Systems of Partial Differential Equations

Computer Session B4

It is very rare that a real life phenomenon can be modeled by a single partial differential equation. Usually it takes a system of coupled partial differential equations to yield a complete model. For example, let us say that we want to compute the distribution of heat within a microwave oven. Then we must first compute the electrical wave E that generates the heat. It is given by the Helmholtz equation $\Delta E + \omega^2 E = 0$, where ω is the frequency of the wave. Second, we must solve the heat equation $-\Delta T = |E|^2$ for the temperature T within the oven. Since T depends on E this is a coupled problem with two partial differential equations. In this computer session we study finite element approximations of such problems.

Instructions

- ✔ To pass this session you must present your solutions to all problems with this mark to the instructor.
- ✰ This mark indicates that the problem is included in the course's problem demonstration sessions (gives bonus points).

Model Problem

We start by considering the model problem of finding u_1 and u_2 such that

$$
\begin{aligned}
-\Delta u_1 + c_{11}u_1 + c_{12}u_2 &= f_1, \quad \text{in } \Omega && \text{(1a)}\\
-\Delta u_2 + c_{21}u_1 + c_{22}u_2 &= f_2, \quad \text{in } \Omega && \text{(1b)}\\
n \cdot \nabla u_1 &= 0, \quad \text{on } \partial\Omega && \text{(1c)}\\
n \cdot \nabla u_2 &= 0, \quad \text{on } \partial\Omega && \text{(1d)}
\end{aligned}
$$

where $c_{ij} > 0$, $i, j = 1, 2$, and f_i, $i = 1, 2$, are given coefficients. As usual, $\Omega \subset \mathbb{R}^2$ is assumed to be a domain with smooth boundary $\partial\Omega$ and outward unit normal n.

Variational Formulation

Let

$$V = H^1(\Omega) = \{v : \|\nabla v\| + \|v\| \leq \infty\} \tag{2}$$

with norm $\|v\|_V = (\|\nabla v\|^2 + \|v\|^2)^{1/2}$.

Multiplying (1a) by a test function $v_1 \in V$ and using partial integration we have

$$\begin{aligned}
(f_1, v_1) &= (-\Delta u_1, v_1) + (c_{11}u_1 + c_{12}u_2, v_1) && (3)\\
&= -(n \cdot \nabla u_1, v_1)_{\partial\Omega} + (\nabla u_1, \nabla v_1) + (c_{11}u_1 + c_{12}u_2, v_1) && (4)\\
&= (\nabla u_1, \nabla v_1) + (c_{11}u_1 + c_{12}u_2, v_1) && (5)
\end{aligned}$$

where the boundary term $(n \cdot \nabla u_1, v_1)$ vanish due to the boundary condition. Similarly, multiplying (1b) by another test function $v_2 \in V$ and integrating by parts yields

$$(f_2, v_2) = (\nabla u_2, \nabla v_2) + (c_{21}u_1 + c_{22}u_2, v_2) \tag{6}$$

Adding (5) and (6) give us the variational equation

$$\begin{aligned}
(f_1, v_1) + (f_2, v_2) = (\nabla u_1, \nabla v_1) &+ (c_{11}u_1 + c_{12}u_2, v_1) \\
+ (\nabla u_2, \nabla v_2) &+ (c_{21}u_1 + c_{22}u_2, v_2)
\end{aligned} \tag{7}$$

We shall now rewrite this using vector notation. To this end we introduce the vectors

$$\boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \qquad \boldsymbol{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{8}$$

We also need the gradient matrix for these vectors, defined by

$$\nabla \boldsymbol{v} = \begin{bmatrix} \partial v_1/\partial x_1 & \partial v_1/\partial x_2 \\ \partial v_2/\partial x_1 & \partial v_2/\partial x_2 \end{bmatrix} \tag{9}$$

With this definitions we can write

$$(\nabla u_1, \nabla v_1) + (\nabla u_2, \nabla v_2) = \sum_{i,j=1}^{2} (\partial u_i/\partial x_j, \partial v_i/\partial x_j) \equiv (\nabla \boldsymbol{u} : \nabla \boldsymbol{v}) \tag{10}$$

where we have introduced the colon operator : between two 2×2 matrices $\boldsymbol{A}$ and $\boldsymbol{B}$

$$\boldsymbol{A} : \boldsymbol{B} = \sum_{i,j=1}^{2} a_{ij} b_{ij} \tag{11}$$

Further, collecting the coefficients c_{ij} into a matrix

$$\boldsymbol{C} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \tag{12}$$

we can write the terms

$$(c_{11}u_1 + c_{12}u_2, v_1) + (c_{21}u_1 + c_{22}u_2, v_2) = (\boldsymbol{C}\boldsymbol{u}, \boldsymbol{v}) \tag{13}$$

