Final Project

Finite Element Method


Professor: Paul J. Atzberger

Due: Tuesday, December 18, 5:00pm

Instructions: In the final project you will develop a basic finite element method to approximate solutions of parabolic and elliptic PDE's. To solve the finite element equations you will write a multigrid solver. Quadratures will then be used to perform an empirical error analysis of the numerical method. You should present your results in the form of a brief report which addresses the specific questions stated below. For the numerical solutions, you are to give the results in the form of a figure or table as appropriate. All codes should be written in Matlab/Octave. Your codes should be submitted in the form of an appendix to the report and distinctly separated from the discussion of results. If you have questions please do feel free to come to office hours or schedule an appointment for assistance.

Finite Element Method

Part 1: Finite Element Equations

The FEM method approximates solutions of a PDE by reformulating the problem on a finite dimensional function space. We shall consider the case of elliptic PDE's where the problem can be reformulated as the minimization of an appropriately defined functional restricted to a subspace. The key in obtaining a practical finite element method will be to find a convenient description of the subspace which allows for the minimizer to be constructed numerically.

Consider the elliptic PDE:

\[-p\Delta u = f(x), \text{ for } x \in \Omega \]
\[u(x) = g(x), \text{ for } x \in \partial \Omega \]

where \(\Omega\) is a convex domain with Lipschitz continuous boundary, \(f, g\) are given functions, and \(p > 0\) is a constant. The variational principle for the PDE is given by the functional

\[I(v) = \int_\Omega p|\nabla v(x)|^2 dx - 2\int_\Omega f(x)v(x)dx\]

minimized over the space \(H^1_0\).

Let \(T\) denote a triangulation of the domain \(\Omega\) and the space of piecewise polynomials of degree at most \(d\) be denoted by \(S_d(T) = \{v^h(x)|v^h|_K \in \mathcal{P}_d(K), K \in T\}\), where \(\mathcal{P}_d(K)\) denotes the space of all polynomials of degree at most \(d\) on \(K\). To obtain an approximation to the PDE we shall minimize the functional on \(S_d\) subject to continuity constraints. The Ritz finite element method is obtained by parameterizing the finite dimensional function space by a vector \(q\) and computing the minimizer by solving numerically the system of equations \(\frac{\partial I}{\partial q} = 0\).

To simplify the implementation we shall only consider the special case in which the triangulation \(T\) consists of equilateral triangles of the same size and use only an approximating
subspace of $S_1$ (continuous piecewise linear elements). A convenient choice for the parameterization (degrees of freedom) is to use the function values $v_i$ at the nodal points $a_i$, $(v^h(a_i) = v_i)$.

a) For the domains in Figure 1 write codes which triangulate the domains into equilateral triangles of the same size having width at most $h$. This should be done as follows: (i) tile the domain with equilateral triangles of equal size, (ii) refine each of the equilateral triangles into four subtriangles by connecting the midpoints of each edge. Repeat the refinement in stages across the mesh until all the refined triangles have the same size with width less than $h$. The tiling in step (i) is given in Figure 1.

b) Compute the element stiffness matrix for each triangular element. To compute the entries of the matrix use the Gaussian quadrature on the triangle with precision $p = 2$, see Table 1. Be sure to use a sparse matrix data structure to store the entries.

c) Perform the assembly of the global stiffness matrix. Be sure to make use of sparse matrix data structures.

