the same as in Figure 4 and fewer than in Figure 5; nevertheless the solution shown in Figure 13 is far superior to the results displayed in these earlier Figures.

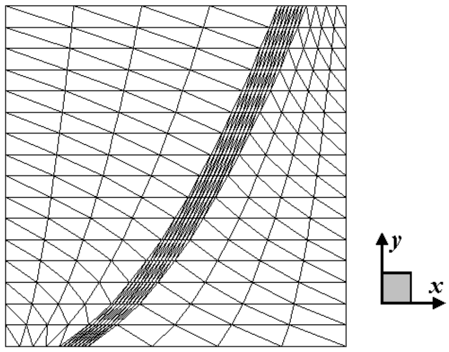

Figure 12: The Shishkin-type mesh for Example 3.3

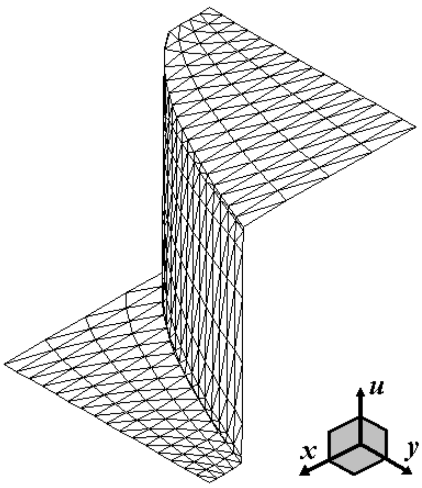

Figure 13: Solution of Example 3.3 on the Shishkin-type mesh

8 Conclusion

The analysis in [13] proves that boundary and interior layers can be computed very accurately with the use of Shishkin meshes. This has been demonstrated experimentally for boundary layers in such papers as [7, 8]. Hitherto, however, there were no published computations examining the effectiveness of this approach when dealing with interior layers. We have now shown how, using the PLTMG package, Shishkin-type meshes may be easily generated for problems with boundary and interior layers.

On comparing results obtained on Shishkin-type meshes with those yielded by uniform and adaptively refined meshes having a comparable number of nodes, we see that the Shishkin approach is far superior; the layers are much sharper and no oscillations are observed near layers. Note that little extra coding is required in generating the Shishkin-type meshes; in fact, except for the Shishkin-type mesh used in Section 7, the same amount of coding was required for generating the uniform mesh. Also the computational time required for each approach is comparable.

Thus we have shown that Shishkin-type meshes are easily constructed using PLTMG and that they yield accurate solutions to convection-diffusion problems with boundary and interior layers while using a small number of mesh nodes.

References

[1] I. Babuška and M. Suri, *On locking and robustness in the finite element method*, SIAM J. Numer. Anal. 29 (1992), 1261–1293

[2] R.E. Bank, *PLTMG: A Software Package for Solving Elliptic Partial Differential Equations. Users' Guide 7.0*, SIAM, Philadelphia (1994).

[3] R.E. Bank, J. F. Bürgler, W. Fichtner and R. K. Smith, *Some upwinding techniques for finite element approximations of convection-diffusion equations*, Numer. Math. 58 (1990), 185–202

[4] W. Eckhaus, *Matched Asymptotic Expansions and Singular Perturbations*, North-Holland, Amsterdam (1977).

[5] E.C. Gartland, Jr., *Graded-mesh difference schemes for singularly perturbed two-point boundary value problems*, Math. Comp. 51, (1988) 631–658

[6] W. Guo, M. Stynes, *Pointwise error estimates for a streamline diffusion scheme on a Shishkin mesh for a convection-diffusion problem*, IMA J. Numer. Anal. (to appear)

[7] A.F. Hegarty, J.J.H. Miller, E. O'Riordan, G.I. Shishkin, *Use of central-difference operators for solution of singularly perturbed problems*, Comm. Numer. Methods Engrg. 10 (1994), 297–302

[8] A.F. Hegarty, J.J.H. Miller, E. O'Riordan, G.I. Shishkin, *Special meshes for finite difference approximations to an advection-diffusion equation with parabolic layers,* J. Comp. Phys. 117, (1994), 47–54

[9] A.F. Hegarty, E. O'Riordan, M. Stynes, *A comparison of uniformly convergent difference schemes for two-dimensional convection-diffusion problems*, J. Comp. Phys. 105 (1993), 24–32

[10] R. Kornhuber, R. Roitzsch, *On adaptive grid refinement in the presence of internal and boundary layers*, IMPACT Comput. Sci. Eng. 2 (1990), 40–72

[11] J.J.H. Miller, E. O'Riordan, G.I. Shishkin, *Solution of Singular Perturbation Problems with ε-Uniform Numerical Methods — Introduction to the Theory for Linear Problems in One and Two Dimensions*, World Scientific, Singapore (1996).

[12] H.-G. Roos, M. Stynes, L. Tobiska, *Numerical Methods for Singularly Perturbed Differential Equations*, Springer-Verlag, Berlin (1996).

[13] G.I. Shishkin, *Methods of constructing grid approximations for singularly perturbed boundary-value problems. Condensing-grid methods*, Russ. J. Numer. Anal. Math. Modelling, 7 (1992) 537–562

[14] G.I. Shishkin, *A difference approximation of a singularly perturbed boundary-value problem for quasilinear elliptic equations degenerating into first-order equations*, Comput. Maths Math. Phys., 32 (1992), 467–480

[15] R. Vulanović, *Non-equidistant finite difference methods for elliptic singular perturbation problems*, in *Computational Methods for Boundary and Interior Layers in Several Dimensions*, (Edited by J.J.H. Miller), 203–223, Boole Press, Dublin (1991).