Editor: Dianne P. O'Leary, oleary@cs.umd.edu

EIGENVALUES: VALUABLE PRINCIPLES

By Dianne P. O'Leary

IN THIS HOMEWORK ASSIGNMENT, WE'LL STUDY EIGENVALUE PROBLEMS ARISING FROM PARTIAL DIFFERENTIAL EQUATIONS. EIGENVALUES HELP US SOLVE DIFFERENTIAL equations analytically, but they also provide valuable information about a physical system's behavior.

Specifically, we'll focus on eigenvalue properties and use them to design a drum with a particular fundamental frequency of vibration.

What Is an Eigenvalue?

To begin, let's review what we mean by an eigenvalue, or *principal value*, of a matrix. Let

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & -4 \end{bmatrix}.$$

Notice that

$$\boldsymbol{A}\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{A}\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 0 \\ 0 \end{bmatrix},$$

and

$$\boldsymbol{A}\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 3 \\ 0 \end{bmatrix}, \quad \boldsymbol{A}\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -4 \end{bmatrix}.$$

In other words, we've found four vectors—called *eigenvectors* of $\boldsymbol{A}$—that have the special property that multiplication by $\boldsymbol{A}$ just scales the vector.

We call the scale factor the *eigenvalue* of $\boldsymbol{A}$, and we can abbreviate the relation as

$$\boldsymbol{A}\mathbf{x}_j = \lambda_j \mathbf{x}_j,$$

where, in our example, the eigenvalues are $\lambda_1 = -4$, $\lambda_2 = 1$, $\lambda_3 = 2$, and $\lambda_4 = 3$, and the eigenvectors $\mathbf{x}_j$ are the unit vectors.

The *eigensystem* (eigenvalues and eigenvectors) of $\boldsymbol{A}$ has several nice properties. When the eigenvalues are distinct, the eigenvectors are unique—except that they can be multiplied

Tools

Matlab's `pdetool` provides finite element algorithms for solving Problems 2 and 4. You might want to use `initmesh`, `refinemesh`, `pdeeig`, `squareg`, and `squareb1`.

In Problem 4, you're solving a nonlinear equation: find a value of α so that the smallest eigenvalue (which is a function of α) equals a given value. The eigenvalue is a monotonic function of α, increasing as α increases, so you can use a root finder such as Matlab's `fzero`.

Mark Gockenbach gives a good introduction to the eigenvalues of differential operators and the theory of finite difference and finite element methods;[1] for a more advanced treatment, see the work by Stig Larsson and Vidar Thomée.[2]

References

1. M.S. Gockenbach, *Partial Differential Equations: Analytical and Numerical Methods*, SIAM Press, 2002, Section 5.2.
2. S. Larsson and V. Thomée, *Partial Differential Equations with Numerical Methods*, Springer, 2003, Chapter 6.

 Copublished by the IEEE CS and the AIP 1521-9615/05/$20.00 © 2005 IEEE

by any nonzero number. The eigenvectors are *linearly independent*, so they form a basis for $\mathcal{R}^n$. In fact, if A is symmetric, then eigenvectors corresponding to distinct eigenvalues are orthogonal. The smallest eigenvalue λ_1 is the value of the function

$$\min_{\mathbf{x}\neq 0} \frac{\mathbf{x}^T \boldsymbol{A}\mathbf{x}}{\mathbf{x}^T \mathbf{x}},$$

and this value is achieved for $\mathbf{x} = \mathbf{x}_1$. The other eigenvalues can also be characterized as solutions to minimization problems (or maximization problems).

