CS 624: Numerical Solution of Differential Equations
Spring 2004
Problem Set 6

Handed out: Mon., Apr. 26.

Due: Wed., May 5 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. 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 $(\mathrm{VAR})_h$, 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 $(\mathrm{VAR})_h$ 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.

3. Let T be one triangle in a mesh of a region Ω, and let $b(T)$ be the product of the two shorter side-lengths of T. Show that $\|\nabla u - \nabla \tilde{u}\|_{L^\infty(T)}$ can be bounded in terms of $b(T)hM$ (rather than $a(T)hM$ used in lecture). Here u is the solution the (VAR) for some boundary conditions and source term, and $\tilde{u}$ is the piecewise linear interpolant to u.

   Some hints: redo the proof in lecture so that w_2 and w_3 are on the two shorter edges. Let the shorter edges be of length $h' \leq h''$. Note that $h'' \sim \hat{h}$ (why?). On the other hand, it is possible that $h' \ll \hat{h}$. Use two different inequalities (unlike lecture, where the same inequality was used for both) to bound $|\nabla q(x) \cdot (v_3 - v_1)|$ and $|\nabla q(x) \cdot (v_2 - v_1)|$. Work with a matrix R in which one row may be scaled (compared to the R from lecture).

4. Implement a mesh-generator for the unit disk in $\mathbf{R}^2$. The mesh generator should take as a parameter the quantity h, which is the desired mesh size. The output from the vector should be three arrays, `xy` with the position of nodal coordinates, `trilist`, with a list of triangles, and `outeredge`, with a list of edges of the mesh that are on the outer boundary.

   The mesh generator should produce a mesh of 6-noded (quadratic) triangles, and all the triangles in the mesh should be well shaped. The should all have maximum side-length less than h.

   Download the prototype `circle_t6_meshgen` from the course web page, and implement your m-file with that same name according to the comments in the m-file. Note that there is a second argument to your routine, `curveit`.

   Then test your routine using `circle_dirichlet_test`. Give some values of h and the resulting error in the finite element solution. Plot the error both for the curved and uncurved settings, and say something about convergence rate for both the gradient and function for both settings.

   Turn in listings of all m-files that you wrote and a printouts of a few example meshes that test various ranges of a, b, h.

   A hint: The `delaunay` function in Matlab can generate the triangulation as long as you give it well spaced points on the interior and boundary of the unit disk.