Finite element methods for initial boundary value problems (IBVPs)

Last week, we developed finite difference methods for the heat equation, a type of initial boundary value problem (IBVP). The key to developing our method was to first discretize the equations in space to arrive at an IVP. We were then able to directly apply our techniques for solving IVPs. In the notes and homework, we used the trapezoid method to advance the spatially discrete IVP.

This week, we will focus on developing a finite element method for the heat equation. The technique for doing this is strongly analogous to what we did last week: use the method of lines to spatially discretize the equation, and solve the resulting IVP using your favorite time stepping method. So what is different now that we are using the finite element method? The difference is that we will spatially discretize the equations using a local spectral method (rather than using local interpolation, which is our approach last week).

If you need a refresher on the heat equation, check out last week's notes! Also, remember that while we developed FD methods for the heat equation, the same techniques can be extended beyond the heat equation and into a broad range of IBVPs!

Here's a fun and rarely mentioned observation: calling this week's approach of solving the heat equation the "finite element method" is something of a misnomer. While it is true that we are using the finite element method to spatially discretize the equations, we will inevitably solve the resulting IVP using a finite difference method (we only ever used FD methods for IVPs in this class!). So really, we are using a combined finite element + finite difference method in this week's approach. However, historical convention suggests that we refer to this hybrid method as a finite element method, and that is the convention we will adopt.

1 *Numerical discretization: the method of lines*

Just as was done last week, we will use the method of lines to first *discretize* the IBVP in space, which will yield an IVP that we can then solve using our favorite time integrator. Remember that our spatial discretization approach this week is the finite element method, so we will be using a local spectral method. Let us see how this process works in detail.

1.1 *Step 1: discretize in space*

If you feel uncomfortable with the presentation in section 1.1, now is a great time to go back to the typed notes on the finite element method for the 1D Poisson problem. You will find many similarities between the two sets of notes!

To implement our local spectral method, we need (i) a spatial discretization of the domain and (ii) a vector space that our solution belongs to. For the first requirements (i), we break up the continuous space variable into a finite number of pieces as

$$x_j = a + \frac{(b-a)(j-1)}{n}, \quad j = 1, \ldots, n+1 \tag{1}$$

Here is a reminder of what the spatial discretization looks like.

For the second requirement (ii), we use the vector space $\mathcal{V}_n^L$, which you will recall from the typed notes on FEMs for BVPs is defined as the collection of functions on $[a, b]$ that can be defined piecewise on the points $x_1, \ldots, x_{n+1}$ and that automatically satisfy the boundary conditions. Just as in our exploration of for FEMs BVPs, we will assume zero Dirichlet boundary conditions when deriving methods. In the notation of last week's notes, this assumption amounts to

There are ways of addressing nonzero boundary conditions, but doing so here would complicate our derivation without adding much to our fundamental understanding.

saying that $g_a(t) = g_b(t) = 0$. Since we are familiar with this vector space $\mathcal{V}_n^L$, let us also note that we have a basis for this space given by $\{\phi_2, \ldots, \phi_n\}$, where ϕ_j ($j = 2, \ldots, n$) is the j^{th} hat function.

Now that we have addressed points (i) and (ii), we can derive our spectral method by requiring that our approximate solution $\hat{u}(x,t) \in \mathcal{V}_n^L$ minimize the least-squares error from the true solution $u(x,t)$. That is, we need

$$\hat{u} = \min_{\tilde{u} \in \mathcal{V}_n^L} ||u - \tilde{u}|| \tag{2}$$

Now remember from our discussion of best least-squares approximation that the $\hat{u}$ that satisfies this requirement obeys the following relationship:

$$(u - \hat{u}, \phi_j) = 0, \quad j = 2, \ldots, n \tag{3}$$

for some inner product $(\cdot, \cdot)$. You will remember that for BVPs the appropriate inner product is the energy inner product $(\cdot, \cdot)_E$, which *depends on the differential equation*. In our case, we remember that the heat equation is just an unsteady version of the Poisson equation, so the same notion of energy (representing heat flux through the domain) defined a few weeks ago applies here as well.

Analogous to the BVP setting, we define the energy inner product for the heat equation as

$$(f, g)_E = \kappa \int_a^b f'(x,t) g'(x,t) dx, \quad \forall f, g \in \mathcal{V}_n^L \tag{4}$$

The only differences here from the BVP setting is that we now allow the functions to depend on time, and we include the diffusivity κ.