Finally, we write

$$(f_1, v_1) + (f_2, v_2) = (\boldsymbol{f}, \boldsymbol{v}) \tag{14}$$

Using vector notation the variation formulation of (1) reads: find $\boldsymbol{u} \in \boldsymbol{V} = V \times V$ such that

$$a(\boldsymbol{u}, \boldsymbol{v}) = l(\boldsymbol{v}), \quad \forall \boldsymbol{v} \in \boldsymbol{V} \tag{15}$$

where the bilinear form $a(\cdot, \cdot)$ and the linear functional $l(\cdot)$ are defined by

$$a(\boldsymbol{u}, \boldsymbol{v}) = (\nabla \boldsymbol{u} : \nabla \boldsymbol{v}) + (\boldsymbol{C}\boldsymbol{u}, \boldsymbol{v}) \tag{16}$$

$$l(\boldsymbol{v}) = (\boldsymbol{f}, \boldsymbol{v}) \tag{17}$$

Problem 1. ✰ Write out the component form of $\nabla(\nabla \cdot \boldsymbol{u}) + \Delta \boldsymbol{u} = \boldsymbol{0}$.

Problem 2. ✰ Make a variational formulation of the system $-\Delta u_1 = u_2$, $-\Delta u_2 = f$ with $u_1 = 0$ and $u_2 = 0$ on the boundary. *Hint:* You do not have to use vector notation.

Problem 3. ✰ Verify that (15) satisfies the requirements for the Lax-Milgram lemma on the space $\boldsymbol{V} = V \times V$ with norm $\|\boldsymbol{v}\|_{\boldsymbol{V}} = (\|v_1\|_V^2 + \|v_2\|_V^2)^{1/2}$. *Hint:* What assumption do the coefficients c_{ij} need to fulfill?

Finite Element Approximation

Let $\mathcal{K} = \{K\}$ be a mesh of Ω into shape regular triangles K, and let $V_h \subset V$ be the space of all continuous piecewise linear functions on $\mathcal{K}$ that vanish on the boundary. The finite element approximation of (15) takes the form: find $\boldsymbol{U} \in \boldsymbol{V}_h = V_h \times V_h$ such that

$$a(\boldsymbol{U}, \boldsymbol{v}) = l(\boldsymbol{v}), \quad \forall \boldsymbol{v} \in \boldsymbol{V}_h \tag{18}$$

Derivation of the Discrete System of Equation

Let $\{\varphi_i\}_{i=1}^{N}$ be the usual basis of hat functions for V_h. A basis for $\boldsymbol{V}_h = V_h \times V_h$ is given by

$$\left\{ \begin{bmatrix} \varphi_1 \\ 0 \end{bmatrix}, \begin{bmatrix} \varphi_2 \\ 0 \end{bmatrix}, \dots, \begin{bmatrix} \varphi_N \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \varphi_1 \end{bmatrix}, \begin{bmatrix} 0 \\ \varphi_2 \end{bmatrix}, \dots, \begin{bmatrix} 0 \\ \varphi_N \end{bmatrix} \right\} = \{\boldsymbol{\varphi}_i\}_{i=1}^{2N} \tag{19}$$

Using this the finite element solution $\boldsymbol{U} = [U_1,\ U_2]$ can be written either as

$$\boldsymbol{U} = \sum_{j=1}^{2N} \xi_j \boldsymbol{\varphi}_j \tag{20}$$

using vector notation, or

$$U_1 = \sum_{j=1}^{N} \eta_j \varphi_j, \qquad U_2 = \sum_{j=1}^{N} \zeta_j \varphi_j, \tag{21}$$

using component form.

