Chapter 10

MULTIGRID

The idea of systematically using sets of coarser grids to accelerate the convergence of iterative schemes that arise from the numerical solution to partial differential equations was made popular by the work of Brandt. There are many variations of the process and many viewpoints of the underlying theory. The viewpoint presented here is a natural extension of the concepts discussed in Chapter 9.

10.1 Motivation

10.1.1 Eigenvector and Eigenvalue Identification with Space Frequencies

Consider the eigensystem of the model matrix $B(1,-2,1)$. The eigenvalues and eigenvectors are given in Sections 4.3.2 and 4.3.3, respectively. Notice that as the magnitudes of the eigenvalues increase, the space-frequency (or wavenumber) of the corresponding eigenvectors also increase. That is, if the eigenvalues are ordered such that

$$|\lambda_1| \leq |\lambda_2| \leq \cdots \leq |\lambda_M| \tag{10.1}$$

then the corresponding eigenvectors are ordered from low to high space frequencies. This has a rational explanation from the origin of the banded matrix. Note that

$$\frac{\partial^2}{\partial x^2}\sin(mx) = -m^2\sin(mx) \tag{10.2}$$

and recall that

$$\delta_{xx}\vec{\phi} = \frac{1}{\Delta x^2}B(1,-2,1)\vec{\phi} = X\left[\frac{1}{\Delta x^2}D(\vec{\lambda})\right]X^{-1}\vec{\phi} \tag{10.3}$$

where $D(\vec{\lambda})$ is a diagonal matrix containing the eigenvalues. We have seen that $X^{-1}\vec{\phi}$ represents a sine transform, and $X\vec{\phi}$, a sine synthesis. Therefore, the operation $\frac{1}{\Delta x^2}D(\vec{\lambda})$ represents the numerical approximation of the multiplication of the appropriate sine wave by the negative square of its wavenumber, $-m^2$. One finds that

$$\frac{1}{\Delta x^2}\lambda_m = \left(\frac{M+1}{\pi}\right)^2\left[-2+2\cos\left(\frac{m\pi}{M+1}\right)\right] \approx -m^2\,, m << M \tag{10.4}$$

Hence, the correlation of large magnitudes of λ_m with high space-frequencies is to be expected for these particular matrix operators. This is consistent with the physics of diffusion as well. However, this correlation is not necessary in general. In fact, the complete counterexample of the above association is contained in the eigensystem for $B(\frac{1}{2}, 1, \frac{1}{2})$. For this matrix one finds, from Appendix B, exactly the opposite behavior.

10.1.2 Properties of the Iterative Method

The second key motivation for multigrid is the following:

- Many iterative methods reduce error components corresponding to eigenvalues of large amplitude more effectively than those corresponding to eigenvalues of small amplitude.

This is to be expected of an iterative method which is time accurate. It is also true, for example, of the Gauss-Seidel method and, by design, of the Richardson method described in Section 9.5. The classical point-Jacobi method does not share this property. As we saw in Section 9.4.1, this method produces the same value of $|\sigma|$ for λ_{min} and λ_{max}. However, the property can be restored by using $h < 1$, as shown in Fig. 9.2.

When an iterative method with this property is applied to a matrix with the above correlation between the modulus of the eigenvalues and the space frequency of the eigenvectors, error components corresponding to high space frequencies will be reduced more quickly than those corresponding to low space frequencies. This is the key concept underlying the multigrid process.

10.2 The Basic Process

First of all we assume that the difference equations representing the basic partial differential equations are in a form that can be related to a matrix which has certain basic properties. This form can be arrived at "naturally" by simply replacing the derivatives in the PDE with difference schemes, as in the example given by Eq. 3.27, or it can be "contrived" by further conditioning, as in the examples given by Eq. 9.11. The basic assumptions required for our description of the multigrid process are:

1. The problem is linear.

2. The eigenvalues, λ_m, of the matrix are all real and negative.

3. The λ_m are fairly evenly distributed between their maximum and minimum values.

4. The eigenvectors associated with the eigenvalues having largest magnitudes can be correlated with high frequencies on the differencing mesh.