d) Compute an approximate load vector for a given function $f(x)$. This can be done to sufficient accuracy by using the Gaussian quadrature over each triangle of precision $p = 2$, see Table 1.
<table>
<thead>
<tr>
<th>p</th>
<th>n</th>
<th>w</th>
<th>$\lambda_1$</th>
<th>$\lambda_2$</th>
<th>$\lambda_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>0.3333333333333333</td>
<td>0.6666666666666667</td>
<td>0.1666666666666667</td>
<td>0.1666666666666667</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>0.116786275726379</td>
<td>0.501426509658179</td>
<td>0.249286745170910</td>
<td>0.063089014491502</td>
</tr>
<tr>
<td>3</td>
<td>0.050844906370207</td>
<td>0.873821971016996</td>
<td>0.249286745170910</td>
<td>0.063089014491502</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.082851075618374</td>
<td>0.053145049844817</td>
<td>0.310352451033784</td>
<td>0.636502499121399</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Quadrature Nodes and Weights for a Triangular Element: The $p$ denotes the precision of the quadrature, $n$ the number of nodal points associated with each weight, and $w$ the weight value. The $\lambda_k$ denote the barycentric coordinates of each generating nodal point. For the triangle with vertices $x_1, x_2, x_3$ this corresponds to the point $x = \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3$. For each weight $n$ denotes the number of distinct nodal points having this weight. They are obtained by considering the distinct points generated by taking permutations of $(\lambda_1, \lambda_2, \lambda_3)$. The values in the table were obtained from the book *Understanding and Implementing the Finite Element Method* by Mark S. Gockenbach and originate from the paper *High Degree Efficient Symmetrical Gaussian Quadrature Rules for the Triangle*, David Dunavant, International Journal for Numerical Methods in Engineering, Volume 21, 1985, pages 1129-1148.

### Part 2: Multigrid Solver

Large sparse linear systems of equations are obtained from finite element methods. The system size is determined by the number of triangular elements used in approximating the solution of the PDE. Unfortunately, using direct methods such as Gaussian elimination often is very expensive as a consequence non-zero entries being filled in during the elimination procedure. Fortunately, for many finite element methods the linear systems have a lot of structure which can be put to good use.

The sparsity of the finite element equations can be exploited by using iterative methods which operate only on the non-zero entries of the matrix. While this may lead to more efficient solvers than Gaussian elimination, many iterative methods suffer from a stalling behavior for large linear systems giving relatively slow convergence after only a few iterations. One approach to improve the rate of convergence of iterative methods is to use multigrid methods.

In terms of the finite element method this corresponds to introducing a hierarchy of linear equations which represent the finite element solution at different levels of refinement. The multigrid method requires three operations: an interpolation operator $I_{\ell}^{r+1}$ which transmits data from refinement level $\ell$ to a more refined level $r$, a restriction operator $I_{\ell}^{\ell}$ transmits data from a refinement level $r$ to a coarser level $\ell$, and a smoothing operator defined for each refinement level $\ell$ which approximates solutions of the linear system $A^{(\ell)}$.

For our finite element method we shall use linear interpolation over each triangular element for the interpolation operator $I_{\ell}^{\ell+1}$ and its transpose for the restriction operator $I_{\ell}^{\ell+1} = (I_{\ell+1}^{\ell})^T$. To obtain a linear system on refinement level $\ell$ we shall take $A^{(\ell)} = I_{\ell+1}^{\ell} A^{(\ell+1)} I_{\ell+1}^{\ell+1}$, where $A^{(\ell*)} = A$ with $A$ the matrix appearing in the finite element equation at the most refined level. For the smoother we shall use Gauss-Siedel iteration ($Av = b$):

$$v^{(\text{new})}_m = b_m - \sum_{k>m} A_{m,k} v^{(\text{old})}_k - \sum_{k<m} A_{m,k} v^{(\text{new})}_k / A_{m,m}.$$  

(3)

a) For a given triangulation on level $\ell$ implement a routine which computes the sparse matrix representation of the interpolation operator $I_{\ell}^{\ell+1}$. 
b) Given a sparse matrix $A^{(\ell*)}$ for the nodal values on level $\ell*$ write a code which computes the sparse matrices $A^{(\ell)}$ for each of the coarser refinement levels $\ell > \ell*$. Be sure to use sparse matrix multiplication.

c) Write a generic routine which computes the Gauss-Siedel iteration of $A^{(\ell)}$ for a given triangulation at level $\ell$.

d) Show analytically that for the functional $I(v) = v^T A v - 2f^T v$ the the Gauss-Siedel update of a single nodal value $v_i$ corresponds to minimizing $I$ in $v_i$ while holding fixed all other nodal values $v_j, (i \neq j)$.

e) Implement the Full Multigrid Method using your restriction, interpolation, and smoother.

