# Simluation of Elasticity, Biomechanics, and Virtual Surgery Problem Session I

Joseph Teran (UCLA) Jeffrey Hellrung (UCLA)

July 13, 2010

## 1 Introduction

This problem session will give a brief introduction to the numerical solution of (variable coefficient) Poisson's equation:

$$-\nabla \cdot (\beta \nabla u) = f \quad \in \Omega \tag{1a}$$
$$u = p \quad \in \partial\Omega_d \tag{1b}$$
$$\beta \nabla u \cdot \hat{n} = q \quad \in \partial\Omega_n \tag{1c}$$

Here, $\Omega \subset \mathbb{R}^d$ is open; $\partial\Omega_d$ and $\partial\Omega_n$ partition the boundary of $\Omega$ and define where *Dirichlet* ($d$) and *Neumann* ($n$) conditions are applied, respectively; $\hat{n}$ is the outward-pointing normal on $\partial\Omega_n$; $\beta, f : \Omega \to \mathbb{R}$, $p : \partial\Omega_d \to \mathbb{R}$, and $q : \partial\Omega_q \to \mathbb{R}$ are given, and $\beta$ is bounded below by some positive number; and the goal is to solve for the unknown function $u : \Omega \to \mathbb{R}$. For now, we'll ignore the details of the function spaces that $u$, $\beta$, $f$, $p$, and $q$ are assumed to belong in.

For the numerical solution of eqs (1), we'll use a discretization based on the Finite Element Method (FEM). Briefly, an FEM discretization focuses on the weak formulation of the partial differential equation. The derivation involves multiplying the differential equation by a test function (chosen from some suitable function space), integrating by parts, and applying boundary conditions:

$$-\nabla \cdot (\beta \nabla u) = f \tag{2a}$$
$$\Rightarrow \int_\Omega -\nabla \cdot (\beta \nabla u)\, v = \int_\Omega f v \tag{2b}$$
$$\Rightarrow \int_\Omega \beta \nabla u \cdot \nabla v = \int_\Omega f v + \int_{\partial\Omega} (\beta \nabla u \cdot \hat{n})\, v \tag{2c}$$
$$\Rightarrow \int_\Omega \beta \nabla u \cdot \nabla v = \int_\Omega f v + \int_{\partial\Omega_n} q v \tag{2d}$$

where the derivation from (2c) to (2d) is possible by stipulating that $v \equiv 0$ on $\partial\Omega_d$.

We then suppose that $u$ can be approximated by a function in a finite-dimensional function space (the *(discretized) solution space*) (e.g., piecewise linear functions over some tessellation of $\Omega$), i.e., $u \approx \tilde{u} = \sum_j u_j e_j$ for basis elements $\{e_j\}$. Likewise, $v$ is selected from some other (often related) finite-dimensional function space (the *(discretized) test space*), thus giving a system of equations for the coefficients $u_j$. For example, if $v$ is taken from the same space as $\tilde{u}$, and we let $v$ be, in turn, each of the basis functions, then we obtain

$$\sum_j \left( \int_\Omega \beta \nabla e_j \cdot \nabla e_i \right) u_j = \int_\Omega f e_i + \int_{\partial\Omega_n} q e_i \tag{3}$$

for all $i$, which may be expressed as a linear system of equations:

$$A\vec{x} = \vec{b}, \tag{4}$$

where

$$A_{ij} = \int_\Omega \beta \nabla e_i \cdot \nabla e_j \tag{5a}$$

$$\vec{x}_i = u_i \tag{5b}$$

$$\vec{b}_i = \int_\Omega f e_i + \int_{\partial\Omega_n} q e_i \tag{5c}$$

Strictly speaking, the above is only correct if $v$ varies *precisely* throughout the same function space which $\tilde{u}$ belongs to. This is generally not the case. For example, in the presence of Dirichlet boundary conditions, $\tilde{u}$ will generally be nonzero along $\Omega_d$, however $v$ will be chosen from a function space which vanishes along $\Omega_d$ (hence these functions spaces will only align under homogeneous Dirichlet boundary conditions). How this manifests itself in the linear system will be investigated in the exercises.

Note that the sparsity of $A$ is directly related to the overlap of the supports of the $e_j$'s. Typically we choose the $e_j$'s such that the support of one basis element overlaps the supports of only a small constant number of other basis elements, in which case $A$ will have a number of nonzero entries proportional to the number of basis elements.

Let's now focus on dimension $d = 1$ and use *linear finite elements*. Let $\Omega$ be an interval $(a, b)$, and let $\{x_j\}$ be grid points in $(a, b)$, with $a = x_0 < x_1 < \cdots < x_n = b$. Often times, the $x_j$'s will be equally spaced, which makes the analysis simpler, but it's not necessary. We'll take $\tilde{u}$ (the discretized solution) and $v$ (the discretized test function) to be continuous and *piecewise linear* on each segment $(x_{j-1}, x_j)$, with $v(a)$ and/or $v(b)$ vanishing if a Dirichlet condition is specified at $a$ and/or $b$, respectively. This space of continuous piecewise linear functions is spanned by the *nodal basis functions* (also called "hat" functions, because of the shape of their graphs) $e_j$, (uniquely) defined such that

$$e_j(x_k) = \delta_{jk} = \begin{cases} 1, & j = k \\ 0, & j \neq k \end{cases} \tag{6}$$

# 2 Exercises

Unless otherwise noted, assume 1-dimensional linear finite elements with the basis $\{e_j\}$ given by (6), although these results extend to higher order and higher dimensional finite elements.

