Solving PDE’s with FEniCS

Heat equation

L. Ridgway Scott

The Institute for Biophysical Dynamics, The Computation Institute, and the Departments of Computer Science and Mathematics, The University of Chicago
Heat can be exchanged between two different bodies by diffusion, convection or radiation.

The **heat equation** describes diffusion of thermal energy in a medium [1].

The simplest form of the heat equation takes the form

\[
\frac{\partial u}{\partial \tau}(\xi, \tau) - c \Delta u(\xi, \tau) = f(\xi, \tau) \quad \forall \xi \in \tilde{\Omega}, \, \tau > 0
\]

\[
u(\xi, 0) = u_0(\xi) \quad \forall \xi \in \tilde{\Omega},
\]

(1)

together with boundary conditions imposed on \(\partial \tilde{\Omega}\) that will be discussed subsequently.
Have distinguished variable $\tau$ that we think of as “time” [2] together with the spatial variable $\xi$.

The natural domain for the problem is a cylinder domain $\Omega \times [0, T]$ as indicated in the figure:
The diffusion coefficient, $c$, takes care of the mismatch in units; $c$ has units of length-squared divided by time.

The heat equation is more generally referred to as the **diffusion equation**, and it governs the diffusion of many materials.

Some examples of diffusion coefficients are given in Table 1.

We see that these coefficients can differ by orders of magnitude.
## Diffusion coefficients

<table>
<thead>
<tr>
<th>material</th>
<th>diffusion coefficient</th>
<th>medium</th>
<th>conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>H$_2$</td>
<td>1.6 cm$^2$/sec</td>
<td>self</td>
<td>T = 273 K, p = 0.1 MPa</td>
</tr>
<tr>
<td>CO$_2$</td>
<td>.106 cm$^2$/sec</td>
<td>self</td>
<td>T = 273 K, p = 0.1 MPa</td>
</tr>
<tr>
<td>CO$_2$</td>
<td>1.92 $\times$ 10$^{-5}$ cm$^2$/sec</td>
<td>water</td>
<td>T = 298 K</td>
</tr>
<tr>
<td>sucrose</td>
<td>0.52 $\times$ 10$^{-5}$ cm$^2$/sec</td>
<td>water</td>
<td>T = 25 C</td>
</tr>
<tr>
<td>hydrogen</td>
<td>1.66 $\times$ 10$^{-9}$ cm$^2$/sec</td>
<td>iron</td>
<td>T = 10 C</td>
</tr>
<tr>
<td>hydrogen</td>
<td>1.24 $\times$ 10$^{-7}$ cm$^2$/sec</td>
<td>iron</td>
<td>T = 100 C</td>
</tr>
<tr>
<td>aluminum</td>
<td>1.3 $\times$ 10$^{-30}$ cm$^2$/sec</td>
<td>copper</td>
<td>T = 20 C</td>
</tr>
</tbody>
</table>

Table 1: Diffusion coefficients for various materials in various media. When the medium is “self” the coefficient is the self-diffusion constant.
It is frequently useful to remove the units in a PDE, that is, to \textbf{nondimensionalize} the equation.

For the heat/diffusion equation, this allows us to develop some useful intuition that is independent of the domain of application.

We can do this by changing the spatial and time variables:

$$x = a\xi, \quad t = b\tau, \quad a \neq 0, \quad b \neq 0.$$
With this change of variables, the first equation in (1) becomes

$$b \frac{\partial u}{\partial t}(x, t) - ca^2 \Delta_x u(x, t) = f(x, t) \quad \forall x \in \Omega, \; t > 0,$$

(2)

where $\Omega = \{ x : a^{-1}x \in \tilde{\Omega} \}$ and we added the subscript $x$ to $\Delta$ to clarify that it is

$$\Delta_x = \frac{\partial^2}{\partial x_1^2} + \cdots + \frac{\partial^2}{\partial x_d^2},$$

as opposed to the meaning in the first equation in (1).
Defining $\hat{f}(x, t) = b^{-1} f(x/a, t/b)$, we find

$$\frac{\partial u}{\partial t}(x, t) - \frac{ca^2}{b} \Delta_x u(x, t) = \hat{f}(x, t) \quad \forall x \in \tilde{\Omega}, \ t > 0. \quad (3)$$

