Lecture 26: Solving the Heat Equation with Finite Elements

9: Heat Equation: Approximate Solution via the Finite Element Method

The Spectral Method gives exact series solutions to "nice" differential equations - problems for which we can easily find the associated eigenvalues and eigenfunctions. When those are not easy to compute, we turn to approximate methods (just as in the static case studied earlier).

(1) Derive the weak form of the PDE.

For \( u_t(x,t) = u_{xx}(x,t) + f(x,t) \) \((*)\)

with Dirichlet boundary conditions

\[ u(0,t) = u(1,t) = 0 \]

we use the space of test functions

\[ V = C^2_0 [0,1]. \]

Multiply \((*)\) by \( v \in V \) to get

\[ u_t(x,t) v(x) = u_{xx}(x,t) v(x) + f(x,t) v(x). \]
Integrate over the domain to get
\[
\int_0^1 U_t(x,t) V(x) \, dx = \int_0^1 U_{xx}(x,t) V(x) \, dx + \int_0^1 f(x,t)V(x) \, dx.
\]

Integrate the first term on the right by parts!
\[
\int_0^1 U_t(x,t) V(x) \, dx = \left[ U_x(x,t) V(x) \right]_0^1 - \int_0^1 U_x(x,t) V_x(x,t) \, dx + \int_0^1 f(x,t)V(x) \, dx.
\]

Since \( v \in V = C_0^2 [0,1] \), \( V(0)=V(1)=0 \) (an "essential" boundary condition) the boundary term is zero, giving:
\[
\int_0^1 U_t(x,t) V(x) \, dx = -\int_0^1 U_x(x,t) V_x(x,t) \, dx + \int_0^1 f(x,t)V(x) \, dx
\]
\[
= \mathcal{A}(U,V) \leq \text{"energy inner product"}
\]

Writing in inner product form,
\[
\langle U_t, V \rangle = -\mathcal{A}(U,V) + \langle f, V \rangle \quad \forall v \in V.
\]

(2) Impose weak form on a finite dimensional space:
Galerkin approximation

Let \( V_N = \text{Span} \{ \phi_1, \ldots, \phi_N \} \)
be an \( N \)-dimensional subspace of \( V \).
Now seek $U_N(x,t) = \sum_{j=1}^{N} a_j(t) \phi_j(x)$

That satisfies the weak form for all $v \in V_N$:

$$\left( \frac{\partial}{\partial t} U_N, v \right) = -a(U_N,v) + (f,v) \quad \forall v \in V_N$$

3. Extract a linear algebra problem from the Galerkin formulation

The Galerkin problem is solved if and only if

$$\left( \frac{\partial}{\partial t} U_N, \phi_k \right) = -a(U_N, \phi_k) + (f, \phi_k) \quad (*)$$

For all $k = 1, \ldots, N$ (since $\{\phi_k\}_{k=1}^{N}$ is a basis for $V_N$). Write $(*)$ out as

$$\left( \frac{\partial}{\partial t} \sum_{j=1}^{N} a_j(t) \phi_j(x), \phi_k \right) = -a \left( \sum_{j=1}^{N} a_j(t) \phi_j(x), \phi_k \right) + (f, \phi_k)$$

Use linearity of the inner product to get

$$\sum_{j=1}^{N} \frac{\partial}{\partial t} a_j(t) (\phi_j, \phi_k) = -a \sum_{j=1}^{N} a_j(t) a(\phi_j, \phi_k) + (f, \phi_k).$$

For example, when $k=1$, this means

$$\begin{bmatrix} (\phi_1, \phi_1) & \cdots & (\phi_N, \phi_1) \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial t} a_1(t) \\
\vdots \\
\frac{\partial}{\partial t} a_N(t) \end{bmatrix} = -a \begin{bmatrix} (\phi_1, \phi_1) & \cdots & (\phi_N, \phi_1) \end{bmatrix} \begin{bmatrix} a_1(t) \\
\vdots \\
a_N(t) \end{bmatrix} + \begin{bmatrix} (f, \phi_1) \\
\vdots \\
(f, \phi_N) \end{bmatrix}$$
Stacking these equations for $k=1,...,N$:

\[
\begin{bmatrix}
(\phi_1, \phi_1) & \cdots & (\phi_1, \phi_N) \\
\vdots & \ddots & \vdots \\
(\phi_N, \phi_1) & \cdots & (\phi_N, \phi_N)
\end{bmatrix}
\begin{bmatrix}
\ddot{a}_1(t) \\
\vdots \\
\ddot{a}_N(t)
\end{bmatrix}
= -\begin{bmatrix}
2(\phi_1, \phi_1) & \cdots & 2(\phi_1, \phi_N) \\
\vdots & \ddots & \vdots \\
2(\phi_N, \phi_1) & \cdots & 2(\phi_N, \phi_N)
\end{bmatrix}
\begin{bmatrix}
a_1(t) \\
\vdots \\
a_N(t)
\end{bmatrix}
+ \begin{bmatrix}
(\phi_1, \phi_1) \\
\vdots \\
(\phi_N, \phi_N)
\end{bmatrix}
\begin{bmatrix}
f_1(t) \\
\vdots \\
f_N(t)
\end{bmatrix}
\]

"Mass matrix" $M$

"Stiffness matrix" $K$

"Load vector" $f(t)$

So we compactly write this as a first order system of $N$ coupled linear ODES:

\[
Ma'(t) = -K\dot{a}(t) + f(t)
\]  

\(\text{(***)}\)

Here, for our usual hat functions,

\[
M = h \begin{bmatrix}
\frac{2}{3} & \frac{1}{6} & 0 \\
\frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\
0 & \frac{1}{6} & \frac{2}{3}
\end{bmatrix}, \quad K = \frac{1}{h} \begin{bmatrix}
-2 & 1 & 0 \\
1 & 1 & 0 \\
0 & 1 & -2
\end{bmatrix}
\]

To make (***) well defined, we need initial conditions for $a(t)$. 

\text{Initial conditions for } a(t).
We can extract initial conditions in two ways:

(3) Make \( U_N(x,0) \) equal the best approximation of \( U_0(x) \) from span \( \{\phi_1, \ldots, \phi_N\} \).

Using techniques from earlier this semester, this will require

\[
\begin{bmatrix}
(\phi_1, \phi_1) & \cdots & (\phi_1, \phi_N) \\
\vdots & \ddots & \vdots \\
(\phi_N, \phi_1) & \cdots & (\phi_N, \phi_N)
\end{bmatrix}
\begin{bmatrix}
a_1(0) \\
\vdots \\
a_N(0)
\end{bmatrix} =
\begin{bmatrix}
(u_1, \phi_1) \\
\vdots \\
(u_N, \phi_N)
\end{bmatrix}
\]

\[\Rightarrow M a(0) = q \quad \text{(solve for } a(0))\]

(6) This is the easy/conventional approach for hat functions: if we want \( U_N(x,0) \) to interpolate \( U_0(x) \) at \( x_1, x_2, \ldots, x_N \), use the fact that

\[ U_N(x_k,0) = \sum_{j=1}^{N} a_j(0) \phi_j(x_k) = \phi_k(x_k) \]

To simply set

\[ \phi_k(x_k) = U_0(x_k) \quad k = 1, \ldots, N \]

\[\Rightarrow a(0) =
\begin{bmatrix}
U_0(x_1) \\
\vdots \\
U_0(x_N)
\end{bmatrix}
\]

Now, solve \( a(t') = -M^t (k a(t) + f(t)) \) for \( a(t) \)......