Now, instead of a matrix operating on $\mathcal{R}^n$, let's consider a differential operator. As an example, define $\mathcal{A}u = -u''$ for $x \in \Omega = (0, 1)$, and require that u satisfy the boundary conditions $u(0) = u(1) = 0$. Notice that for $j = 1, 2, \ldots,$

$$\mathcal{A} \sin(j\pi x) = (j\pi)^2 \sin(j\pi x).$$

In other words, we've found functions $w_j(x) = \sin(j\pi x)$—called *eigenfunctions* of $\mathcal{A}$—that satisfy the boundary conditions and have the special property that applying $\mathcal{A}$ just scales the function. We call the scale factor the *eigenvalue* of $\mathcal{A}$, and we can abbreviate the relation as

$$\mathcal{A}w_j = \lambda_j w_j,$$

where

$$\lambda_j = (j\pi)^2.$$

All the properties that we listed for eigenvectors also hold for eigenfunctions. When the eigenvalues are distinct, the eigenfunctions are unique, except that they can be multiplied by any nonzero number. The eigenfunctions are linearly independent, so they form a basis for functions defined on $\mathcal{R}^n$. If $\mathcal{A}$ is self-adjoint, then eigenvectors corresponding to distinct eigenvalues are orthogonal. The smallest eigenvalue solves the minimization problem

$$\min_{w\neq 0} \frac{(w, \mathcal{A}w)}{(w, w)},$$

where

$$(u, v) = \int_\Omega u(x)v(x)dx,$$

and minimization is over all functions w that are zero on the boundary of Ω, and for which the integrals over Ω of $|w|^2$ and $\|\nabla w\|^2$ exist.

Let's find the eigenvalues for a two-dimensional problem.

PROBLEM 1.

Define the domain $\Omega = (0, b) \times (0, b)$. Consider the elliptic partial differential equation

$$-u_{xx} - u_{yy} = \lambda u$$

for $(x, y) \in \Omega$, with $u(x, y) = 0$ on the boundary of Ω.

Show that the function

$$w_{m\ell}(x, y) = \sin(m\pi x/b) \sin(\ell \pi y/b),$$

where m and ℓ are positive integers, satisfies this equation. Determine the corresponding eigenvalue $\lambda_{m\ell}$.

How Can We Compute Approximations to the Eigenvalues?

Suppose we want to compute approximations to the eigenvalues and eigenfunctions of

$$\mathcal{A}u = -\nabla \cdot (a \nabla u)$$

on the domain Ω, with $u = 0$ on the boundary of Ω. Assume that $a > 0$ is a smooth function. In last issue's assignment, we computed an approximate solution to a differential equation by replacing it with a matrix problem. Here, we do the same:

- Replace $\mathcal{A}$ with $\boldsymbol{A}_h$, where $\boldsymbol{A}_h$ is the *finite difference* or *finite element* approximation to $-\nabla \cdot (a \nabla u)$. The parameter h describes the mesh size for the finite difference approximation or the triangle diameter for the finite element approximation.
- Use the eigenvalues $\lambda_{k,h}$ of $\boldsymbol{A}_h$ as approximations to the eigenvalues of $\mathcal{A}$. Because $\mathcal{A}$ has an infinite number of eigenvalues and $\boldsymbol{A}_h$ has finitely many, we can't hope to get good approximations to all eigenvalues of $\mathcal{A}$, but the smallest ones will be well approximated.
- For finite differences, the eigenvectors of $\mathcal{A}_h$ contain approximate values of the eigenfunctions at the mesh points.
- For finite elements, the eigenvectors of $\mathcal{A}_h$ contain coefficients in an expansion of the eigenfunction in the finite element basis.

Suppose Ω is a convex polygon and we use a piecewise linear finite element approximation. Let $\lambda_{k,h}$ be the kth eigenvalue of A_h, and let λ_k be the kth eigenvalue of $\mathcal{A}$ (ascending order). There then exist constants C and h_0, depending on k, such that when h is small enough,

$$\lambda_k \le \lambda_{k,h} \le \lambda_k + Ch^2.$$

PROBLEM 2.