We then have wide latitude to choose the time and/or space coordinates so that

$$ca^2 = b.$$

In such coordinates, the diffusion equation simplifies to

$$\frac{\partial u}{\partial t}(x, t) - \Delta_x u(x, t) = \hat{f}(x, t) \quad \forall x \in \Omega, \ t > 0. \quad (4)$$

We now explore this in detail in one space dimension.
In its simplest, one-dimensional form, it may be written

\[
\frac{\partial u}{\partial t}(x, t) - \frac{\partial^2 u}{\partial x^2}(x, t) = f(x, t) \quad \forall x \in [0, 1], \ t > 0
\]

(5)

\[u(x, 0) = u_0(x) \quad \forall x \in [0, 1]\]

where \(u(x, t)\) denotes temperature of medium at point \(x\) and time \(t\).

Simple example: transfer of heat across a window.

The variable \(x\) denotes the distance from one face of the window pane to the other, in the direction perpendicular to the plane of the window.
Near outer edges of the window, three dimensional effects would be evident.

But in the middle of the window, equation (5) would accurately describe the evolution of the temperature $u$ inside the window.

The function $f$ is included for completeness, but in many cases such a body source of heat would be zero.

These equations must be supplemented by boundary conditions similar to the ones considered previously.
They could be of purely Dirichlet (or essential) type, viz.

\[ u(0, t) = g_0(t), \quad u(1, t) = g_1(t) \quad \forall t > 0, \quad (6) \]

or of purely Neumann (or natural) type, viz.

\[ \frac{\partial u}{\partial x}(0, t) = g_0(t), \quad \frac{\partial u}{\partial x}(1, t) = g_1(t) \quad \forall t > 0, \quad (7) \]

Or a combination of the two:

\[ u(0, t) = g_0(t), \quad \frac{\partial u}{\partial x}(1, t) = g_1(t) \quad \forall t > 0. \quad (8) \]

Here \( g_i, \ i = 0, 1, \) are given functions of \( t. \)
The main characteristic of the heat equation is that it smooths any roughness in the initial data.

For example, in Figure 1 we show the solution at time $t = 0.001$ for the case

$$u_0(x) = \frac{1}{2} - |x - \frac{1}{2}|.$$  \hspace{1cm} (9)

We see that the discontinuity of the derivative of $u_0$ at $x = \frac{1}{2}$ is instantly smoothed.

This has a corollary for the backwards heat equation that we will explore subsequently.
Figure 1: Solution of (5) with initial data (9) at time $t = 0.001$. Computed with piecewise linears with 50 mesh points (uniform mesh).
Compatibility conditions

One type of nonsmooth behavior stems from a mismatch between boundary data and initial data.

This is governed by compatibility conditions.

It is interesting to note that the pure Neumann condition (7) for the heat equation (5) does not suffer the same limitations on the data, or nonuniqueness of solutions, that the steady state counterpart does.

However, there are compatibility conditions required to obtain smooth solutions, linking the boundary and initial data for the heat equation in order to have a smooth solution.
Compatibility conditions derived from observation that the values of $u$ on the spatial boundary have been specified twice at $t = 0$.

Consider the case (8) of combined Dirichlet and Neumann boundary conditions.

The first set of compatibility conditions is

$$u_0(0) = u(0, 0) = g_0(0) \text{ and } u'_0(1) = u_x(1, 0) = g_1(0). \quad (10)$$

These are obtained by matching the two ways of specifying the solution at the boundary points $(x, t) = (0, 0)$ and $(x, t) = (1, 0)$. 
In the case of pure Dirichlet conditions (6) the compatibility conditions become

\[ u_0(0) = u(0, 0) = g_0(0) \text{ and } u_0(1) = u(1, 0) = g_1(0). \]  
(11)

In the case of pure Neumann conditions (7) the compatibility conditions become

\[ u'_0(0) = u_x(0, 0) = g_0(0) \text{ and } u'_0(1) = u_x(1, 0) = g_1(0). \]  
(12)

Conditions involving derivatives (Neumann boundary conditions) are higher-order than those for function values (Dirichlet boundary conditions).

