

## Lecture 42

# Determining Internal Node Values

As seen in the previous section, a finite element solution of a boundary value problem boils down to finding the best values of the constants  $\{C_j\}_{j=1}^n$ , which are the values of the solution at the nodes. The interior nodes values are determined by *variational principles*. Variational principles usually amount to **minimizing internal energy**. It is a physical principle that systems seek to be in a state of minimal energy and this principle is used to find the internal node values.

### Variational Principles

For the differential equations that describe many physical systems, the internal energy of the system is an integral. For instance, for the steady state heat equation

$$u_{xx}(x, y) + u_{yy}(x, y) = g(x, y) \quad (42.1)$$

the internal energy is the integral

$$I[u] = \iint_R u_x^2(x, y) + u_y^2(x, y) + 2g(x, y)u(x, y) dA, \quad (42.2)$$

where  $R$  is the region on which we are working. It can be shown that  $u(x, u)$  is a solution of (42.1) if and only if it is minimizer of  $I[u]$  in (42.2).

### The finite element solution

Recall that a finite element solution is a linear combination of finite element functions:

$$U(x, y) = \sum_{j=1}^n C_j \Phi_j(x, y),$$

where  $n$  is the number of nodes. To obtain the values at the internal nodes, we will plug  $U(x, y)$  into the energy integral and minimize. That is, we find the minimum of

$$I[U]$$

for all choices of  $\{C_j\}_{j=1}^m$ , where  $m$  is the number of internal nodes. In this as with any other minimization problem, the way to find a possible minimum is to differentiate the quantity with respect to the variables

and set the results to zero. In this case the free variables are  $\{C_j\}_{j=1}^m$ . Thus to find the minimizer we should try to solve

$$\frac{\partial I[U]}{\partial C_j} = 0 \quad \text{for } 1 \leq j \leq m. \quad (42.3)$$

We call this set of equations the **internal node equations**. At this point we should ask whether the internal node equations can be solved, and if so, is the solution actually a minimizer (and not a maximizer). The following two facts answer these questions. These facts make the finite element method practical:

- For most applications the internal node equations are linear.
- For most applications the internal node equations give a minimizer.

We can demonstrate the first fact using an example.

### Application to the steady state heat equation

If we plug the candidate finite element solution  $U(x, y)$  into the energy integral for the heat equation (42.2), we obtain

$$I[U] = \iint_R U_x(x, y)^2 + U_y(x, y)^2 + 2g(x, y)U(x, y) dA. \quad (42.4)$$

Differentiating with respect to  $C_j$  we obtain the internal node equations

$$0 = \iint_R 2U_x \frac{\partial U_x}{\partial C_j} + 2U_y \frac{\partial U_y}{\partial C_j} + 2g(x, y) \frac{\partial U}{\partial C_j} dA \quad \text{for } 1 \leq j \leq m. \quad (42.5)$$

Now we have several simplifications. First note that since

$$U(x, y) = \sum_{j=1}^n C_j \Phi_j(x, y),$$

we have

$$\frac{\partial U}{\partial C_j} = \Phi_j(x, y).$$

Also note that

$$U_x(x, y) = \sum_{j=1}^n C_j \frac{\partial}{\partial x} \Phi_j(x, y),$$

and so

$$\frac{\partial U_x}{\partial C_j} = (\Phi_j)_x.$$

Similarly,  $\frac{\partial U_y}{\partial C_j} = (\Phi_j)_y$ . The integral (42.5) then becomes

$$0 = 2 \iint U_x(\Phi_j)_x + U_y(\Phi_j)_y + g(x, y)\Phi_j(x, y) dA \quad \text{for } 1 \leq j \leq m.$$

Next we use the fact that the region  $R$  is subdivided into triangles  $\{T_i\}_{i=1}^p$  and the functions in question have different definitions on each triangle. The integral then is a sum of the integrals:

$$0 = 2 \sum_{i=1}^p \iint_{T_i} U_x(\Phi_j)_x + U_y(\Phi_j)_y + g(x, y)\Phi_j(x, y) dA \quad \text{for } 1 \leq j \leq m.$$