So far, much of what we have done is entirely analogous to the procedure described in the BVP setting. So what is the difference? The difference is that this week, our solution is time-dependent, and associated with a differential equation that is time-dependent. Let us see what implications this time dependence has on the derivation of method.

We will write our approximate solution as a linear combination of our basis functions

$$\hat{u}(x,t) = \sum_{i=2}^{n} u_i(t) \phi_i(x) \tag{5}$$

Remember from the BVP setting that using our hat function basis, the coefficients u_i in the expansion (5) are simply an approximation of the solution $u(x,t)$ at the point $x = x_i$. However, unlike in the BVP setting, these coefficients must now depend on time since the real solution $u(x_i, t)$ depends on time.

Plugging our expression (5) into the orthogonality relation (3) and assuming the energy inner product gives

$$\left(u - \sum_{i=2}^{n} u_i \phi_i, \phi_j \right)_E = 0, \quad j = 2, \ldots, n \tag{6}$$

$$\implies \quad \sum_{i=2}^{n} u_i \left(\phi_i, \phi_j \right)_E = \left(u, \phi_j \right)_E, \quad j = 2, \ldots, n \tag{7}$$

We can simplify this relation further by writing out the righthand

side as

$$\left(u, \phi_j\right)_E = \kappa \int_a^b u'(x,t)\phi_j'(x)dx \tag{8}$$

$$= \kappa[u'(x,t)\phi_j(x)]_a^b - \kappa \int_a^b u''(x,t)\phi_j(x)dx \quad \text{[Integrate by parts]} \tag{9}$$

$$= -\kappa \int_a^b u''(x,t)\phi_j(x)dx \quad \left[\phi_j \in \mathcal{V}_n^L \text{ so it satisfies the BCs}\right] \tag{10}$$

$$= \int_a^b \left[g(x,t) - \frac{\partial u}{\partial t}\right]\phi_j(x)dx \quad [u(x,t) \text{ satisfies the heat equation}] \tag{11}$$

$$= \int_a^b g(x,t)\phi_j(x)dx - \int_a^b \frac{\partial u}{\partial t}\phi_j(x)dx \tag{12}$$

At this point, it may seem like we are running in circles: after all our hard work we have ended up at a point where we have the unknown solution in the right hand side. Do not give up hope! The way we will overcome this obstacle is to embrace the fact that we are approximating u by $\hat{u}$, so we may replace u in (12) with $\hat{u}$ and use the expression (5) for $\hat{u}$ in terms of the hat basis functions to rewrite (12) as

$$\left(u, \phi_j\right)_E = \int_a^b g(x,t)\phi_j(x)dx - \int_a^b \frac{\partial}{\partial t}\left[\sum_{i=2}^n u_i(t)\phi_i(x)\right]\phi_j(x)dx \tag{13}$$

$$= \int_a^b g(x,t)\phi_j(x)dx - \frac{\partial}{\partial t}\left[\sum_{i=2}^n u_i(t)\right]\int_a^b \phi_i(x)\phi_j(x)dx \tag{14}$$

$$= \int_a^b g(x,t)\phi_j(x)dx - \sum_{i=2}^n \dot{u}_i(t)\int_a^b \phi_i(x)\phi_j(x)dx \tag{15}$$

Now we are in business. Let us substitute this final expression (15) for the righthand side of (7) back into (7) to get

$$\sum_{i=2}^n u_i \left(\phi_i, \phi_j\right)_E = \int_a^b g(x,t)\phi_j(x)dx - \sum_{i=2}^n \dot{u}_i(t)\int_a^b \phi_i(x)\phi_j(x)dx, \quad j = 2, \ldots, n \tag{17}$$

$$= \left(g, \phi_j\right)_s - \sum_{i=2}^n \dot{u}_i(t)\left(\phi_i, \phi_j\right)_s \tag{18}$$

We can move the time derivative term over to the lefthand side and the lefthand side term to the righthand side, and express the

Remember our definition that $(\cdot,\cdot)_s$ is the standard inner product defined as

$$(q,p)_s = \int_a^b q(x,t)p(x,t)dx \tag{16}$$

for $q, p \in \mathcal{V}_n^L$.

