CS 624: Numerical Solution of Differential Equations
Spring 2006
Problem Set 6

Handed out: Mon., Apr. 24.

Due: Wed., May 3 in lecture.

1. Consider the wave equation $u_{tt} = \triangle u$ defined for spatial coordinates $(x, y) \in \Omega$ where Ω is a polygonal domain in the plane. Consider a standing wave solution given by $u(x, y, t) = \sin(\omega t)v(x, y)$. Assume that u is identically zero on the boundary of Ω.

   (a) Argue that $v(x, y)$ must be a solution to

$$\begin{aligned} \triangle v + \lambda v &= 0 \quad \text{on } \Omega, \\ v &= 0 \quad \text{on } \partial\Omega. \end{aligned}$$

   Here λ is an unknown scalar. How is λ related to ω? This problem is called an "eigenvalue problem."

   (b) Show how to discretize the eigenvalue problem by applying various finite element techniques from lecture such as Green's theorem. Your goal is to transform this PDE to a discrete linear-algebra problem of the following form: Find solutions to $A\mathbf{u} = \lambda M\mathbf{u}$, where A and M are given $n \times n$ matrices, λ is an unknown scalar, and $\mathbf{u}$ is an unknown n-vector. It is not necessary to propose an algorithm for solving the discrete problem; for information about algorithms for $A\mathbf{u} = \lambda M\mathbf{u}$ see Golub & Van Loan, 3rd ed, section 8.7.

   (c) Verify that in your discretized problem, the matrix A you derive is the same as the assembled stiffness matrix for the finite element method for Poisson's equation with Dirichlet boundary conditions. Also, show that the matrix M that you end up with is symmetric, positive definite and sparse. This matrix M is often called the "mass matrix."

2. Let T_0 be the *reference triangle*, that is, the triangle whose vertices are at $(0, 0)$, $(0, 1)$, and $(1, 0)$. Let $u(x, y)$ be the affine linear function on T_0 with nodal values u_1, u_2, u_3 in that order.

   (a) Show that

$$\int\int_{T_0} u(x, y)^2 \, dy \, dx = q(u_1, u_2, u_3)$$

   where

$$q(u_1, u_2, u_3) = \frac{1}{24}((u_1 + u_2)^2 + (u_1 + u_3)^2 + (u_2 + u_3)^2).$$

   Note: it is OK to hand in a printout from a symbolic math package such as the symbolic toolbox in Matlab.

(b) Show that there are constants $c, d > 0$ such that

$$c(u_1^2 + u_2^2 + u_3^2) \leq q(u_1, u_2, u_3) \leq d(u_1^2 + u_2^2 + u_3^2).$$

Hint: The right inequality can be be derived by considering the matrix (operator) 2-norm of the matrix $A = [1, 1, 0; 1, 0, 1; 0, 1, 1]$; the left inequality by considering the 2-norm of A^{-1}. Matlab will find the numerical value of these norms; but you should figure out how they solve the problem.

Note: This question is useful for analyzing the mass matrix M that was obtained in 1(c). More comments about this later.

3. Consider a pure Neumann problem $\triangle u = -f$ on Ω, $\partial u/\partial n = g$ on $\partial\Omega$. Here Ω is a connected polygonal domain in one, two or three dimensions. Consider its discretized variational form $(\text{VAR})_h$, which is as follows. Let T be a triangulation of Ω, and let V_h be the set of all continuous piecewise linear functions (piecewise with respect to T). Then $(\text{VAR})_h$ is: find $u \in V_h$ such that

$$\text{for all } v \in V_h, \int_\Omega \nabla u \cdot \nabla v = \int_\Omega vf + \int_{\partial\Omega} vg.$$

As mentioned in lecture, this problem does not have a unique solution. Suppose the definition of V_h is modified so as to include only those piecewise linear continuous v satisfying

$$\int_\Omega v = 0.$$

(a) Show that with this new definition of V_h, there exists a unique solution to $(\text{VAR})_h$.
(b) In the case of one dimension (so that $\Omega = [a, b]$), find a basis for this modified V_h with the property that for most pairs (i, j), the supports of ϕ_i and ϕ_j are disjoint. Be sure to prove that you have found a basis.

4. Download Shewchuk's Triangle package (winner of the 2003 Wilkinson prize) from the web. Use it to implement a Matlab mesh generator for ellipses. Your routine should have the form

```
[xy,trilist,isbdry] = ellipse_meshgen(a,b,h);
```

where the input arguments a and b are the lengths of the axes of the ellipse (assumed to be centered at the origin with axes parallel to the xy coordinate axes) and h is the desired mesh spacing. The output arguments are `xy`, an $n \times 2$ array of mesh vertices with the (x, y) coordinates of the nodes (one node per row), `trilist`, an $m \times 3$ array of triangles (one triangle per row; entries are integers between 1 and n), and `isbdry`, an $n \times 1$ vector of 0's and 1's in which 1 indicates that the corresponding node lies on the boundary of the ellipse.

There are three challenges to overcome in writing this routine. (1) First, you must download Triangle, compile it, and figure out how it works from the web documentation. (2) You must write the necessary Matlab wrapper routine involving `fopen`, `fscanf`, `fprintf`, `dos` that write out an input file for Triangle, invoke Triangle, and then read back in the resulting mesh. (3) You must figure out how to place roughly evenly spaced nodes at distance $O(h)$ apart from each other on the boundary of an ellipse to make an input file for Triangle. One possible approach for defining boundary nodes is to locate the four points on the boundary of the ellipse where the slope of the tangent is ± 1. Between the top pair and bottom pair, use even $O(h)$ spacing in the x-coordinate, and between the left pair and right pair, use even $O(h)$ spacing in the y-coordinate.

Hand in listings of all m-files and sample runs (i.e., plots of the resulting meshes, which could be plotted by `triplot`).