Finite Element Methods: Numerical Exercises

Paul J. Atzberger

1. Show that each of the elements have the stated regularity as follows:

   (a) Lagrange triangular element based on $\mathcal{P}_k$ with $k+1$ distinct nodes along each edge is C^0.

   (b) Hermite triangular element based on $\mathcal{P}_3$ is C^0.

   (c) Argyris triangular element based on $\mathcal{P}_5$ is C^1 in the normal direction across edges.

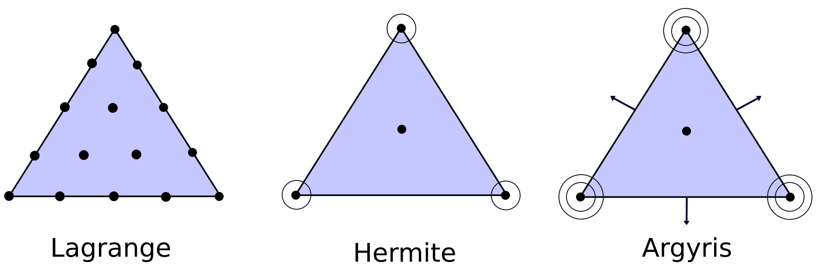

Figure 1: Triangular Elements.

2. There are many ways to develop quadratures for triangulations $\mathcal{T}$ to approximate

$$\int\int_{\mathcal{T}_0} f(\mathbf{x})d\mathbf{x} \approx \sum_k w_k f(\mathbf{x}_k), \quad \mathbf{x} = (x_1, x_2).$$

   (a) Consider Duffy's Transform from a reference triangular element to a quadrilateral element as shown in Figure 2. This is given by

$$\xi = \left(\frac{1+\xi^{'}}{2}\right)\left(\frac{1-\eta'}{2}\right), \quad \eta = \frac{1+\eta^{'}}{2}$$
$$\xi' = \frac{2\xi}{1-\eta} - 1, \quad \eta' = 2\eta - 1,$$

   where $\eta \in [0,1]$, $\xi \in [0, 1-\eta]$, $\xi', \eta' \in [0,1]$. We can express integration over the triangular element as

$$\int_0^1 \int_0^{1-\eta} f(\xi,\eta)d\eta d\xi = \int_{-1}^1 \int_{-1}^1 f(\xi,\eta)J(\xi',\eta')d\eta' d\xi',$$

   where the Jacobian for Duffy's Transform is given by $J(\xi',\eta') = \frac{1}{8}(1-\eta')$.

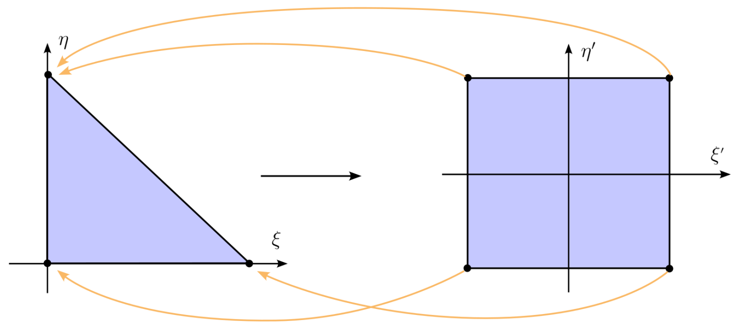

Figure 2: Duffy's Transform.

Use Gaussian quadratures for the cases of 2 and 3 nodes to construct quadratures for the iterated integrals for the quadrilateral. Determine the corresponding nodes and weights for the triangle and construct a quadrature table for the triangular elements for the Gaussian quadrature cases.

(b) Alternatively, we can use for the weights w_k and nodes $\mathbf{x}_k$ from Table 1. For $n = 4, 7$, compare this with the Duffy's Transform approach for the test functions (i) $3x^3y^2$, (ii) $\sin(\pi xy/2)$, and (iii) $\exp(-3x^2 + 3y^2)$. In each case, which yields the more accurate approximation.

d	n	k	$\mathbf{x}_k$	ω_k	k	$\mathbf{x}_k$	w_k	k	$\mathbf{x}_k$	w_k	k	$\mathbf{x}_k$	w_k
1	1	1	(1/3,1/3)	1/2									
2	3	1	(1/6,1/6)	1/6	2	(2/3,1/6)	1/6	3	(1/6,2/3)	1/6			
3	4	1	(1/3,1/3)	-9/32	2	(3/5,1/5)	25/96	3	(1/5,3/5)	25/96	4	(1/5,1/5)	25/96
4	7	1	(0,0)	1/40	2	(1/2,0)	1/15	3	(1,0)	1/40			
		4	(1/2,1/2)	1/15	5	(0,1)	1/40	6	(0,1/2)	1/15	7	(1/3,1/3)	9/40

Table 1: Quadratures on triangulations for $\int_0^1\int_0^{1-x_1} f(\mathbf{x})d\mathbf{x} \approx \sum_k f(\mathbf{x}_k)w_k$, $\mathbf{x} = (x_1, x_2)$. The d is the quadrature order, n number of nodes, $\mathbf{x}_k$ nodes, and ω_k weights. For affine reference element map $\mathbf{x} = \psi(\mathbf{X})$ with $\psi(\mathcal{T}_\ell) = \mathcal{T}_0$ and Jacobian $J(\mathbf{X}) = |\det \partial\psi/\partial\mathbf{X}|$, the quadrature is applied using $\int_{\mathcal{T}_\ell} F(\mathbf{X})d\mathbf{X} = \int_{\mathcal{T}_0} F(\psi^{-1}(\mathbf{x}))J^{-1}d\mathbf{x}$.