5. The iterative procedure used greatly reduces the amplitudes of the eigenvectors associated with eigenvalues in the range between $\frac{1}{2}|\lambda|_{max}$ and $|\lambda|_{max}$.

These conditions are sufficient to ensure the validity of the process described next.

Having preconditioned (if necessary) the basic finite differencing scheme by a procedure equivalent to the multiplication by a matrix C, we are led to the starting formulation

$$C[A_b\vec{\phi}_\infty - \vec{f}_b] = 0 \tag{10.5}$$

where the matrix formed by the product CA_b has the properties given above. In Eq. 10.5, the vector $\vec{f}_b$ represents the boundary conditions and the forcing function, if any, and $\vec{\phi}_\infty$ is a vector representing the desired exact solution. We start with some initial guess for $\vec{\phi}_\infty$ and proceed through n iterations making use of some iterative process that satisfies property 5 above. We do not attempt to develop an optimum procedure here, but for clarity we suppose that the three-step Richardson method illustrated in Fig. 9.6 is used. At the end of the three steps we find $\vec{r}$, the residual, where

$$\vec{r} = C[A_b\vec{\phi} - \vec{f}_b] \tag{10.6}$$

Recall that the $\vec{\phi}$ used to compute $\vec{r}$ is composed of the exact solution $\vec{\phi}_\infty$ and the error $\vec{e}$ in such a way that

$$A\vec{e} - \vec{r} = 0 \tag{10.7}$$

where

$$A \equiv CA_b \tag{10.8}$$

If one could solve Eq. 10.7 for $\vec{e}$ then

$$\vec{\phi}_\infty = \vec{\phi} - \vec{e} \tag{10.9}$$

Thus our goal now is to solve for $\vec{e}$. We can write the exact solution for $\vec{e}$ in terms of the eigenvectors of A, and the σ eigenvalues of the Richardson process in the form:

$$\vec{e} = \sum_{m=1}^{M/2} c_m \vec{x}_m \prod_{n=1}^{3} [\sigma(\lambda_m h_n)] + \underbrace{\sum_{m=M/2+1}^{M} c_m \vec{x}_m \prod_{n=1}^{3} [\sigma(\lambda_m h_n)]}_{\text{very low amplitude}} \tag{10.10}$$

Combining our basic assumptions, we can be sure that the high frequency content of $\vec{e}$ has been greatly reduced (about 1% or less of its original value in the initial guess). In addition, assumption 4 ensures that the error has been *smoothed.*

Next we construct a permutation matrix which separates a vector into two parts, one containing the odd entries, and the other the even entries of the original vector (or any other appropriate sorting which is consistent with the interpolation approximation to be discussed below). For a 7-point example

$$\begin{bmatrix} e_2 \\ e_4 \\ e_6 \\ e_1 \\ e_3 \\ e_5 \\ e_7 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \end{bmatrix} \quad ; \begin{bmatrix} \vec{e}_e \\ \vec{e}_o \end{bmatrix} = P\vec{e} \tag{10.11}$$

Multiply Eq. 10.7 from the left by P and, since a permutation matrix has an inverse which is its transpose, we can write

$$PA[P^{-1}P]\vec{e} = P\vec{r} \tag{10.12}$$

The operation PAP^{-1} partitions the A matrix to form

$$\begin{bmatrix} A_1 & A_2 \\ & \\ A_3 & A_4 \end{bmatrix} \begin{bmatrix} \vec{e}_e \\ \vec{e}_o \end{bmatrix} = \begin{bmatrix} \vec{r}_e \\ \vec{r}_o \end{bmatrix} \tag{10.13}$$

Notice that

$$A_1\vec{e}_e + A_2\vec{e}_o = \vec{r}_e \tag{10.14}$$

is an exact expression.

