Introduction to Finite Element Modelling in Geosciences

Basics of the FE method

Boris Kaus (kaus@erdw.ethz.ch)
Dave May (may@erdw.ethz.ch)

FS 2011

The model equation

We consider the equation
\[
A \frac{d^2 u(x)}{dx^2} + B = 0
\]

This equation is a first order, inhomogeneous ordinary differential equation.

Physically it describes for example:

• 1) The deflection of a stretched wire under a lateral load
  \( u = \text{deflection}, A = \text{tension}, B = \text{lateral load} \)
• 2) Steady state heat conduction with radiogenic heat production
  \( u = \text{temperature}, A = \text{thermal diffusivity}, B = \text{heat production} \)
• 3) Viscous fluid flow between parallel plates
  \( u = \text{velocity}, A = \text{viscosity}, B = \text{pressure gradient} \)
The analytical solution

We find the general analytical solution by integration

\[ \frac{\partial^2 u(x)}{\partial x^2} = -\frac{B}{A} \quad \text{Integrate} \int \, dx \]

\[ \frac{\partial u(x)}{\partial x} = -\frac{B}{A} x + C_1 \quad \text{Integrate} \int \, dx \]

\[ u(x) = -\frac{B}{2A} x^2 + C_1 x + C_2 \]

The two integration constants are found by two boundary conditions

\[ u(x = 0) = 0 \Rightarrow C_2 = 0 \]
\[ u(x = L) = 0 \Rightarrow C_1 = \frac{BL}{2A} \]

The special solution is

\[ u(x) = \frac{B}{2A} x^2 + \frac{BL}{2A} x \]

The weak formulation - 1

The original equation

\[ A \frac{\partial^2 u(x)}{\partial x^2} + B = 0 \]

Multiply with test function

\[ N(x) \left[ A \frac{\partial^2 u(x)}{\partial x^2} + B \right] = 0 \]

Integrate by parts

\[ \int_a^b \left[ \frac{\partial}{\partial x} \left( N(x) A \frac{\partial u(x)}{\partial x} \right) - N(x) \frac{\partial^2 u(x)}{\partial x^2} \right] \, dx = 0 \]

The product rule of differentiation

\[ \frac{\partial}{\partial x} (ab) = \frac{\partial a}{\partial x} b + a \frac{\partial b}{\partial x} \]

\[ a = N(x), \quad b = A \frac{\partial u(x)}{\partial x} \]
The weak formulation - 2

\[ N(x)A \frac{\partial u(x)}{\partial x} \bigg|_0^L - \int_0^L \left( \frac{\partial N(x)}{\partial x} A \frac{\partial u(x)}{\partial x} - N(x)B \right) dx = 0 \]

The first term requires our function \( u(x) \) at the boundaries \( x=0 \) and \( x=L \). Yet, here we specified the boundary conditions \( u(x=0)=0 \) and \( u(x=L)=0 \). Therefore, we do not need to test \( u(x) \) at these boundary points.

The above equation is the weak form of our original equations, because it has weaker constraints on the differentiability on our solution (only first derivative compared to second derivative in original equation).

Linear interpolation

The original equation:

\[ A \frac{\partial^2 u(x)}{\partial x^2} + B = 0 \]

Linear interpolation:

\[ u(x) = \hat{u}(x) = \frac{u_{i+1} - u_i}{\Delta x} x + u_i \]

\[ \hat{u}(x) = \frac{x}{\Delta x} u_{i+1} + \left( 1 - \frac{x}{\Delta x} \right) u_i \]

Straight line:

\[ y = ax + b \]
The finite element approximation - 1

We approximate our unknown solution as a sum of known functions, the so-called shape functions, multiplied with unknown coefficients. The shape functions are one at only one nodal point and zero at all other nodal points, so that the coefficient corresponding to a certain shape function is the solution at the node.