In this problem, we study the elliptic eigenvalue problem $-\nabla \cdot (\nabla u) = \lambda u$ on the square $(-1, 1) \times (-1, 1)$ with zero boundary conditions. We know the true eigenvalues from Problem 1, so we can determine how well the discrete approximation performs.

a. Form a finite difference or finite element approximation to the problem and find the eigenfunctions corresponding to the five smallest eigenvalues:

- Describe in words the shape of each of these eigenfunctions. How does the shape change as the eigenvalue increases?
- Theory tells us that we have good approximations with a coarse grid only for the eigenfunctions corresponding to the smallest eigenvalues. How does the shape of the eigenfunctions make this result easier to understand?

b. Create five plots—for eigenvalues 1, 6, 11, 16, and 21—of the error in the approximate eigenvalue versus $1/h^2$. (Use at least four different matrix sizes, with the finest $h < 1/50$.) Discuss:

- What convergence rate do you observe for each eigenvalue?
- How does it compare with the theoretical convergence rate? (Explain any discrepancy.)
- Are all the eigenvalues well-approximated by coarse meshes?

Some Useful Properties of Eigenvalues

Eigenvalues of elliptic operators have many useful properties. We'll consider two of them in the next problem.

PROBLEM 3.

a. Suppose $\mathcal{A}w = \lambda w$ in Ω, where $\mathcal{A}w = -\nabla \cdot (a \nabla w)$, and $a > 0$. Prove that $\lambda_1 > 0$. Hint: use integration by parts to replace $(w, \mathcal{A}w)$ with $\int_\Omega a(x)\nabla w(x) \cdot \nabla w(x)dx$.

b. Suppose we have two domains $\Omega \subseteq \tilde{\Omega}$. Prove that $\lambda_1(\Omega) \ge \lambda_1(\tilde{\Omega})$. Hint: You can extend the eigenfunction for Ω to be a candidate for the minimization problem for $\tilde{\Omega}$.

How Are Eigenvalues and Eigenfunctions Used?

Eigenvalues and eigenfunctions are useful *mathematical* quantities. If we can compute the eigensystem for a differential operator on a domain Ω analytically, then we can express the solution to the differential equation involving that operator and that domain as a linear combination of the eigenfunctions; determining the coefficients is then relatively simple.

Eigenvalues and eigenfunctions are also useful *physical* quantities. Suppose we model the vibration of a drum with surface Ω through the problem

$$-u_{tt} - c^2 \nabla \cdot (\nabla u) = 0 \text{ in } \Omega.$$

We impose the boundary conditions $u(x, t) = 0$ for x on the boundary of Ω for all $t > 0$, holding the edge of the drum fixed. The eigenvalues λ_j of $\nabla \cdot (\nabla u)$ determine the *characteristic frequencies* of the drum's vibration, and is sometimes called the *fundamental frequency*. If we excite the drum so that it vibrates according to the corresponding $c\sqrt{\lambda_1}/(2\pi)$ eigenfunction, then the vibration will persist.

From Problem 3, we know, for example, that the fundamental frequency of a square drum $\tilde{\Omega}$ of size $a \times a$ is no higher than that of a circular drum Ω of diameter a because $\Omega \subset \tilde{\Omega}$.

PROBLEM 4.

Determine the dimension of a square drum that has a fundamental frequency equal to 1 when $c = 1$. Use numerical methods to find an elliptical domain $\alpha x^2 + 2\alpha y^2 < 1$ with the same fundamental frequency.

You might repeat Problem 4 for domains of different shapes or for different differential operators.

FINITE DIFFERENCES AND FINITE ELEMENTS: GETTING TO KNOW YOU

By Dianne P. O'Leary

IN THIS HOMEWORK, WE EXPLORE THE NUTS AND BOLTS OF FINITE DIFFERENCE AND FINITE ELEMENT APPROXIMATIONS TO A SIMPLE PROBLEM:

$$-(a(x)u'(x))' + c(x)u(x) = f(x) \text{ for } x \in (0, 1),$$

with the functions a, c, and f given and $u(0) = u(1) = 0$. We'll assume that $a(x) \geq a_0$, where a_0 is a positive number, and $c(x) \geq 0$ for $x \in [0, 1]$.