The finite element method (15) is equivalent to

$$a(\boldsymbol{U}, \boldsymbol{\varphi}_i) = l(\boldsymbol{\varphi}_i), \quad i = 1, \dots, 2N \tag{22}$$

Inserting $\boldsymbol{U} = \sum_{j=1}^{2N} \xi_j \boldsymbol{\varphi}_j$ into (22) gives

$$b_i = l(\boldsymbol{\varphi}_i) = \sum_{j=1}^{2N} \xi_j a(\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i) = \sum_{i,j=1}^{2N} A_{ij} \xi_j, \quad i = 1, \dots, 2N \tag{23}$$

where we have introduced the notation

$$A_{ij} = a(\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_j), \quad i, j = 1, \dots, 2N \tag{24}$$

$$b_i = l(\boldsymbol{\varphi}_i), \quad i = 1, \dots, 2N \tag{25}$$

This is just a $2N \times 2N$ linear system

$$\boldsymbol{A}\boldsymbol{\xi} = \boldsymbol{b} \tag{26}$$

where the entries of the matrix $\boldsymbol{A}$, and the vector $\boldsymbol{b}$ are defined by (24) and (25), respectively. The vector $\boldsymbol{\xi}$ contains the nodal values of the finite element solution $\boldsymbol{U}$ and takes the form

$$\boldsymbol{\xi} = [\xi_1, \dots, \xi_{2N}]^T = [\eta_1, \dots, \eta_N,\ \zeta_1, \dots, \zeta_N]^T \tag{27}$$

The ordering of the hat functions in the construction of the basis for $\boldsymbol{V}_h$ leads to a block structure of the matrix $\boldsymbol{A}$

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{K} + \boldsymbol{M}^{(c_{11})} & \boldsymbol{M}^{(c_{12})} \\ \boldsymbol{M}^{(c_{21})} & \boldsymbol{K} + \boldsymbol{M}^{(c_{22})} \end{bmatrix} \tag{28}$$

where $\boldsymbol{K}$ and $\boldsymbol{M}^{(c)}$ are the $N \times N$ stiffness and mass matrix with entries

$$K_{ij} = (\nabla\varphi_j, \nabla\varphi_i), \quad i, j = 1, \dots, N \tag{29}$$

$$M_{ij}^{(c)} = (c\varphi_j, \varphi_i), \quad i, j = 1, \dots, N \tag{30}$$

A similar block structure applies to the vector $\boldsymbol{b}$, which takes the form

$$\boldsymbol{b} = \begin{bmatrix} \boldsymbol{F}^{(f_1)} \\ \boldsymbol{F}^{(f_2)} \end{bmatrix} \tag{31}$$

where

$$F_i^{(f)} = (f, \varphi_i), \quad i = 1, \dots, N \tag{32}$$

Problem 4. ✔ How would the vector (31) change if we defined the basis for $\boldsymbol{V}_h$ by

$$\left\{ \begin{bmatrix} \varphi_1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \varphi_1 \end{bmatrix}, \begin{bmatrix} \varphi_2 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \varphi_2 \end{bmatrix}, \dots, \begin{bmatrix} \varphi_N \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \varphi_N \end{bmatrix} \right\} \tag{33}$$

Matlab Implementation

Using the build-in assembly routine `assema` it is very easy to assemble the linear system (26) and compute the finite element solution $\boldsymbol{U}$. We list the code below.

```
[p,e,t]=initmesh(geom,'hmax',0.1);
N=size(p,2);
% find triangle midpoints
i=t(1,:); j=t(2,:); k=t(3,:);
x=(p(1,i)+p(1,j)+p(1,k))/3;
y=(p(2,i)+p(2,j)+p(2,k))/3;
% evaluate coefficients and assemble
[K,Mc11,Ff1]=assema(p,t,1,c11(x,y),f1(x,y));
[K,Mc22,Ff2]=assema(p,t,1,c22(x,y),f2(x,y));
[unused,Mc12,unused]=assema(p,t,0,c12(x,y),0);
[unused,Mc21,unused]=assema(p,t,0,c21(x,y),0);
A=[K+Mc11 Mc12; Mc21 K+Mc22];
b=[Ff1; Ff2]
% solve linear system
xi=A\b;
% visualize solution
eta=xi(1:N); zeta=xi(N+1:end);
figure(1), pdesurf(p,t,eta)
figure(2), pdesurf(p,t,zeta)
```

Here, `c11`, `c12`, etc., are subroutines defining the coefficients c_{11}, c_{12} etc. For example,

```
function z=c11(x,y)
z=x+1;
```

Problem 5. ✔ Implement the code outlined above and solve the system (1) on a domain of your choice with $c_{11} = c_{22} = 1$, $c_{12} = c_{21} = 0$, and $f_1 = \sin(x_1)$ and $f_2 = \sin(x_2)$. Repeat with $c_{22} = 10$ and $c_{21} = 1$.