\[ u(x) = u_i N(x)_i + u_{i+1} N(x)_{i+1} = \{N(x)_i, N(x)_{i+1}\} \begin{bmatrix} u_i \\ u_{i+1} \end{bmatrix} = N^T u \]

\[ N(x)_i = 1 - \frac{x - x_i}{x_{i+1} - x_i} = 1 - \frac{x}{\Delta x} \]

\[ N(x)_{i+1} = \frac{x - x_i}{x_{i+1} - x_i} = \frac{x}{\Delta x} \]

\[ \begin{array}{c}
\int_a^b \frac{\partial}{\partial x} \left[ \begin{array}{c}
N(x) \\
N(x)_i
\end{array} \right] \begin{array}{c}
\frac{\partial}{\partial x} \left[ \begin{array}{c}
N(x)_i \\
N(x)_{i+1}
\end{array} \right] \\
\frac{\partial}{\partial x} \left[ \begin{array}{c}
N(x) \\
N(x)_i
\end{array} \right.
\end{array} \right] \begin{array}{c}
u_i \\
u_{i+1}
\end{array} - \begin{array}{c}
N(x)_i \\
N(x)_{i+1}
\end{array} \begin{array}{c}
u_i \\
u_{i+1}
\end{array} B \ dx = 0
\end{array} \]

\[ \int_a^b \frac{\partial}{\partial x} \left[ \begin{array}{c}
N(x) \\
N(x)_i
\end{array} \right] \begin{array}{c}
\frac{\partial}{\partial x} \left[ \begin{array}{c}
N(x)_i \\
N(x)_{i+1}
\end{array} \right] \\
\frac{\partial}{\partial x} \left[ \begin{array}{c}
N(x) \\
N(x)_i
\end{array} \right.
\end{array} \right] \begin{array}{c}
u_i \\
u_{i+1}
\end{array} - \begin{array}{c}
N(x)_i \\
N(x)_{i+1}
\end{array} \begin{array}{c}
u_i \\
u_{i+1}
\end{array} B \ dx = 0 \]

\[ Ku - F = 0 \]
The finite element approximation - 3

\[ Ku - F = 0 \]

\[ K = \int_0^L \begin{bmatrix} \frac{\partial N(x)}{\partial x} & \frac{\partial N(x)}{\partial x} & \frac{\partial N(x)}{\partial x} & \frac{\partial N(x)}{\partial x} \\ \frac{\partial N(x)}{\partial x} & \frac{\partial N(x)}{\partial x} & \frac{\partial N(x)}{\partial x} & \frac{\partial N(x)}{\partial x} \\ \frac{\partial N(x)}{\partial x} & \frac{\partial N(x)}{\partial x} & \frac{\partial N(x)}{\partial x} & \frac{\partial N(x)}{\partial x} \\ \frac{\partial N(x)}{\partial x} & \frac{\partial N(x)}{\partial x} & \frac{\partial N(x)}{\partial x} & \frac{\partial N(x)}{\partial x} \end{bmatrix} A \, dx, \quad F = \int_0^L \begin{bmatrix} N(x) \\ N(x) \end{bmatrix} B \, dx \]

\[ K_u = \int_0^L \frac{\partial N(x)}{\partial x} \frac{\partial N(x)}{\partial x} A \, dx = A \int_0^L \frac{1}{\Delta x} \, dx = 0 \]

\[ A \frac{\partial N(x)}{\partial x} \]

\[ N(x) = 1 - \frac{x-x_i}{\Delta x} = 1 - \frac{x}{\Delta x} \]

\[ N(x_i) = \frac{x-x_i}{\Delta x} = \frac{x_i}{\Delta x} \]

The finite element approximation - 4

Local system for element 1

\[ \begin{bmatrix} -\frac{A}{\Delta x} & \frac{A}{\Delta x} & 0 \\ \frac{A}{\Delta x} & -\frac{A}{\Delta x} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} \frac{-B \Delta x}{2} \\ \frac{-B \Delta x}{2} \\ 0 \end{bmatrix} \]

Global system for sum of all elements

\[ \begin{bmatrix} -\frac{A}{\Delta x} & \frac{A}{\Delta x} & 0 \\ \frac{A}{\Delta x} & -\frac{A}{\Delta x} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} \frac{-B \Delta x}{2} \\ \frac{-B \Delta x}{2} \\ 0 \end{bmatrix} \]
The finite element approximation - 5

Global system without boundary conditions

\[
\begin{bmatrix}
\frac{A}{\Delta x} & \frac{A}{\Delta x} & 0 \\
\frac{A}{\Delta x} & \frac{2A}{\Delta x} & \frac{A}{\Delta x} \\
0 & \frac{A}{\Delta x} & \frac{-A}{\Delta x}
\end{bmatrix}
\begin{bmatrix}
u_1 \\ u_2 \\ u_3
\end{bmatrix}
= \begin{bmatrix}
-\frac{B\Delta x}{2} \\ -\frac{B\Delta x}{2} \\ -\frac{B\Delta x}{2}
\end{bmatrix}
\]