In the finite difference approach, we approximate each derivative of u by a finite difference:

$$u'(x) = \frac{u(x) - u(x-h)}{h} + O(h),$$

$$u''(x) = \frac{u(x-h) - 2u(x) + u(x+h)}{h^2} + O(h^2) \,.$$

PROBLEM 1.

Let $M = 6$, $a(x) = 1$, and $c(x) = 0$, and write the four finite difference equations for u at $x = 0.2, 0.4, 0.6$, and 0.8.

Answer:

$$\frac{1}{h^2}\begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix},$$

where $h = 1/5$, $u_j \approx u(jh)$, and $f_j = f(jh)$.

PROBLEM 2.

The Matlab function `finitediff1.m` on the Web site (www.computer.org/cise/homework/) implements the finite difference method for our equation. The inputs are the parameter M and the functions `a`, `c`, and `f` that define the equation. Each of these functions takes a vector of points as input and returns a vector of function values. (The function `a` also returns a second vector of values of a'.) The outputs of `finitediff1.m` are a vector `ucomp` of computed estimates of u at the mesh points `xmesh`, along with the matrix `A` and the right-hand side `g` from which `ucomp` was computed, so that `A ucomp = g`. Add documentation to the function `finitediff1.m` so that a user could easily use it, understand the method, and modify the function if necessary.

Answer:

Documentation is posted on the Web site for the program `finitediff2.m` of Problem 3, which is very similar to `finitediff1.m` but more useful. If you use a code like `finitediff1.m`, include the name of the code's author, or at least a reference to the Web site from which you obtained it. Your implementation of `finitediff2.m` should probably include a statement like, "Derived from `finitediff1.m` by Dianne O'Leary."

PROBLEM 3.

Define a central difference approximation to the first derivative by

$$u'(x) \approx \frac{u(x+h) - u(x-h)}{2h}.$$

a. Use the Taylor series expansions

$$u(x+h) = u(x) + hu'(x) + \frac{h^2}{2}u''(x) + \frac{h^3}{6}u'''(\xi_1),$$

$$u(x-h) = u(x) - hu'(x) + \frac{h^2}{2}u''(x) - \frac{h^3}{6}u'''(\xi_2),$$

where ξ_1 is some point between x and $x + h$, and ξ_2 is some point between x and $x - h$, to show that the difference between $u'(x)$ and our approximation is $O(h^2)$ if u has a continuous third derivative.

b. Modify the function of Problem 2 to produce a function `finitediff2.m` that uses this approximation in place of the first-order approximation.

Answer:

a. From the given equations, we obtain

$$u(x+h)-u(x-h)=2hu'(x)+\frac{h^3}{6}u'''(\xi_1)-\frac{h^3}{6}u'''(\xi_2) \quad ,$$

so, rearranging and using the mean value theorem from calculus, we get

$$u'(x)-\frac{u(x+h)-u(x-h)}{2h}=-\frac{h^2}{3}u'''(\xi) ,$$

where ξ is some point in the interval $[x-h, x+h]$. We can derive our other finite difference approximations similarly.

b. See `finitediff2.m` on the Web site.

In using piecewise linear finite elements, we can express our approximate solution u_h as

$$u_h(x)=\sum_{j=1}^{M-2} u_j\phi_j(x)$$

for some coefficients u_j, where

$$\phi_j(x)=\begin{cases} \dfrac{x-x_{j-1}}{x_j-x_{j-1}} & x\in[x_{j-1},x_j] \\ \dfrac{x-x_{j+1}}{x_j-x_{j+1}} & x\in[x_j,x_{j+1}] \\ 0 & \text{otherwise.} \end{cases}$$

Then define

$$a(u, v) = \int_0^1 (a(x)u'(x)v'(x) + c(x)u(x)v(x))dx,$$

$$(f, v) = \int_0^1 f(x)v(x)dx.$$

PROBLEM 4.