Now note that the function  $\Phi_j(x, y)$  is linear on triangle  $T_i$  and so

$$\Phi_j(x, y) = \Phi_j|_{T_i}(x, y) = a_{ij} + b_{ij}x + c_{ij}y.$$

This gives us the simplifications

$$(\Phi_j)_x(x, y) = b_{ij} \quad \text{and} \quad (\Phi_j)_y(x, y) = c_{ij}.$$

Also,  $U_x$  and  $U_y$  restricted to  $T_i$  have the form

$$U_x = \sum_{k=1}^n C_k b_{ik} \quad \text{and} \quad U_y = \sum_{k=1}^n C_k c_{ik}.$$

The internal node equations then reduce to

$$0 = \sum_{i=1}^p \iint_{T_i} \left( \sum_{k=1}^n C_k b_{ik} \right) b_{ij} + \left( \sum_{k=1}^n C_k c_{ik} \right) c_{ij} + g(x, y)\Phi_{ij}(x, y) dA \quad \text{for } 1 \leq j \leq m.$$

Now notice that  $(\sum_{k=1}^n C_k b_{ik}) b_{ij}$  is just a constant on  $T_i$ , and, thus, we have

$$\iint_{T_i} \left( \sum_{k=1}^n C_k b_{ik} \right) b_{ij} + \left( \sum_{k=1}^n C_k c_{ik} \right) c_{ij} = \left[ \left( \sum_{k=1}^n C_k b_{ik} \right) b_{ij} + \left( \sum_{k=1}^n C_k c_{ik} \right) c_{ij} \right] A_i,$$

where  $A_i$  is just the area of  $T_i$ . Finally, we apply the Three Corners rule to make an approximation to the integral

$$\iint_{T_i} g(x, y)\Phi_{ij}(x, y) dA.$$

Since  $\Phi_{ij}(x_k, y_k) = 0$  if  $k \neq j$  and even  $\Phi_{ij}(x_j, y_j) = 0$  if  $T_i$  does not have a corner at  $(x_j, y_j)$ , we get the approximation

$$\Phi_{ij}(x_j, y_j)g(x_j, y_j)A_i/3.$$

If  $T_i$  does have a corner at  $(x_j, y_j)$  then  $\Phi_{ij}(x_j, y_j) = 1$ .

Summarizing, the internal node equations are

$$0 = \sum_{i=1}^p \left[ \left( \sum_{k=1}^n C_k b_{ik} \right) b_{ij} + \left( \sum_{k=1}^n C_k c_{ik} \right) c_{ij} + \frac{1}{3}g(x_j, y_j)\Phi_{ij}(x_j, y_j) \right] A_i \quad \text{for } 1 \leq j \leq m.$$

While not pretty, these equations are in fact linear in the unknowns  $\{C_j\}$ .

## Experiment

Download the program `myfiniteelem.m` from the class web site. This program produces a finite element solution for the steady state heat equation without source term:

$$u_{xx} + u_{yy} = 0.$$

To use it, you first need to set up the region and boundary values by running a script such as `mywasher` or `mywedge`. Try different settings for the boundary values `z`. You will see that the program works no matter what you choose.

## Exercises

42.1 Study for the final!

# Review of Part IV

## Methods and Formulas

### Initial Value Problems

#### Reduction to First order system:

For an  $n$ -th order equation that can be solved for the  $n$ -th derivative

$$x^{(n)} = f\left(t, x, \dot{x}, \ddot{x}, \dots, \frac{d^{n-1}x}{dt^{n-1}}\right) \quad (42.6)$$

use the standard change of variables:

$$\begin{aligned} y_1 &= x \\ y_2 &= \dot{x} \\ &\vdots \\ y_n &= x^{(n-1)} = \frac{d^{n-1}x}{dt^{n-1}}. \end{aligned} \quad (42.7)$$

Differentiating results in a first-order system:

$$\begin{aligned} \dot{y}_1 &= \dot{x} = y_2 \\ \dot{y}_2 &= \ddot{x} = y_3 \\ &\vdots \\ \dot{y}_n &= x^{(n)} = f(t, y_1, y_2, \dots, y_n). \end{aligned} \quad (42.8)$$