3. Consider the elliptic PDE (Poisson problem) given by

$$\Delta u(\mathbf{x}) = -f(\mathbf{x}), \quad \mathbf{x} \in \Omega, \qquad u(\mathbf{x}) = g(x), \quad \mathbf{x} \in \partial\Omega,$$

where $\Omega = [-L, L] \times [-L, L] \subset \mathbb{R}^2$, and $g(x)$ are the boundary values. In the Ritz-Galerkin approximation, we seek a solution $u_h \in \mathcal{V}_h \subset \mathcal{V} = H_0^1(\Omega)$ with

$$a(u_h, w) = -\langle f, w\rangle_{L^2}, \quad \forall w \in \mathcal{V}_h,$$

where $a(u_h, w) = \int_\Omega \nabla_{\mathbf{x}} u_h(\mathbf{x}) \cdot \nabla_{\mathbf{x}} w(\mathbf{x}) d\mathbf{x}$ and $\langle f, w\rangle_{L^2} = \int_\Omega f(\mathbf{x})w(\mathbf{x})d\mathbf{x}$. Consider a basis of functions $\{\phi_k\}_{k=1}^N$ for $\mathcal{V}_h$. We can represent any $v \in V_h$ by $v(\mathbf{x}) = \sum_i v_i\phi_i(\mathbf{x})$,

$u_h(\mathbf{x}) = \sum_i u_i\phi_i(\mathbf{x})$, and approximate f by $f_h(\mathbf{x}) = \sum_i f_i\phi_i(\mathbf{x})$. The FEM approximation u_h can be expressed as solving the linear system

$$A\mathbf{u} = -M\mathbf{f}.$$

The A is the *stiffness matrix* given by $A_{ij} = a(\phi_i, \phi_j)$, M is the *mass matrix* given by $M_{ij} = \langle\phi_i, \phi_j\rangle_{L^2}$, and $[\mathbf{u}]_i = u_i$, $[\mathbf{f}]_i = f_i$.

To handle the Dirichlet boundary conditions we need to use that the boundary values $g(x)$ determine some of the nodal variables. By ordering the nodal indices appropriately, we can split the system into components as $\mathbf{u} = [\mathbf{u}_I, \mathbf{u}_B]$ and $A = [A_I|A_B]$. The $\mathbf{u}_I$ corresponds to the nodal locations interior to the domain Ω and $\mathbf{u}_B$ correspond to the nodal locations on the boundary $\partial\Omega$. Since the values $\mathbf{u}_B$ are known, be sure to move these to the right-hand-side (RHS) of the linear system when solving. By restricting to the rows of the system for the indices of $\mathbf{u}_I$, we obtain the linear system $A_I\mathbf{u}_I = -M\mathbf{f} - A_B\mathbf{u}_B$.

(a) (Meshing) Discretize the domain Ω into elements $\mathcal{T} = \{\mathcal{T}_\ell\}_{\ell=1}^m$, where $\mathcal{T}_\ell$ are triangular elements. For the square domain $\Omega = [-L, L] \times [-L, L] \subset \mathbb{R}^2$, one way to discretize is to define a coarse mesh. A basic algorithm to obtain a more refined discretization is to loop over each triangle and bisect the edges to obtain four smaller triangles, see Figure 3. Data structures for this are a list of vertices $\mathbf{v}_i \in \mathbb{R}^2$ and tuples (i_1, i_2, i_3) which give the indices of the vertices of each triangle.

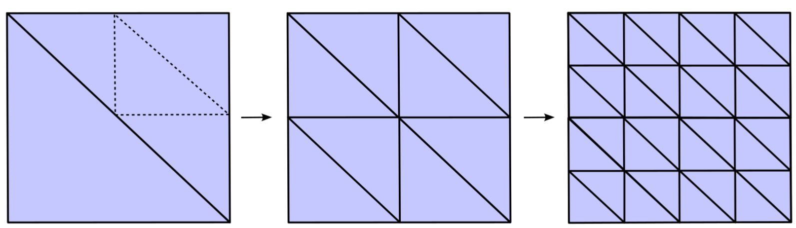

Figure 3: Mesh triangulation and refinement by triangle bisection.

Implement this meshing algorithm for the triangulation in Figure 3. Plot the triangulations when this refinement procedure is done up to $n = 5$ times.