a. Since the functions ϕ_j form a basis for S_h, any function $v_h \in S_h$ can be written as

$$v_h(x)=\sum_{j=1}^{M-2} v_j\phi_j(x)$$

for some coefficients v_j. Show that if

$$a(u_h, \phi_j) = (f, \phi_j)$$

for $j = 1, \ldots, M - 2$, then

$$a(u_h, v_h) = (f, v_h)$$

for all $v_h \in S_h$.

b. Putting the unknowns u_j in a vector $\mathbf{u}$, we can write the resulting system of equations as $A\mathbf{u} = \mathbf{g}$, where the (j, k) entry in A is $a(\phi_j, \phi_k)$, and the jth entry in $\mathbf{g}$ is (f, ϕ_j). Write this system of equations for $M = 6$, $a(x) = 1$, and $c(x) = 0$, and then compare with your solution to Problem 1.

Answer:

First notice that if α and β are constants and v and z are functions of x, then

$$a(u, \alpha v + \beta z) = \alpha a(u, v) + \beta a(u, z),$$

because we can compute the integral of a sum as the sum of the integrals and then move the constants outside the integrals. Thus,

$$a(u_h,v_h)=a\left(u_h, \sum_{j=1}^{M-2} v_j\phi_j\right)$$

$$=\sum_{j=1}^{M-2} v_j a(u_h,\phi_j)$$

$$=\sum_{j=1}^{M-2} v_j (f,\phi_j)$$

$$=\left(f, \sum_{j=1}^{M-2} v_j\phi_j\right)$$

$$=(f, v_h).$$

b. We compute

$$a(\phi_j,\phi_j)=\int_0^1\left(\phi_j'(x)\right)^2 dx$$

$$=\int_{(j-1)h}^{(j+1)h}\left(\phi_j'(x)\right)^2 dx$$

$$=2\int_{(j-1)h}^{jh}\frac{1}{h^2}dx$$

$$= \frac{2}{h}$$

and

$$a(\phi_j, \phi_{j+1}) = \int_0^1 \phi'_j(x)\phi'_{j+1}(x)dx$$

$$= \int_{jh}^{(j+1)h} \phi'_j(x)\phi'_{j+1}(x)dx$$

$$= \int_{jh}^{(j+1)h} \frac{(-1)}{h}\frac{1}{h}dx$$

$$= -\frac{1}{h}.$$

So our system becomes

$$\frac{1}{h}\begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix},$$

where u_j is the coefficient of ϕ_j in the representation of u_h and

$$f_j = \int_0^1 f(x)\phi_j(x)dx = \int_{(j-1)h}^{(j+1)h} f(x)\phi_j(x)dx,$$

which is h times a weighted average of f over the jth interval. The only difference between the finite difference system and this system is that we've replaced point samples of f with average values. Note that if $a(x)$ isn't constant, then the systems will look even more different.

Problem 5.

Write a function `fe_linear.m` that has the same inputs and outputs as `finitediff1.m` but computes the finite element approximation to the solution using piecewise linear elements. Remember to store A as a sparse matrix.

Answer:
See the code posted on the Web site (www.computer.org/cise/homework/).

Problem 6.

Write a function `fe_quadratic.m` that has the same inputs and outputs as `finitediff1.m` but computes the finite element approximation to the solution using piecewise quadratic elements.

Answer:
See the code posted on the Web site (www.computer.org/cise/homework/).

Problem 7.

Use your four algorithms to solve seven problems. Compute three approximations for each algorithm and each problem, with the number of unknowns in the problem chosen to be 9, 99, and 999. For each approximation, print $\| \mathbf{u}_{computed} - \mathbf{u}_{true} \|_\infty$, where $\mathbf{u}_{true}$ is the vector of true values at the points jz, where z = 1/10, 1/100, or 1/1,000, respectively.

