Lecture 28: Approximate Solution of the Heat Equation in Time.

Thus far we have discretized the heat equation in space to obtain the ODE

$$(*) \qquad a'(t) = -M^{-1}K\,a(t)$$

which we solved <u>exactly in time</u> via the matrix exponential

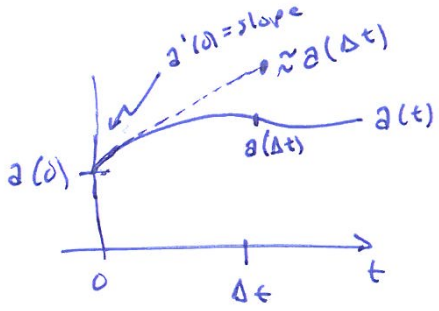
$$a(t) = e^{-M^{-1}Kt}\,a(0).$$

This has the advantage that we can evaluate $a(t)$ at any t, if we are curious about the solution at some specific time. However, the exponential of a matrix is expensive to compute. (See the famous paper by Moler and Van Loan: "Nineteen Dubious Ways to Compute the Exponential of a Matrix.")

In many cases, we will prefer to solve $(*)$ approximately in time. In this lecture we use the simplest possible method.

Note that $(*)$ gives a formula for the <u>slope</u> of the solution at time t: we can use this insight to step forward from $t=0$ to time $t=\Delta t$:

$$a(\Delta t) \approx a(0) + \Delta t\, a'(0)$$



Similarly, we can derive this formula via the definition of the derivative:

$$a'(0) = \lim_{\Delta t \to 0} \frac{a(\Delta t) - a(0)}{\Delta t}$$

For fixed Δt, we approximate

$$a'(0) \approx \frac{a(\Delta t) - a(0)}{\Delta t}$$

$$\Rightarrow a(\Delta t) \approx a(0) + \Delta t\, a'(0).$$

Now with an approximation to $a(\Delta t)$, we can similarly get an approximation to $a(2\Delta t)$:

$$a(2\Delta t) \approx a(\Delta t) + \Delta t\, a'(\Delta t)$$

We introduce the notation

$$a_j \approx a(j\Delta t)$$

to approximate the solution at time $t = j\Delta t$:

FORWARD EULER METHOD

$$a_{j+1} = a_j + \Delta t \underbrace{\left(-M^{-1}\left(Ka + f(j\Delta t)\right)\right)}_{\text{Slope of solution from } a'(t) = -M^{-1}(Ka(t) + f(t))}.$$

For a general differential equation,

$$y'(t) = f(t, y(t))$$

Forward Euler is written as

$$y_{j+1} = y_j + \Delta t\, f(j\Delta t, y_j).$$

We focus on the case $f=0$, so the forward Euler method becomes

$$a_{j+1} = a_j - \Delta t\, M^{-1} K a_j$$
$$= (I - \Delta t\, M^{-1} K) a_j.$$

Thus we can write the j^{th} iterate as

$$\boxed{a_j = (I - \Delta t\, M^{-1} K)^j a_0}$$

(This is not a good way to implement forward Euler, but it is very helpful for analysis.)

Now we try to implement this in MATLAB: see

heat_fem_fe.m

on the class website

Notice: if Δt is not carefully chosen, the approximate solution a_j blows up as $j \to \infty$, despite the fact that the exact solution gives $u(x,t) \to 0$ as $t \to \infty$. If Δt is taken sufficiently small, the solution does mimic the exact solution: $a_j \to 0$.

To understand the interplay of Δt and the number N of basis functions in the finite element method.

KEY

[If N increases, how does the largest stable Δt change? Does it increase, decrease, or stay the same?