#### Euler's method:

$$\mathbf{y}_{i+1} = \mathbf{y}_i + hf(t_i, \mathbf{y}_i).$$

#### Modified (or Improved) Euler method:

$$\begin{aligned} \mathbf{k}_1 &= hf(t_i, \mathbf{y}_i) \\ \mathbf{k}_2 &= hf(t_i + h, \mathbf{y}_i + \mathbf{k}_1) \\ \mathbf{y}_{i+1} &= \mathbf{y}_i + \frac{1}{2}(\mathbf{k}_1 + \mathbf{k}_2) \end{aligned}$$

## Boundary Value Problems

### Finite Differences:

Replace the Differential Equation by Difference Equations on a grid.  
Review the lecture on Numerical Differentiation.

### Explicit Method Finite Differences for Parabolic PDE (heat):

$$u_t \mapsto \frac{u_{i,j+1} - u_{ij}}{k} \quad \text{and} \quad u_{xx} \mapsto \frac{u_{i-1,j} - 2u_{ij} + u_{i+1,j}}{h^2} \quad (42.9)$$

leads to

$$u_{i,j+1} = ru_{i-1,j} + (1 - 2r)u_{i,j} + ru_{i+1,j},$$

where  $h = L/m$ ,  $k = T/n$ , and  $r = ck/h^2$ . The stability condition is  $r < 1/2$ .

### Implicit Method Finite Differences for Parabolic PDE (heat):

$$u_t \mapsto \frac{u_{i,j+1} - u_{ij}}{k} \quad \text{and} \quad u_{xx} \mapsto \frac{u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}}{h^2} \quad (42.10)$$

leads to

$$u_{i,j} = -ru_{i-1,j+1} + (1 + 2r)u_{i,j+1} - ru_{i+1,j+1},$$

which is always stable and has truncation error  $O(h^2 + k)$ .

### Crank-Nicholson Method Finite Differences for Parabolic PDE (heat):

$$-ru_{i-1,j+1} + 2(1 + r)u_{i,j+1} - ru_{i+1,j+1} = ru_{i-1,j} + 2(1 - r)u_{i,j} + ru_{i+1,j},$$

which is always stable and has truncation error  $O(h^2 + k^2)$ .

### Finite Difference Method for Elliptic PDEs:

$$u_{xx} + u_{yy} = f(x, y) \mapsto \frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{h^2} + \frac{u_{i,j+1} - 2u_{ij} + u_{i,j-1}}{k^2} = f(x_i, y_j) = f_{ij},$$

### Finite Elements:

Based on triangles instead of rectangles.

Can be used for irregularly shaped objects.

An element: Pyramid shaped function at a node.

A finite element solution is a linear combination of finite element functions:

$$U(x, y) = \sum_{j=1}^n C_j \Phi_j(x, y),$$

where  $n$  is the number of nodes, and where  $U$  is an approximation of the true solution.

$C_j$  is the value of the solution at node  $j$ .

$C_j$  at the boundary nodes are given by boundary conditions.

$C_j$  at interior nodes are determined by variation principles.

The last step in determining  $C_j$ 's is solving a linear system of equations.

## MATLAB

Initial value problem solver that uses the Runge-Kutta 45 method, which has error  $O(h^5)$ . The input  $y0$  is the initial vector and  $tspan$  is the time span. You can either make  $f$  a vector valued anonymous function and do

```
df = @(t,y)[-y(2);y(1)]
[T Y ] = ode45(dy,tspan,y0)
```

or make a function program that outputs a vector

```
function dy = myf(t,y)
    dy = [-y(2);y(1)];
end
```

and then do

```
[T Y ] = ode45(@myf,tspan,y0)
```

The program `ode45` and other MATLAB IVP solvers use adaptive step size to achieve a desired local and global accuracy, with a default of `tol = 10-6` for the global error.

The chief benefit of higher order methods and variable step size is that they allow a program to take only as few steps as necessary.