At this point we make our one crucial assumption. It is that there is some connection between $\vec{e}_e$ and $\vec{e}_o$ brought about by the smoothing property of the Richardson relaxation procedure. Since the top half of the frequency spectrum has been removed, it is reasonable to suppose that the odd points are the average of the even points. For example

$$\begin{aligned} e_1 &\approx \frac{1}{2}(e_a + e_2) \\ e_3 &\approx \frac{1}{2}(e_2 + e_4) \\ e_5 &\approx \frac{1}{2}(e_4 + e_6) \qquad \text{or } \vec{e}_o = A_2' \vec{e}_e \\ e_7 &\approx \frac{1}{2}(e_6 + e_b) \end{aligned} \tag{10.15}$$

It is important to notice that e_a and e_b represent errors on the boundaries where the error is zero if the boundary conditions are given. It is also important to notice that we are dealing with the relation between $\vec{e}$ and $\vec{r}$ so the original boundary conditions and forcing function (which are contained in $\vec{f}$ in the basic formulation) no longer appear in the problem. Hence, no aliasing of these functions can occur in subsequent steps. Finally, notice that, in this formulation, the averaging of $\vec{e}$ is our only approximation, no operations on $\vec{r}$ are required or justified.

If the boundary conditions are Dirichlet, e_a and e_b are zero, and one can write for the example case

$$A_2' = \frac{1}{2}\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \tag{10.16}$$

With this approximation Eq. 10.14 reduces to

$$A_1\vec{e}_e + A_2A_2'\vec{e}_e \quad = \quad \vec{r}_e \tag{10.17}$$

or

$$A_c\vec{e}_e - \vec{r}_e = 0 \tag{10.18}$$

where

$$A_c = [A_1 + A_2A_2'] \tag{10.19}$$

The form of A_c, the matrix on the coarse mesh, is completely determined by the choice of the permutation matrix and the interpolation approximation. If the original A had been $B(7:1,-2,1)$, our 7-point example would produce

$$PAP^{-1} = \begin{bmatrix} -2 & & & 1 & 1 & & \\ & -2 & & & 1 & 1 & \\ & & -2 & & & 1 & 1 \\ 1 & & & -2 & & & \\ 1 & 1 & & & -2 & & \\ & 1 & 1 & & & -2 & \\ & & 1 & & & & -2 \end{bmatrix} = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix} \tag{10.20}$$

and Eq. 10.18 gives

$$\overbrace{\begin{bmatrix} -2 & & \\ & -2 & \\ & & -2 \end{bmatrix}}^{A_1} + \overbrace{\begin{bmatrix} 1 & 1 & & \\ & 1 & 1 & \\ & & 1 & 1 \end{bmatrix}}^{A_2} \cdot \overbrace{\frac{1}{2}\begin{bmatrix} 1 & & \\ 1 & 1 & \\ & 1 & 1 \\ & & 1 \end{bmatrix}}^{A_2'} = \overbrace{\begin{bmatrix} -1 & 1/2 & \\ 1/2 & -1 & 1/2 \\ & 1/2 & -1 \end{bmatrix}}^{A_c} \tag{10.21}$$

If the boundary conditions are mixed Dirichlet-Neumann, A in the 1-D model equation is $B(1,\vec{b},1)$ where $\vec{b} = [-2,-2,...,-2,-1]^T$. The eigensystem is given by Eq. B.19. It is easy to show that the high space-frequencies still correspond to the eigenvalues with high magnitudes, and, in fact, all of the properties given in Section 10.1 are met. However, the eigenvector structure is different from that given in Eq. 9.55 for Dirichlet conditions. In the present case they are given by

$$x_{jm} = \sin\left[j\left(\frac{(2m-1)\pi}{2M+1}\right)\right] \quad ; \quad m = 1,2,\cdots,M \tag{10.22}$$

and are illustrated in Fig. 10.1. All of them go through zero on the left (Dirichlet) side, and all of them reflect on the right (Neumann) side.