1. Here is one way to incorporate the Dirichlet boundary conditions into eqs (5). For concreteness, suppose $\Omega_d = \{a\}$ (with $\Omega = (a, b)$). Then we still have $\tilde{u} = \sum_{j=0}^{n} u_j e_j$, but our space of test functions $\{v\}$ is slightly smaller, $\text{span}\{e_1, \ldots, e_n\}$, to ensure that $v(a) = 0$. Thus we obtain $n$ equations of the form (3), plus one more equation from the Dirichlet boundary condition ($u_0 = u(a)$). This (ostensibly) gives enough equations to solve for the $n + 1$ coefficients $\{u_j\}$.

   Another way to conceptually incorporate the Dirichlet boundary conditions is to consider the function $w = u - u(a)e_0$, which satisfies a similar Poisson problem as $u$ but now with homogeneous Dirichlet boundary conditions. Thus, the space of test functions $\{v\}$ and the solution space of $w$ coincide, and one may use eqs (5) directly.

   In practice, one can actually just ignore the Dirichlet boundary conditions when initially computing the entries of $A$ and $\vec{b}$ (i.e., just use (5a) and (5c)), then "correct" $A$ and $\vec{b}$ to account for the Dirichlet boundary conditions afterwards. Describe a *simple* algorithm to effect this "correction" procedure. Try to keep $A$ symmetric.

2. Show that the linear system (4) has a unique solution if one specifies Dirichlet boundary conditions at either $x = a$ or $x = b$ or both (i.e., $u(a) = u_a$ and/or $u(b) = u_b$). For simplicity, you may assumed $\beta \equiv 1$. What happens if both boundary conditions are Neumann? Specifically address the conditions which $\vec{b}$ must satisfy, how this translates into a condition on $f$, and "how uniquely" $\tilde{u}$ is determined.

3. Compute the integral in (5a) when $\beta \equiv 1$. If you wish, for simplicitly, you may assume the $x_j$'s are equally spaced, i.e., $x_j = a + jh$, where $h = (b - a)/n$. What is the structure of the matrix $A$ (e.g., describe the sparsity)?

4. Give an expression to reasonably approximate the integrals in (5a) and (5c) for arbtrary $\beta$, $f$, and $q$ (assume $\beta$ and $f$ are sufficiently "well-behaved"). You may have to consider boundary and interior grid vertices separately.

5. Write some code (e.g., a Matlab script) to solve the following boundary value problem using linear finite elements:

$$-\left((1+x)\,u'(x)\right)' = (1+x)^{-2} \quad x \in (0,1) \tag{7a}$$

$$u(0) = 1 \tag{7b}$$

$$(1+(1))\,u'(1) = -1/2 \tag{7c}$$

6. The FEM terminology for the matrix $A$ in (4) is the *(global) stiffness matrix* (the meaning of the term "global" will become obvious shortly). With 1-dimensional linear finite elements, it is relatively straightforward to compute the entries of the stiffness matrix, even given that one has to account for differences between boundary and interior grid vertices. However, the situation becomes more complicated with higher order or higher dimensional finite elements, and it becomes tedious to consider all the special boundary cases. As such, it is common practice to assemble the global stiffness matrix element by element (i.e., cell by cell), with each element contributing to several entries of the global stiffness matrix. The collection of these contributions from a single element may be put into a small matrix, the *local stiffness matrix* for that element. The computation of the local stiffness matrix is identical to that of the global stiffness matrix upon replacing the integrations over $\Omega$ with integrations localized over the element. For example, back to 1-dimensional linear finite elements, the local stiffness matrix over $I_j = (x_{j-1}, x_j)$ would consist of $2^2$ (nonzero) entries (since only 2 basis functions are supported on $I_j$). Each of these entries would be added to accumulated global stiffness matrix entries:

$$A_{j-1,j-1} \mathrel{+}= \int_{I_j} \beta \nabla e_{j-1} \cdot \nabla e_{j-1} \tag{8a}$$

$$A_{j-1,j}, A_{j,j-1} \mathrel{+}= \int_{I_j} \beta \nabla e_{j-1} \cdot \nabla e_{j} \tag{8b}$$

$$A_{jj} \mathrel{+}= \int_{I_j} \beta \nabla e_{j} \cdot \nabla e_{j} \tag{8c}$$

This suggests the following procedure to build the (global) stiffness matrix $A$. First, initialize the sparsity pattern of $A$ (with zeros). Then iterate through each element and add the contributions of the element's local stiffness matrix to the corresponding entries in $A$.

Verify this procedure computes the same stiffness matrix as before, and modify your program from Exercise 5 to build the stiffness matrix in this fashion.