(b) (Assembly and Quadratures) For the discretization into triangular elements $\mathcal{T} = \{\mathcal{T}_\ell\}_{\ell=1}^m$, take $\{\phi_k\}_{k=1}^N$ to be the nodal basis functions for Lagrange elements with polynomial shape functions of degree d so that $v_h|\mathcal{T}_\ell \in \mathcal{P}_d$. The stiffness matrix A is obtained through an assembly procedure where we compute the integral by breaking it into parts summing up the inner-products over each element $\mathcal{T}_\ell$ as $A_{ij} = a(\phi_i, \phi_j) = \sum_{\ell=1}^m \int_{\mathcal{T}_\ell} \nabla_{\mathbf{x}}\phi_i(\mathbf{x}) \cdot \nabla_{\mathbf{x}}\phi_j(\mathbf{x})d\mathbf{x} = \sum_{\ell=1}^m A_{\ell,ij}$, and similarly, $M_{ij} = \langle\phi_i, \phi_j\rangle_{L^2} = \sum_{\ell=1}^m \int_{\mathcal{T}_\ell} \phi_i(\mathbf{x})\phi_j(\mathbf{x})d\mathbf{x} = \sum_{\ell=1}^m M_{\ell,ij}$. Integrals are approximated by high-precision quadratures

$$\tilde{A}_{\ell,ij} = \sum_k \omega_k \nabla_{\mathbf{x}}\phi_i(\mathbf{x}_k) \cdot \nabla_{\mathbf{x}}\phi_j(\mathbf{x}_k), \quad \tilde{M}_{\ell,ij} = \sum_k \omega_k \phi_i(\mathbf{x}_k)\phi_j(\mathbf{x}_k).$$

The $\{\omega_k\}$ are the quadrature weights and $\{\mathbf{x}_k\}$ are the quadrature nodes. Note in general the quadrature nodes can differ from the finite element nodes. We use these approximations to obtain

$$\tilde{A}\mathbf{u} = -\tilde{M}\mathbf{f}.$$

For the case of Lagrange elements using polynomial spaces of degree d, we use quadratures that have order $2d$. This allows for computing the integrals up to round-off errors. For quadratures on triangulations, see Figure 4 and Table 1.

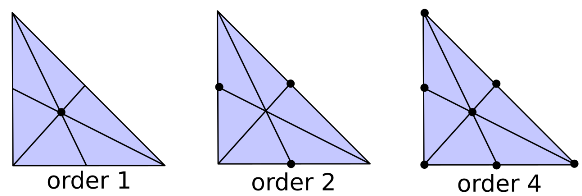

Figure 4: Quadrature Nodes.

Using this assembly + quadrature approach, implement codes to compute for a given triangulation the stiffness and mass matrices when $d = 1$ and $d = 2$.

Consider the FEM approximation for the solutions u with $L = \pi$ and (i) $u(x_1, x_2) = \cos(5x_1)\sin(5x_2)$ and (ii) $u(x_1, x_2) = \exp\left(-\cos(3x_1) + \sin(3x_2)\right)$. Use $f(\mathbf{x}) = -\Delta u$ evaluated at the nodal points to obtain the numerical data for these test problems.

Make a log-log plot of the solution error vs mesh size $h^{-1} = 2^{-n}$ for meshes with refinements $n = 1, 2, \ldots, 5$. What is the exhibited order of accuracy of the Lagrange FEMs when $d = 1$ and $d = 2$?

(c) (Iterative Methods) To solve approximately

$$A\mathbf{u} = \mathbf{b}, \text{ where } \mathbf{b} = -M\mathbf{f},$$

iterative methods can be used of the form

$$B\mathbf{u}^{n+1} = C\mathbf{u}^n + \mathbf{b}.$$

For convergence, $B - C = A$ and the spectral radius of $B^{-1}C$ is taken to satisfy $\rho(B^{-1}C) < 1$. It is common to decompose the matrix as $A = D - L - U$, where D is the diagonal entries, $-L$ the lower entries, and $-U$ the upper entries. A few example iterative methods are

i. Direct Relaxation with $B = I$ and $C = I + \eta A$, with small enough η s.t. $\eta \leq 2/\lambda$ or smaller, where λ is the largest eigenvalue of A.
ii. Jacobi Iteration with $B = D$ and $C = L + U$.
iii. Gauss-Seidel Iteration with $B = D + L$ and $C = U$.

Compare these methods for approximating the solution $\mathbf{u}$ when $L = \pi$ and (i) $u(x_1, x_2) = \cos(5x_1)\sin(5x_2)$ and
(ii) $u(x_1, x_2) = \exp(-\cos(3x_1) + \sin(3x_2))$. Use $f(\mathbf{x}) = -\Delta u$ evaluated at the nodal points to obtain the numerical data for these test problems.

Make a log-log plot of the number iterations and the error for meshes with $n = 5$ refinements. How many iterations does each method need to converge to 1% accuracy for solving the linear system? We remark that in practice these convergence rates are further enhanced by using preconditioners.