For Neumann conditions, the interpolation formula in Eq. 10.15 must be changed. In the particular case illustrated in Fig. 10.1, e_b is equal to e_M. If Neumann conditions are on the left, $e_a = e_1$. When $e_b = e_M$, the example in Eq. 10.16 changes to

$$A_2' = \frac{1}{2}\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{bmatrix} \tag{10.23}$$

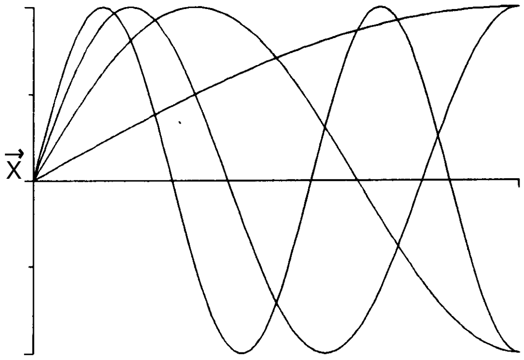

Figure 10.1: Eigenvectors for the mixed Dirichlet–Neumann case.

The permutation matrix remains the same and both A_1 and A_2 in the partitioned matrix PAP^{-1} are unchanged (only A_4 is modified by putting -1 in the lower right element). Therefore, we can construct the coarse matrix from

$$\overbrace{\begin{bmatrix} -2 & & \\ & -2 & \\ & & -2 \end{bmatrix}}^{A_1} + \overbrace{\begin{bmatrix} 1 & 1 & & \\ & 1 & 1 & \\ & & 1 & 1 \end{bmatrix}}^{A_2} \cdot \frac{1}{2} \overbrace{\begin{bmatrix} 1 & & \\ 1 & 1 & \\ & 1 & 1 \\ & & 2 \end{bmatrix}}^{A_2'} = \overbrace{\begin{bmatrix} -1 & 1/2 & \\ 1/2 & -1 & 1/2 \\ & 1/2 & -1/2 \end{bmatrix}}^{A_c} \quad (10.24)$$

which gives us what we might have "expected."

We will continue with Dirichlet boundary conditions for the remainder of this Section. At this stage, we have reduced the problem from $B(1,-2,1)\vec{e} = \vec{r}$ on the fine mesh to $\frac{1}{2}B(1,-2,1)\vec{e}_e = \vec{r}_e$ on the next coarser mesh. Recall that our goal is to solve for $\vec{e}$, which will provide us with the solution $\vec{\phi}_\infty$ using Eq. 10.9. Given $\vec{e}_e$ computed on the coarse grid (possibly using even coarser grids), we can compute $\vec{e}_o$ using Eq. 10.15, and thus $\vec{e}$. In order to complete the process, we must now determine the relationship between $\vec{e}_e$ and $\vec{e}$.

In order to examine this relationship, we need to consider the eigensystems of A and A_c:

$$A = X\Lambda X^{-1}, \qquad A_c = X_c \Lambda_c X_c^{-1} \quad (10.25)$$

For $A = B(M: 1, -2, 1)$ the eigenvalues and eigenvectors are

$$\lambda_m = -2\left[1 - \cos\left(\frac{m\pi}{M+1}\right)\right], \quad \vec{x}_m = \sin\left[j\left(\frac{m\pi}{M+1}\right)\right], \quad \begin{matrix} j = 1, 2, \cdots, M \\ m = 1, 2, \cdots, M \end{matrix} \quad (10.26)$$

Based on our assumptions, the most difficult error mode to eliminate is that with $m = 1$, corresponding to

$$\lambda_1 = -2\left[1 - \cos\left(\frac{\pi}{M+1}\right)\right], \quad \vec{x}_1 = \sin\left[j\left(\frac{\pi}{M+1}\right)\right], \quad j = 1, 2, \cdots, M \tag{10.27}$$

For example, with $M = 51$, $\lambda_1 = -0.003649$. If we restrict our attention to odd M, then $M_c = (M-1)/2$ is the size of A_c. The eigenvalue and eigenvector corresponding to $m = 1$ for the matrix $A_c = \frac{1}{2}B(M_c, 1, -2, 1)$ are

$$(\lambda_c)_1 = -\left[1 - \cos\left(\frac{2\pi}{M+1}\right)\right], (\vec{x}_c)_1 = \sin\left[j\left(\frac{2\pi}{M+1}\right)\right], j = 1, 2, \cdots, M_c \tag{10.28}$$

For $M = 51$ ($M_c = 25$), we obtain $(\lambda_c)_1 = -0.007291 = 1.998\lambda_1$. As M increases, $(\lambda_c)_1$ approaches $2\lambda_1$. In addition, one can easily see that $(\vec{x}_c)_1$ coincides with $\vec{x}_1$ at every second point of the latter vector, that is, it contains the even elements of $\vec{x}_1$.