result in matrix form to get

$$\begin{bmatrix} (\phi_2,\phi_2)_s & (\phi_3,\phi_2)_s & \cdots & (\phi_{n-1},\phi_2)_s & (\phi_n,\phi_2)_s \\ (\phi_2,\phi_3)_s & (\phi_3,\phi_3)_s & \cdots & (\phi_{n-1},\phi_3)_s & (\phi_n,\phi_3)_s \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ (\phi_2,\phi_{n-1})_s & (\phi_3,\phi_{n-1})_s & \cdots & (\phi_{n-1},\phi_{n-1})_s & (\phi_n,\phi_{n-1})_s \\ (\phi_2,\phi_n)_s & (\phi_3,\phi_n)_s & \cdots & (\phi_{n-1},\phi_n)_s & (\phi_n,\phi_n)_s \end{bmatrix} \begin{bmatrix} \dot{u}_2 \\ u_3 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix} =$$

$$- \begin{bmatrix} (\phi_2,\phi_2)_E & (\phi_3,\phi_2)_E & \cdots & (\phi_{n-1},\phi_2)_E & (\phi_n,\phi_2)_E \\ (\phi_2,\phi_3)_E & (\phi_3,\phi_3)_E & \cdots & (\phi_{n-1},\phi_3)_E & (\phi_n,\phi_3)_E \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ (\phi_2,\phi_{n-1})_E & (\phi_3,\phi_{n-1})_E & \cdots & (\phi_{n-1},\phi_{n-1})_E & (\phi_n,\phi_{n-1})_E \\ (\phi_2,\phi_n)_E & (\phi_3,\phi_n)_E & \cdots & (\phi_{n-1},\phi_n)_E & (\phi_n,\phi_n)_E \end{bmatrix} \begin{bmatrix} u_2 \\ u_3 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix} + \begin{bmatrix} (g,\phi_2)_s \\ (g,\phi_3)_s \\ \vdots \\ (g,\phi_{n-1})_s \\ (g,\phi_n)_s \end{bmatrix} \tag{19}$$

Let us write this matrix system more succinctly as $\boldsymbol{M}\dot{\boldsymbol{u}} = \boldsymbol{A}\boldsymbol{u} + \boldsymbol{g}(t)$. This is great progress! We have taken the heat equation IBVP and used a local spectral method to remove the dependence on space and obtain an initial value problem. The only issue is that because of the matrix $\boldsymbol{M}$ on the lefthand side, the IVP is not yet in the desired form $\dot{\boldsymbol{u}} = \boldsymbol{f}(\boldsymbol{u}, t)$. We can fix this by pre-multiplying both sides of the equation by $\boldsymbol{M}^{-1}$, giving the desired IVP $\dot{\boldsymbol{u}} = \boldsymbol{f}(\boldsymbol{u}, t)$ with $\boldsymbol{f}(\boldsymbol{u}, t) = \boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{u} + \boldsymbol{g}(t))$.

Before discussing how we will advance the system in time, let us simplify the $\boldsymbol{M}$ and $\boldsymbol{A}$ matrices a bit further. First, remember from our study of FEMs for BVPs that $\boldsymbol{A}$ is the tridiagonal matrix given by

$$\boldsymbol{A} = \frac{\kappa}{\Delta x} \begin{bmatrix} -2 & 1 & \cdots & 0 & 0 \\ 1 & -2 & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & -2 & 1 \\ 0 & 0 & \cdots & 1 & -2 \end{bmatrix} \tag{20}$$

Now, regarding $\boldsymbol{M}$, the question reduces to computing $(\phi_i, \phi_j)_s$ ($i = 2, \ldots, n$; $j = 2, \ldots, n$). Given the narrow sub-region of the domain $[a, b]$ over which each hat function is nonzero, we can intuit that the only terms that are nonzero will be

$$(\phi_{i-1}, \phi_i)_s, \quad (\phi_i, \phi_i)_s, \quad (\phi_i, \phi_{i+1})_s \tag{21}$$

These expressions may be evaluated as

$$\begin{aligned} (\phi_{i-1}, \phi_i)_s &= \int_{a+(i-1)\Delta x}^{a+i\Delta x} \phi_{i-1}(x)\phi_i(x)dx = \frac{\Delta x}{6} \\ (\phi_i, \phi_i)_s &= \int_{a+(i-1)\Delta x}^{a+(i+1)\Delta x} \phi_i(x)\phi_i(x)dx = \frac{2\Delta x}{3} \\ (\phi_i, \phi_{i+1})_s &= \int_{a+i\Delta x}^{a+(i+1)\Delta x} \phi_i(x)\phi_{i+1}(x)dx = \frac{\Delta x}{6} \end{aligned} \tag{22}$$

We computed the specific numerical value of each of the integrals by plugging in the mathematical expressions for each of the hat functions from week 12 into the integral expression and evaluating it using Mathematica.