Problem 6. ✔ Try to solve the system (1) on a domain of your choice with $c_{11} = c_{12} = 0$, $c_{21} = c_{22} = 1$, and $f_1 = f_2 = 1$. What is the problem?

Extension to Time-Dependent Problems

We next extend the discussion to the time-dependent problem

$$
\begin{aligned}
\dot{u}_1 - \Delta u_1 + c_{11}u_1 + c_{12}u_2 &= f_1, \quad \text{in } \Omega \times I && \text{(34a)}\\
\dot{u}_2 - \Delta u_2 + c_{21}u_1 + c_{22}u_2 &= f_2, \quad \text{in } \Omega \times I && \text{(34b)}\\
n \cdot \nabla u_1 &= 0, \quad \text{on } \partial\Omega \times I && \text{(34c)}\\
n \cdot \nabla u_2 &= 0, \quad \text{on } \partial\Omega \times I && \text{(34d)}\\
u_1(\cdot, 0) &= u_1^0, \quad \text{in } \Omega && \text{(34e)}\\
u_2(\cdot, 0) &= u_2^0, \quad \text{in } \Omega && \text{(34f)}
\end{aligned}
$$

where the dot superscript means differentiation with respect to time t and $I = (0, T]$ is the time interval with final time T. Moreover, u_1^0 and u_2^0 denotes two given initial conditions.

To obtain a numerical method we shall first apply finite elements in space. This will lead to a system of ordinary differential equations in time, which we subsequently solve using the Euler backward time stepping scheme.

A variational formulation of (34) in space reads: find $\boldsymbol{u} \in \boldsymbol{V}$ such that for every fixed t

$$
(\dot{\boldsymbol{u}}, \boldsymbol{v}) + a(\boldsymbol{u}, \boldsymbol{v}) = l(\boldsymbol{v}), \quad \forall \boldsymbol{v} \in \boldsymbol{V},\ t \in I \tag{35}
$$

where $a(\cdot,\cdot)$ and $l(\cdot)$ are defined by (16) and (17), respectively. The corresponding finite element approximation in space takes the form: find $\boldsymbol{U} \in \boldsymbol{V}_h$ such that for every fixed t

$$
(\dot{\boldsymbol{U}}, \boldsymbol{v}) + a(\boldsymbol{U}, \boldsymbol{v}) = l(\boldsymbol{v}), \quad \forall \boldsymbol{v} \in \boldsymbol{V}_h,\ t \in I \tag{36}
$$

An ansatz for $\boldsymbol{U}$ is given by

$$
\boldsymbol{U} = \sum_{i=1}^{2N} \xi_i(t) \boldsymbol{\varphi}_i \tag{37}
$$

where $\boldsymbol{\varphi}_i$ are the vector valued hat basis functions of (19). Comparing with (20) we see that the big difference between the construction of $\boldsymbol{U}$ for time-dependent and time-independent problems are the coefficients ξ_j. For time-dependent problems $\xi_j = \xi_j(t)$ are functions of time t, whereas they are constants for time-independent problems.

Substituting the ansatz into (36) with $\boldsymbol{v} = \boldsymbol{\varphi}_i$ we get

$$
\begin{aligned}
b_i &= l(\boldsymbol{\varphi}_i) && (38)\\
&= \sum_{j=1}^{2N} \dot{\xi}_j(t)(\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i) + \xi_j(t)a(\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i) && (39)\\
&= \sum_{i,j=1}^{2N} M_{ij}\dot{\xi}_j(t) + A_{ij}\xi_j(t), \quad i = 1, \dots, 2N && (40)
\end{aligned}
$$

where we have introduced the notation

$$M_{ij} = (\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_j), \quad i, j = 1, \dots, 2N \tag{41}$$

This is a system of $2N$ ordinary differential equations. In matrix form we write

$$\boldsymbol{M}\dot{\boldsymbol{\xi}}(t) + \boldsymbol{A}\boldsymbol{\xi}(t) = \boldsymbol{b} \tag{42}$$

To solve (42) we make a discretization in time. Let

$$0 = t_0 < \cdots < t_n < \cdots < t_L = T \tag{43}$$

be a partition of the time interval I into $L + 1$ discrete time levels t_n spaced Δt apart. Further, let $\boldsymbol{\xi}^n$ denote an approximation to $\boldsymbol{\xi}(t_n)$. Replacing the time derivative $\dot{\boldsymbol{\xi}}$ by the simplest difference quotient we arrive at the Euler backward method.