Now let us consider the case in which all of the error consists of the eigenvector component $\vec{x}_1$, i.e., $\vec{e} = \vec{x}_1$. Then the residual is

$$\vec{r} = A\vec{x}_1 = \lambda_1\vec{x}_1 \tag{10.29}$$

and the residual on the coarse grid is

$$\vec{r}_e = \lambda_1(\vec{x}_c)_1 \tag{10.30}$$

since $(\vec{x}_c)_1$ contains the even elements of $\vec{x}_1$. The exact solution on the coarse grid satisfies

$$\vec{e}_e = A_c^{-1}\vec{r}_e = X_c\Lambda_c^{-1}X_c^{-1}\lambda_1(\vec{x}_c)_1 \tag{10.31}$$

$$= \lambda_1 X_c \Lambda_c^{-1}\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{10.32}$$

$$= \lambda_1 X_c \begin{bmatrix} 1/(\lambda_c)_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{10.33}$$

$$= \frac{\lambda_1}{(\lambda_c)_1}(\vec{x}_c)_1 \tag{10.34}$$

$$\approx \frac{1}{2}(\vec{x}_c)_1 \tag{10.35}$$

Since our goal is to compute $\vec{e} = \vec{x}_1$, in addition to interpolating $\vec{e}_e$ to the fine grid (using Eq. 10.15), we must *multiply the result by 2.* This is equivalent to solving

$$\frac{1}{2}A_c\vec{e}_e = \vec{r}_e \tag{10.36}$$

or

$$\frac{1}{4}B(M_c : 1, -2, 1)\vec{e}_e = \vec{r}_e \tag{10.37}$$

In our case, the matrix $A = B(M : 1, -2, 1)$ comes from a discretization of the diffusion equation, which gives

$$A_b = \frac{\nu}{\Delta x^2}B(M : 1, -2, 1) \tag{10.38}$$

and the preconditioning matrix C is simply

$$C = \frac{\Delta x^2}{\nu}I \tag{10.39}$$

Applying the discretization on the coarse grid with the same preconditioning matrix as used on the fine grid gives, since $\Delta x_c = 2\Delta x$,

$$C\frac{\nu}{\Delta x_c^2}B(M_c : 1, -2, 1) = \frac{\Delta x^2}{\Delta x_c^2}B(M_c : 1, -2, 1) = \frac{1}{4}B(M_c : 1, -2, 1) \tag{10.40}$$

which is precisely the matrix appearing in Eq. 10.37. Thus we see that the process is recursive. The problem to be solved on the coarse grid is *the same as that solved on the fine grid*.

The remaining steps required to complete an entire multigrid process are relatively straightforward, but they vary depending on the problem and the user. The reduction can be, and usually is, carried to even coarser grids before returning to the finest level. However, in each case the appropriate permutation matrix and the interpolation approximation define both the down- and up-going paths. The details of finding optimum techniques are, obviously, quite important but they are not discussed here.

10.3 A Two-Grid Process

We now describe a two-grid process for the linear problem $A\vec{\phi} = \vec{f}$, which can be easily generalized to a process with an arbitrary number of grids due to the recursive nature of multigrid. Extension to nonlinear problems requires that both the solution and the residual be transferred to the coarse grid in a process known as *full approximation storage* multigrid.