Part 3: Finite Element Solutions We now have a finite element method which can solve the PDE given in equation 1. We shall now consider some basic test problems.

a) Solve the steady-state heat equation 1 for the domain $\Omega_1$ when $p = 1, g(x, y) = (\sqrt{3}y - 2y^2)\cos(5\pi x)$ and $f(x, y) = 25\pi^2(\sqrt{3}y - 2y^2 + (4/25\pi^2))\cos(5\pi x)$. Give plots of the solution for $h = 1/2, h = 1/16, h = 1/32, h = 1/64$.

b) Solve the steady-state heat equation 1 for the domain $\Omega_2$ when $p = 1, g(x, y) = 1 - x^2 - y^2$ and $f(x, y) = 4$. Give a plot of the solution for $h = 1/2, h = 1/16, h = 1/32, h = 1/64$. 
Part 4: Error Analysis Consider the three measures of errors for the finite element method $H^0$-norm, $H^1$-norm and the energy-norm. The errors can be evaluated numerically by using the exact solutions and a sufficiently accurate quadrature which incurs errors significantly smaller than the finite element error being measured. The exact solution on the domain $\Omega_1$ is $v(x, y) = (\sqrt{3}y - 2y^2) \cos(5\pi x)$ and on domain $\Omega_2$ is $v(x, y) = 1 - x^2 - y^2$. The integrals appearing in the expressions for the error can be numerically approximated by Gaussian quadratures.

a) Implement the Gaussian quadrature for the triangular elements of precision $p = 6$ using the nodes and weights given in Table 1.

b) Use the exact solution $v$ on each of the domains to compute the finite element error measured by the $H^0$-norm, $H^1$-norm, energy-norm. Use Gaussian quadrature to evaluate the integrals. Give the empirical rate of convergence in each case in terms of $h^\alpha$.

c) Discuss how the empirical results correspond to the error analysis discussed in lecture.
Part 5: Initial Value Problem (Optional) The above finite element method can also be used in the solution of initial value problems. Consider the parabolic PDE:

\[
\frac{\partial u}{\partial t} = p\Delta u + f(x), \quad \text{for } x \in \Omega \tag{4}
\]

\[u(x, t) = g(x, t), \quad \text{for } x \in \partial \Omega \]

where \(\Omega\) is a convex domain with Lipschitz continuous boundary, \(f, g\) are given functions, and \(p > 0\) is constant. While the finite element method could in principle be reformulated for the full initial value problem on space-time, directly doing so would be computationally expensive since this would require solving for all times simultaneously. Another approach is to use the finite element method only to discretize the equations in space and to use finite differences to discretize in time.

a) Implement a Crank-Nicolson method for the initial value problem (equation 4) using

\[
\frac{v^{(n+1)} - v^{(n)}}{k} = \frac{1}{2}Av^{(n+1)} + \frac{1}{2}Av^{(n)} + f^{(n)}.
\]

where \(v^{(n)}\) denotes the solution at time step \(t_n\), \(k = t_{n+1} - t_n\), and \(A\) denotes the finite element discretization of the Laplacian. This will require each time step solving the equation \((I - \frac{k}{2}A)v^{(n+1)} = r^n\). This can be done using the multigrid solver by replacing \(A\) with the matrix \(I - \frac{k}{2}A\) on the most refined level \(\ell^*\).

b) Compute the solution of the initial value problem (equation 4) for the domain \(\Omega_2\) when \(p = 1, u(x, y, 0) = 1, g(x, t) = 1 - x^2 - y^2\), and \(f(x, t) = 4\). Use at least \(h = 1/16\) and compute the solution to steady-state. Give a plot of the solutions for a few time steps and check the steady-state corresponds to the solution obtained in Part 3.