They affect bounds of higher-order derivatives.
Figure 2: Heat equation with incompatible data after one time step with $\Delta t = 10^{-5}$; degree = 1, 20 mesh intervals. Initial values $u_0 = 0.5$. 
Figure 3: Heat equation with incompatible data after one time step with $\Delta t = 10^{-5}$; degree = 2, 10 mesh intervals. Initial values $u_0 = 0.5$. 
Higher order conditions

For an arbitrary order of smoothness, there are infinitely many compatibility conditions.

Second set of conditions uses differential equation $u_{xx} = u_t$ to trade spatial derivatives for temporal ones, then applying this at $t = 0$ and $x = 0$:

**Dirichlet case:**

$$u''_0(0) = u_{xx}(0, 0) = u_t(0, 0) = g'_0(0).$$

**Neumann case:**

$$u'''_0(1) = u_{xxx}(1, 0) = u_{xt}(1, 0) = g'_1(0).$$
Compatibility implications

If compatibilities are not satisfied by data, oscillations will result near \( t = 0 \) and \( x = 0, 1 \).

In nonlinear problems, this can cause completely wrong results to occur.

Compatibility conditions do not have to be satisfied for heat equation (5) to be well posed.

There is a unique solution in any case, but the physical model may be incorrect as a result if it is supposed to have a smooth solution.
Compatibility conditions are a subtle form of constraint on model quality.

In many problems they can be described in terms of local differential-algebraic constraints.

However, such compatibility conditions for the Navier-Stokes equations can lead to global constraints that are hard to verify or satisfy in practice.
It is possible to derive a variational formulation involving integration over both $x$ and $t$, but it is more common to use a variational formulation based on $x$ alone.

We seek a function $\tilde{u}(t)$ of time with values in $V$ such that $\tilde{u}(0) = u_0$

$$
(\tilde{u}'(t), v)_{L^2(\Omega)} + a(\tilde{u}(t), v) = F(v) \quad \forall v \in V, \quad t \geq 0, \quad (13)
$$

where $\Omega = [0, 1]$ and $a(w, v) = \int_0^1 w'(x)v'(x) \, dx$.

Since it is a bit awkward to work with a function of one variable ($t$) which is a function of another ($x$), we often write (13) in terms of $u(x, t) = \tilde{u}(t)(x)$. 
Using subscript notation for partial derivatives, it becomes

\[(u_t(\cdot, t), v)_{L^2(\Omega)} + a(u(\cdot, t), v) = F(v) \quad \forall v \in V. \quad (14)\]

for all \(t\). If we remember the dependence on \(t\), we can write this as

\[(u_t, v)_{L^2(\Omega)} + a(u, v) = F(v) \quad \forall v \in V. \quad (15)\]

A stability estimate follows immediately from the variational formulation.

For simplicity, suppose that the right-hand-side form \(F \equiv 0\) and that the boundary data vanishes as well (i.e., only the initial data is non-zero).
Using $v = u$ (at fixed $t$, i.e., $v = u(\cdot, t)$) in (15), we find

$$\frac{1}{2} \frac{\partial}{\partial t} \|u\|_{L^2(\Omega)}^2 = (u_t, u)_{L^2(\Omega)} = -a(u, u) \leq 0 \quad \forall t \geq 0, \quad (16)$$

where $\Omega$ denotes the spatial interval $[0, 1]$. From this, it follows by integrating in time that

$$\|u(\cdot, t)\|_{L^2(\Omega)} \leq \|u(\cdot, 0)\|_{L^2(\Omega)} = \|u_0\|_{L^2(\Omega)} \quad \forall t \geq 0. \quad (17)$$

This result is independent of any compatibility conditions.

However, all it says is that the mean-square of the temperature $u$ remains bounded by its initial value.
If $F$ is nonzero but bounded on $V$, i.e.,

$$|F(v)| \leq \|F\|_{H^{-1}(\Omega)} \|v\|_{H^1(\Omega)} \quad \forall v \in V,$$

then we retain a bound on $\|u(\cdot, t)\|_{L^2(\Omega)}$:

$$\frac{1}{2} \frac{\partial}{\partial t} \|u\|^2_{L^2(\Omega)} = (u_t, u)_{L^2(\Omega)} = F(u) - a(u, u)$$

$$\leq \|F\|_{H^{-1}(\Omega)} \|u\|_{H^1(\Omega)} - a(u, u).$$

The form $a(\cdot, \cdot)$ always satisfies at least a weak type of coercivity of the form

$$\|v\|^2_{H^1(\Omega)} \leq \gamma_1 a(v, v) + \gamma_2 \|v\|^2_{L^2(\Omega)} \quad \forall v \in V,$$

known as Gårding’s inequality.
For example, this holds for the pure Neumann problem (7) with $V = H^1(\Omega)$ whereas the stronger form of coercivity we saw earlier does not in this case.