Discuss the results:

- How easy is it to program each of the four methods? Estimate how much work Matlab does to form and solve the linear systems. (The work to solve the tridiagonal systems should be about $5M$ multiplications, and the work to solve the five-diagonal systems should be about $11M$ multiplications, so you just need to estimate the work in forming each system.)
- For each problem, note the observed convergence rate r: if the error drops by a factor of 10^r when M is increased by a factor of 10, then the observed convergence rate is r.
- Explain any deviations from the theoretical convergence rate: $r = 1$ and $r = 2$ for the two finite difference implementations, and $r = 2$ and $r = 3$ for the finite element implementations.

Answer:
Here are the results, in dull tables, but with interesting entries; FD stands for finite difference and FE stands for finite element.

In problem one, we use coefficient functions $a(1)$ and $c(1)$ with true solution $u(1)$; here are infinity norms of the errors at the mesh points for various methods and various numbers of interior mesh points M:

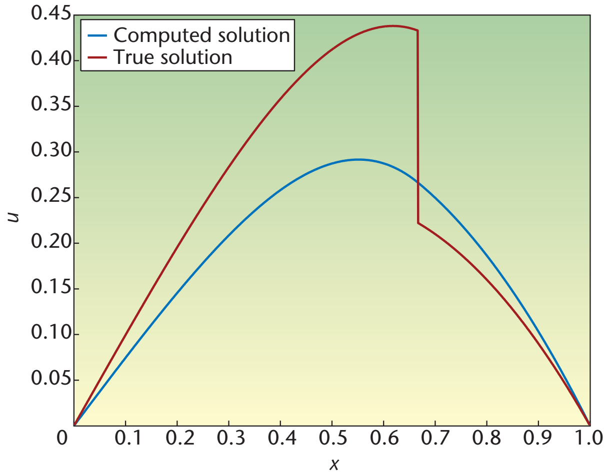

Figure A. The solution to the seventh test problem. We compute an accurate answer to a different problem.

M =	9	99	999
1st order FD	2.1541e-03	2.1662e-05	2.1662e-07
2nd order FD	2.1541e-03	2.1662e-05	2.1662e-07
Linear FE	1.3389e-13	1.4544e-14	1.4033e-13
Quadratic FE	3.1004e-05	3.5682e-09	3.6271e-13

Problem two: coefficient functions $a(1)$ and $c(2)$ with true solution $u(1)$

M =	9	99	999
1st order FD	1.7931e-03	1.8008e-05	1.8009e-07
2nd order FD	1.7931e-03	1.8008e-05	1.8009e-07
Linear FE	6.1283e-04	6.1378e-06	6.1368e-08
Quadratic FE	2.7279e-05	3.5164e-09	1.7416e-12

Problem three: coefficient functions $a(1)$ and $c(3)$ with true solution $u(1)$

M =	9	99	999
1st order FD	1.9405e-03	1.9529e-05	1.9530e-07
2nd order FD	1.9405e-03	1.9529e-05	1.9530e-07
Linear FE	4.3912e-04	4.3908e-06	4.3906e-08
Quadratic FE	2.8745e-05	3.5282e-09	3.6134e-13

Problem four: coefficient functions $a(2)$ and $c(1)$ with true solution $u(1)$

M =	9	99	999
1st order FD	1.5788e-02	1.8705e-03	1.8979e-04
2nd order FD	3.8465e-03	3.8751e-05	3.8752e-07
Linear FE	1.3904e-03	1.3930e-05	1.3930e-07
Quadratic FE	1.6287e-04	1.9539e-08	1.9897e-12

Problem five: coefficient functions $a(3)$ and $c(1)$ with true solution $u(1)$

M =	9	99	999
1st order FD	1.1858e-02	1.4780e-03	1.5065e-04
2nd order FD	3.6018e-03	3.6454e-05	3.6467e-07
Linear FE	8.3148e-04	8.2486e-06	1.2200e-06
Quadratic FE	1.0981e-04	1.6801e-06	2.5858e-06