Impose boundary conditions: 
\(u_1\) and \(u_3 = 0\)

\[
\begin{bmatrix}
1 & 0 & 0 \\
\frac{A}{\Delta x} & \frac{2A}{\Delta x} & \frac{A}{\Delta x} \\
0 & \frac{A}{\Delta x} & \frac{-A}{\Delta x}
\end{bmatrix}
\begin{bmatrix}
u_2 \\ u_3
\end{bmatrix}
= \begin{bmatrix}
0 \\ -B\Delta x \\ 0
\end{bmatrix}
\]

Put zeros on rows 1,3 and add 1 on the diagonal

Programming finite elements - 1

Nine finite elements

Introduce a node matrix!

The NODE matrix assigns to each local element the corresponding global node
Programming finite elements - 2

Nine finite elements

Global nodes
Local nodes

Ten nodes

Local system for finite element 4

\[
\begin{bmatrix}
\frac{A}{\Delta x} & \frac{A}{\Delta x} & u_1 \\
\frac{A}{\Delta x} & -\frac{A}{\Delta x} & u_2
\end{bmatrix}
\begin{bmatrix}
1 & 2 \\
2 & 3 \\
3 & 4 \\
4 & 5
\end{bmatrix}
\begin{bmatrix}
\frac{B\Delta x}{2} \\
\frac{B\Delta x}{2}
\end{bmatrix}
\begin{bmatrix}
u_4 \\
u_5
\end{bmatrix}
\]

Programming finite elements - 3

The global stiffness matrix is the sum of all local stiffness matrices.

In our code we will first setup a global stiffness matrix \(K_{\text{Global}}\) full of zeros.

We will then

1) loop through the finite elements,
2) identify where the local element is positioned within the global matrix and
3) add the local stiffness matrix at the correct position to the global matrix.

\[
\begin{bmatrix}
\frac{d^2}{dx^2} & \frac{d^2}{dx^2} \\
\frac{d^2}{dx^2} & \frac{d^2}{dx^2}
\end{bmatrix}
\]

\[
\begin{bmatrix}
\frac{d^2}{dx^2} & \frac{d^2}{dx^2} \\
\frac{d^2}{dx^2} & \frac{d^2}{dx^2}
\end{bmatrix}
\]

\[
\begin{bmatrix}
\frac{d^2}{dx^2} & \frac{d^2}{dx^2} \\
\frac{d^2}{dx^2} & \frac{d^2}{dx^2}
\end{bmatrix}
\]

\[
K_{\text{Global}}
\]

Exactly the same is done for the right hand side vector \(F\).
The FEM Maple code

```maple
restart;

# FEM
# Derivation of small matrix for 1D 2nd order steady state
# Number of nodes per element
nonel := 2;

# Shape functions
N[1] := (dx-x)/dx:
N[2] := x/dx:

# FEM approximation
u := sum( t[j]*N[j],j=1..nonel):

# Weak formulation of 1D equation
for i from 1 to nonel do
   Eq_weak[i] := int( -A*diff(u,x)*diff(N[i],x) + B*N[i] ,x=0..dx)
   od:

# Create element matrices for conductive and transient terms
K:=matrix(nonel,nonel,0):
F:=matrix(nonel,1,0):
for i from 1 to nonel do
   for j from 1 to nonel do
      K[i,j] := coeff(Eq_weak[i],t[j]):
   od:
   F[i,1] := -coeff(Eq_weak[i],B)*B:
od:

# Display K and F
matrix(K);
matrix(F);
```

The FEM Matlab code

```matlab
In the numerical solution all parameters can be made variable easily, while it gets more difficult to find analytical solutions for variable parameters, e.g., A and B.
```
FEM applications

Obtain the algebraic equations for each element (this is easy!)

Put all the element equations together

\[ [K] \{u\} = \{F\} \]

Taken from MIT lecture, de Weck and Kim

FEM element types

![Diagram of FEM element types](image)

Figure 6.1: Typical finite element geometries in one through three dimensions.

Taken from unknown web script
FEM applications

Separation of bubbles under influence of gravity

FEM applications

Viscous detachment folding (3D Stokes problem – you’ll do it in 2D)