We may therefore write $\boldsymbol{M}$ as

$$\boldsymbol{M} = \frac{\Delta x}{6}\begin{bmatrix} 4 & 1 & \cdots & 0 & 0 \\ 1 & 4 & \cdots & 0 & 0 \\ \vdots & \vdots & \ldots & \vdots & \vdots \\ 0 & 0 & \cdots & 4 & 1 \\ 0 & 0 & \cdots & 1 & 4 \end{bmatrix} \tag{23}$$

We have finally arrived at our desired IVP, which is given by $\dot{\boldsymbol{u}} = \boldsymbol{f}(\boldsymbol{u}, t)$ with $\boldsymbol{f}(\boldsymbol{u}, t) = \boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{u} + \boldsymbol{g}(t))$, where $\boldsymbol{A}$ is given by (20) and $\boldsymbol{M}$ is given by (23). Since this is an IVP, you will recall that we need an initial condition to advance the solution in time. Using the initial condition from last week's typed notes that the true solution $u(x, t = 0) = \eta(x)$, we may create our initial condition for $\boldsymbol{u}(t)$ as

$$\boldsymbol{u}(t = 0) = \begin{bmatrix} \eta(x_2) \\ \eta(x_3) \\ \vdots \\ \eta(x_{n-1}) \\ \eta(x_n) \end{bmatrix} \tag{24}$$

1.2 *Step 2: solve the initial value problem*

Now that we have our IVP in the required first order form, we may use a finite difference method to advance the solution in time. As was discussed in last week's notes, the IVP induced by spatially discretizing the heat equation imposes severe stability restrictions on our time stepping algorithm, and we must therefore use an implicit method with a large stability region to advance the system in time.

The reason that a time stepping method with a large stability region is useful is that the spatially discrete heat equation is *stiff*: it has some eigenvalues very near the origin but some that extend very far into the left-half complex plane, requiring very small time steps for methods with small stability regions. Remember that this stiff property of the spatially discrete IVP is a feature of the heat equation, and *not* generic to spatially discrete IVPs. There are many IBVPs that produce non-stiff IVPs when spatially discretized, which may be handled using a favorite explicit time stepping method.

A common choice for the time stepping method is the trapezoid method, which becomes

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \frac{\Delta t}{2}(\boldsymbol{f}(\boldsymbol{u}_k, t_k) + \boldsymbol{f}(\boldsymbol{u}_{k+1}, t_{k+1})) \tag{25}$$

Using $\boldsymbol{f}(\boldsymbol{u}, t) = \boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{u} + \boldsymbol{g}(t))$, we can recast (25) as

$$\begin{aligned} \boldsymbol{u}_{k+1} &= \boldsymbol{u}_k + \frac{\Delta t}{2}\left[\boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{u}_k + \boldsymbol{g}(t_k)) + \boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{u}_{k+1} + \boldsymbol{g}(t_{k+1}))\right] \\ \implies \left(\boldsymbol{I} - \frac{\Delta t}{2}\boldsymbol{M}^{-1}\boldsymbol{A}\right)\boldsymbol{u}_{k+1} &= \boldsymbol{u}_k + \frac{\Delta t}{2}\boldsymbol{M}^{-1}\left[\boldsymbol{A}\boldsymbol{u}_k + \boldsymbol{g}(t_k) + \boldsymbol{g}(t_{k+1})\right] \\ \implies \boldsymbol{u}_{k+1} &= \left(\boldsymbol{I} - \frac{\Delta t}{2}\boldsymbol{M}^{-1}\boldsymbol{A}\right)^{-1}\left[\boldsymbol{u}_k + \frac{\Delta t}{2}\boldsymbol{M}^{-1}\left[\boldsymbol{A}\boldsymbol{u}_k + \boldsymbol{g}(t_k) + \boldsymbol{g}(t_{k+1})\right]\right] \end{aligned} \tag{26}$$

We can advance this system in time using the initial condition $\boldsymbol{u}_0 = \boldsymbol{u}(t = 0)$ provided in (24).

2 *Some more notes*

Just as we did last week, we have left many questions unanswered. Does the numerical solution converge to the true solution? If so, how does this convergence depend on Δt and Δx? Unfortunately, we will not have time to discuss these topics in detail in this class. So here are the punchline answers to those questions:

1. Using a finite element method in space and a finite difference method in time does produce convergent solutions, provided the FEM and FDM are convergent in space and time, respectively.

2. The convergence rate in space is equal to the order of the spatial discretization used, and the convergence rate in time is equal to the convergence rate of the time stepping method used for the IVP.