1. Perform n_1 iterations of the selected relaxation method on the fine grid, starting with $\vec{\phi} = \vec{\phi}_n$. Call the result $\vec{\phi}^{(1)}$. This gives[1]

$$\vec{\phi}^{(1)} = G_1^{n_1}\vec{\phi}_n + (I - G_1^{n_1})\, A^{-1}\vec{f} \tag{10.41}$$

where

$$G_1 = I + H_1^{-1}A_1 \tag{10.42}$$

and H_1 is defined as in Chapter 9 (e.g., Eq. 9.21). Next compute the residual based on $\vec{\phi}^{(1)}$:

$$\begin{aligned} \vec{r}^{(1)} = A\vec{\phi}^{(1)} - \vec{f} &= AG_1^{n_1}\vec{\phi}_n + A\,(I - G_1^{n_1})\, A^{-1}\vec{f} - \vec{f} \\ &= AG_1^{n_1}\vec{\phi}_n - AG_1^{n_1}A^{-1}\vec{f} \end{aligned} \tag{10.43}$$

2. Transfer (or *restrict*) $\vec{r}^{(1)}$ to the coarse grid:

$$\vec{r}^{(2)} = R_1^2\vec{r}^{(1)} \tag{10.44}$$

In our example in the preceding section, the restriction matrix is

$$R_1^2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{10.45}$$

that is, the first three rows of the permutation matrix P in Eq. 10.11. This type of restriction is known as "simple injection." Some form of weighted restriction can also be used.

3. Solve the problem $A_2\vec{e}^{(2)} = \vec{r}^{(2)}$ on the coarse grid *exactly*:[2]

$$\vec{e}^{(2)} = A_2^{-1}\vec{r}^{(2)} \tag{10.46}$$

[1]See problem 1 of Chapter 9.

[2]Note that the coarse grid matrix denoted A_2 here was denoted A_c in the preceding section.

Here A_2 can be formed by applying the discretization on the coarse grid. In the preceding example (eq. 10.40), $A_2 = \frac{1}{4}B(M_c : 1, -2, 1)$. It is at this stage that the generalization to a multigrid procedure with more than two grids occurs. If this is the coarsest grid in the sequence, solve exactly. Otherwise, apply the two-grid process recursively.

4. Transfer (or *prolong*) the error back to the fine grid and update the solution:

$$\vec{\phi}_{n+1} = \vec{\phi}^{(1)} - I_2^1 \vec{e}^{(2)} \tag{10.47}$$

In our example, the prolongation matrix is

$$I_2^1 = \begin{bmatrix} 1/2 & 0 & 0 \\ 1 & 0 & 0 \\ 1/2 & 1/2 & 0 \\ 0 & 1 & 0 \\ 0 & 1/2 & 1/2 \\ 0 & 0 & 1 \\ 0 & 0 & 1/2 \end{bmatrix} \tag{10.48}$$

which follows from Eq. 10.15.

Combining these steps, one obtains

$$\vec{\phi}_{n+1} = [I - I_2^1 A_2^{-1} R_1^2 A] G_1^{n_1} \vec{\phi}_n - [I - I_2^1 A_2^{-1} R_1^2 A] G_1^{n_1} A^{-1} \vec{f} + A^{-1} \vec{f} \tag{10.49}$$

Thus the basic iteration matrix is

$$[I - I_2^1 A_2^{-1} R_1^2 A] G_1^{n_1} \tag{10.50}$$

The eigenvalues of this matrix determine the convergence rate of the two-grid process.

The basic iteration matrix for a three-grid process is found from Eq. 10.50 by replacing A_2^{-1} with $(I - G_3^2) A_2^{-1}$, where

$$G_3^2 = [I - I_3^2 A_3^{-1} R_2^3 A_2] G_2^{n_2} \tag{10.51}$$

In this expression n_2 is the number of relaxation steps on grid 2, I_3^2 and R_2^3 are the transfer operators between grids 2 and 3, and A_3 is obtained by discretizing on grid 3. Extension to four or more grids proceeds in similar fashion.

10.4 Problems

1. Derive Eq. 10.51.

2. Repeat problem 4 of Chapter 9 using a four-grid multigrid method together with

   (a) the Gauss-Seidel method,

   (b) the 3-step Richardson method derived in Section 9.5.

   Solve exactly on the coarsest grid. Plot the solution after the residual is reduced by 2, 3, and 4 orders of magnitude. Plot the logarithm of the L_2-norm of the residual vs. the number of iterations. Determine the asymptotic convergence rate. Calculate the theoretical asymptotic convergence rate and compare.