Algorithm 1 Euler Backward Method.

1: Given ξ^0.
2: **for** $n = 0, \dots, L$ **do**
3: Solve the linear system

$$\boldsymbol{M}\frac{\boldsymbol{\xi}^{n+1} - \boldsymbol{\xi}^n}{\Delta t} + \boldsymbol{A}\boldsymbol{\xi}^{n+1} = \boldsymbol{b} \tag{44}$$

4: **end for**

The initial vector $\boldsymbol{\xi}^0$ is almost always taken as the nodal interpolant on $\boldsymbol{V}$ of the initial conditions u_1^0 and u_2^0, that is, $\boldsymbol{\xi}^0 = [\pi u_1^0,\ \pi u_2^0]$.

A Predator-Prey Model

We finally consider a classic application of (34) to ecology. Let u_1 and u_2 be the number of rabbits (prey) and foxes (predators) per acre within a forest Ω. A first crude model for the interaction between the two species could be

$$\dot{u}_1 - a_1 \Delta u_1 = c_1 u_1 (\bar{u}_2 - u_2) \tag{45a}$$

$$\dot{u}_2 - a_2 \Delta u_2 = c_2 u_2 (u_1 - \bar{u}_1) \tag{45b}$$

where a_i, c_i, and $\bar{u}_i$, $i = 1, 2$, are given constants. Roughly speaking we can think of $\bar{u}_2$ as a critical fox density for which the rabbits can reproduce at the same rate as they are killed. Similarly, $\bar{u}_1$ is a critical rabbit density at which the rabbits can precisely feed the foxes. The tendency of the species to move, or spread, to the surroundings are governed by the diffusion parameters a_i.

The boundary conditions can be of different types. For example, on the boundary of a large water reservoir we should have that $n \cdot \nabla u_i = 0$, since foxes and rabbits do not like to swim. However, along the boundary to a highway with heavy traffic, without a fence, and with attractive lands across the road, we should rather have $u_i = 0$, which means that all animals trying to pass the highway are killed by the traffic. We assume the former type of boundary conditions.

Problem 7. Make a variational formulation of the predator-pray problem (45). Formulate a finite element approximation and write down the resulting nonlinear discrete system of equations.

Below we list a code to compute the density of rabbits and foxes within a forest defined by the geometry matrix `geom`. For simplicity we have set all coefficients to unity.

```
[p,e,t]=initmesh(geom);
N=size(p,2);
eta =rand(N,1); % initial rabbit population
zeta=rand(N,1); %         fox
[K,M,ununsed]=assema(p,t,1,1,0); % assemble K and M
dt=0.01; % time step
time=0;
while time < 1 % time loop
  eta_old=eta;  zeta_old=zeta;
  for fixpt=1:2 % make two fixed point iterations
    eta =(M/dt+K)\(M/dt* eta_old+M*(eta.*(1-zeta)));
```

```
    zeta=(M/dt+K)\(M/dt*zeta_old+M*(zeta.*(eta-1)));
  end
  time=time+dt;
  figure(1), pdesurf(p,t,eta)
  figure(2), pdesurf(p,t,zeta)
end
```

Problem 8. ✔ Explain how the code `M*(zeta.*(eta-1))` occurring above can be used to approximate the load vector $F_i = (u_2(u_1 - 1), \varphi_i)$, $i = 1, \ldots, N$.

Problem 9. ✔ Simulate the density of rabbits and foxes on the unit square $\Omega = [0, 1]^2$ during the time span $0 \leq t \leq 1$. Start from a random distribution of rabbits and foxes. Make plots of your results.

Problem 10. ✔ Make a plot of the number of predators, pray and the total number of animals in the domain against time for $t \in [0, 20]$. Note that you at a given time may calculate the number of predators N_{pred} in the domain as

$$N_{\text{pred}} = \int_\Omega u_2 \, dx = [1, 1, \cdots, 1]\boldsymbol{M}[\zeta_1, \zeta_2, \cdots, \zeta_N]^T$$

Can you explain why the second equality holds? *Hint:* Note that you may write $1 = \sum_{i=1}^N \varphi_i(x)$.