GIAN Workshop on Theory and Computation of SPDEs, December 2017

Lab 2: Fitted mesh methods for ODEs

Niall Madden (Niall.Madden@NUIGalway.ie) http://www.maths.nuigalway.ie/~niall/TCSPDEs2017

In this lab you will implement a standard central differencing technique for a scalar singularly perturbed reaction-diffusion equation. Using that code, you will verify that analysis from this morning is not sharp. Then you will modify the code to

(a) Correctly choose the mesh transition point, τ_ε.

(b) Modify the method to solve a convection-diffusion problem, and investigate stability issues.

(c) Extend the code to solve coupled systems.

1 The finite difference method for a reaction-diffusion problem

We will start by using a simple finite difference scheme for solving

$$-\varepsilon^2 u'' + b(x)u(x) = f(x) \quad \text{on } (0,1), \tag{1a}$$

and with the boundary conditions

$$u(0) = u(1) = 0. \tag{1b}$$

The scheme we will use can be applied on an arbitrary mesh

$$\bar{\Omega}^N = \{x_0, x_1, \dots, x_N\}.$$

Define $h_i = x_i - x_{i-1}$ and $\overline{h}_i = x_{i+1} - x_{i-1}$. The approximation is $U_0 = 0$, $U_N = 0$, and

$$-\varepsilon^2 \left(\frac{U_{i+1} - U_i}{h_{i+1}} - \frac{U_i - U_{i-1}}{h_i} \right) + \overline{h}_i b(x_i) U_i = \overline{h} f(x_i) \qquad \text{for } i = 1, \dots, N-1.$$

This is implemented as a *MATLAB function file* `Solve_1DRD.m`, which you can download from `http://www.maths.nuigalway.ie/~niall/TCSPDEs2017`
If N is a positive integer, then
`Solve_1DRD(0.1, linspace(0,1,N+1)', @(x)(x+1), @(x)exp(x));`
will return the numerical solution to

$$-\frac{1}{100}u'' + (x+1)u = e^x \qquad \text{on } (0,1),$$

with homogeneous boundary conditions, and solved on a uniform mesh with N intervals.
The test harness `Test_1DRD.m` runs this programme for various values of N and ε. To run it, you'll also need

- `u_true.m`, which stores the true solution to

$$-\varepsilon^2 u'' + u = e^x, \qquad u(0) = u(1) = 0. \tag{2}$$

- `Make_1D_Fitted_Mesh.m`, which can generate a uniform or Shishkin mesh, as required. (Inspect the code to see how it is used).

Running the test harness yields a table of the errors in the numerical solutions to (2). These should be similar to the table shown in Section 2. You should be able to observe that the numerical solution is not satisfactory.

2 The Shishkin mesh

Change Line 38 of `Test_1DRD.m` so that a Shishkin mesh is used. In Section 2 we proved that, if τ_ε is defined as $\min\{1/4, \varepsilon/\beta \ln N\}$, then the method is almost first-order accurate:

$$\|u - U^N\|_{\Omega^N} \le C N^{-1} \ln N,$$

where u is the true solution, and U^N is the solution on a mesh with N intervals[1]. Modify the code so that the rates of convergence are also computed. For example, suppose that $E_N = \|u - U^N\|_{\Omega^N}$ for some fixed ε. Then the rate of convergence could be computed as

$$\rho_N = \log_2(E_N/E_{2N}).$$

From this, you should be able to observe that the method appears to be at least first-order accurate.
Verify that, by setting $\tau_\varepsilon = \min\{1/4, 2\varepsilon/\beta \ln N\}$, the rate of convergence is improved, and that

$$\|u - U^N\|_{\Omega^N} \leq CN^{-2} \ln^2 N.$$

To see that this is sharp, estimate C. Is it independent of N and ε?

3 Convection-diffusion problems

Next we will adapt the code to solve

$$-\varepsilon u'' + a(x)u'(x) = f(x) \quad \text{on } (0,1), \tag{3a}$$

and with the boundary conditions

$$u(0) = u(1) = 0. \tag{3b}$$

First, try solving the problem on a uniform mesh, and with the central difference approximation of $u'(x_i)$:

$$D^0 u_i = \frac{u_{i+1} - u_{i-i}}{x_{i+1} - x_{i-1}}. \tag{4}$$

Incorporate this into the codes above, and, by plotting some numerical solutions, verify that this approach is highly unsatisfactory.
If modifying the code in `Solve_1DRD.m`*, take care to note that rows of the linear system have been scaled by* $\overline{h}_i$*. Also, the leading term in* (3a) *is* ε*, and not* ε^2*.*
Next, repeat this experiment using the first-order backward difference operator, instead of the central difference operator in (4).

$$D^- u_i = \frac{u_i - u_{i-i}}{x_i - x_{i-1}}.$$

Finally, use a piecewise uniform Shishkin mesh for this problem. Note that the optimal Shishkin mesh is a little different for this problem. Since there is only a single layer, near $x = 1$, it requires only two subregions: $[0, 1 - \tau_\varepsilon]$ and $[1 - \tau_\varepsilon, 1]$.

4 Coupled Systems

Extend your code to solve the coupled system:

$$-\begin{pmatrix} \varepsilon & 0 \\ 0 & \mu \end{pmatrix}^2 \boldsymbol{u}'' + B\boldsymbol{u} = \boldsymbol{f}.$$

To start with, take

$$B = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} \qquad \text{and} \qquad \boldsymbol{f} = \begin{pmatrix} 2 - x \\ 1 + e^x \end{pmatrix}$$

We don't have the exact solution for this problem, so use the *double mesh principle*: if U^N is the solution on Ω^N, and $\hat{U}^{2N}$ is the solution obtained by bisecting each interval of Ω^N then we can approximate the error as

$$E_N \approx \max_{i=0,\ldots,N} |U_i^N - \hat{U}_{2i}^{2N}|.$$

5 Looking forward

In Lab 2 we will see how to extend the code to problems in two dimensions, and also how to construct a suitable graded mesh.

[1] Earlier, we just denoted this as U. But now we need to compare errors for different values of N