Problem six: coefficient functions $a(1)$ and $c(1)$ with true solution $u(2)$

M =	9	99	999
1st order FD	8.9200e-02	9.5538e-02	9.6120e-02
2nd order FD	8.9200e-02	9.5538e-02	9.6120e-02
Linear FE	8.6564e-02	9.5219e-02	9.6086e-02
Quadratic FE	8.6570e-02	9.5224e-02	9.6088e-02

Finally, in problem seven, we use coefficient functions $a(1)$ and $c(1)$ with true solution $u(3)$:

M =	9	99	999
1st order FD	1.5702e-01	1.6571e-01	1.6632e-01
2nd order FD	1.5702e-01	1.6571e-01	1.6632e-01
Linear FE	1.4974e-01	1.6472e-01	1.6622e-01
Quadratic FE	1.4975e-01	1.6472e-01	1.6622e-01

Discussion

The bulk of the work in these methods is in function evaluations. We need $O(M)$ evaluations of a, c, and f to form each matrix. For finite differences, the constant is close to 1, but `quad` (the numerical integration routine) uses many function evaluations per call (on the order of 10), making formation of the finite element matrices about 10 times as expensive.

We calculate the experimental rate of convergence as the $\log_{10}$ of the successive errors (because we increase the number of mesh points by a factor of 10 each time). There are several departures from the expected rate of convergence:

- `finitediff1` is expected to have a linear convergence rate ($r = 1$), but has $r = 2$ for the first three problems because $a' = 0$ and the approximation is the same as that in `finitediff2`.
- The quadratic finite element approximation has $r = 4$ on test problems one through four, better than the $r = 3$ we might expect. This is called *superconvergence* and happens because we only measured the error at the mesh points, whereas the $r = 3$ result was for the average value of the error over the entire interval.
- Linear finite elements give almost an exact answer to test

problem one at the mesh points (but not between the mesh points). This occurs because our finite element equations demand that

$$a(u_h, \phi_j) = (u_h', \phi_j') = [-u_h(x_{j-1}) + 2u_h(x_j) - u_h(x_{j+1})]/h = (f, \phi),$$

and our true solution also satisfies this relation.

- In test problem five, the coefficient function a has a discontinuous derivative at $x = 1/3$. The matrix entries that the numerical integration routine computes aren't very accurate, so the finite element methods appear to have slow convergence. This can be fixed by extra calls to `quad` so that it never tries to integrate across the discontinuity.
- The "solution" to test problem six has a discontinuous derivative, and the "solution" to test problem seven is discontinuous. None of our methods compute good approximations, although all of them return a reasonable answer (see Figure A) that could be mistaken for what we're looking for. The finite difference approximations lose accuracy because their error term depends on u''. We derived the finite element equations from the weak formulation of our problem, and when we used integration by parts, we left off the boundary term that we would have gotten at $x = 2/3$, so our equations are wrong. This is a case of, "Be careful what you ask for."
- The entries in the finite element matrices are only approximations to the true values, due to inaccuracy in integral estimation. This means that as the mesh size is decreased, we need to reduce the tolerance we send to `quad` to keep the matrix accurate enough.
- The theoretical convergence rate only holds down to the machine's round-off level, so if we took even finer meshes (much larger M), we would fail to see the expected rate.

On these simple one-dimensional examples, we uncovered many pitfalls in the naive use of finite differences and finite elements. Nevertheless, both methods are quite useful when used with care. CiSE

Dianne P. O'Leary is a professor of computer science and a faculty member in the Institute for Advanced Computer Studies and the Applied Mathematics Program at the University of Maryland. She received a BS in mathematics from Purdue University and a PhD in computer science from Stanford. O'Leary is a member of SIAM, ACM, and AWM. Contact her at oleary@cs.umd.edu; www.cs.umd.edu/users/oleary/.