Applying (20) in (19) gives

$$\frac{\partial}{\partial t} \| u \|^2_{L^2(\Omega)} \leq 2 \| F \|_{H^{-1}(\Omega)} \| u \|_{H^1(\Omega)} - \frac{2}{\gamma_1} \| u \|^2_{H^1(\Omega)} + \frac{2\gamma_2}{\gamma_1} \| u \|^2_{L^2(\Omega)}. \quad (21)$$

The arithmetic-geometric mean inequality

$$2rs \leq \delta r^2 + \frac{1}{\delta} s^2 \quad (22)$$

holds for any $\delta > 0$ and any real numbers $r$ and $s$. 
Using the arithmetic-geometric mean inequality we find

\[ 2 \| F \|_{H^{-1}(\Omega)} \| u \|_{H^1(\Omega)} \leq \frac{\gamma_1}{2} \| F \|_{H^{-1}(\Omega)}^2 + \frac{2}{\gamma_1} \| u \|_{H^1(\Omega)}^2. \]

Thus

\[ \frac{\partial}{\partial t} \| u \|_{L^2(\Omega)}^2 \leq \frac{\gamma_1}{2} \| F \|_{H^{-1}(\Omega)}^2 + \frac{2\gamma_2}{\gamma_1} \| u \|_{L^2(\Omega)}^2 \]  

(23)

Gronwall’s Lemma [4] implies

\[ \| u(\cdot, t) \|_{L^2(\Omega)} \leq \| u_0 \|_{L^2(\Omega)} + e^{t(\gamma_2/\gamma_1)} \| F \|_{H^{-1}(\Omega)} \quad \forall t \geq 0. \]

(24)
Another stability result can be derived by using $v = u_t$ (assuming $F \equiv 0$ and the boundary data are zero) in (15), to find

$$\|u_t\|_{L^2(\Omega)}^2 = -a(u, u_t) = -\frac{1}{2} \frac{\partial}{\partial t} a(u, u).$$

(25)

From (25), it follows that

$$\frac{\partial}{\partial t} a(u, u) = -2 \|u_t\|_{L^2(\Omega)}^2 \leq 0 \quad \forall t \geq 0.$$

(26)

Again integrating in time and using (17), we see that

$$\|u(\cdot, t)\|_{H^1(\Omega)} \leq \|u(\cdot, 0)\|_{H^1(\Omega)} = \|u_0\|_{H^1(\Omega)} \quad \forall t \geq 0.$$
This result requires first-order compatibility conditions.

Presupposes that \( u_0 \in V \), and this may not hold.

Says mean-square of gradient of temperature \( u \) also remains bounded by its initial value.

Moreover, if the data \( F \) is not zero, this result will not hold.

In particular, if the compatibility condition (10) does not hold, then \( u_0 \not\in V \) and \( \|u(\cdot, t)\|_{H^1(\Omega)} \) will not remain bounded as \( t \to 0 \).
Simplest discretization uses a finite element method for spatial differential equation and a finite difference method for the temporal part.

Allows us to **reuse existing software** already developed for the spatial problem.

Many time dependent problems can be treated in the same manner. This technique goes by many names:

- (time) **splitting** since the time and space parts are separated and treated by independent methods
- the **method of lines** since the problem is solved on a sequence of lines (copies of the spatial domain), one for each time step.
The simplest time discretization method for the heat equation uses the forward (or explicit) Euler difference method. It takes the form

\[ u^{n+1}(x) = u^n(x) + \Delta t \frac{\partial^2 u^n}{\partial x^2}(x, t) \quad \forall x \in [0, 1], \]

\[ u^0(x) = u_0(x) \quad \forall x \in [0, 1] \]

\[ u^n(0) = g_0(t) \quad \text{and} \quad u^n(1) = g_1(t) \quad \forall n > 0 \]  \hspace{1cm} (28)

where \( u^n(x) \) denotes an approximation to \( u(x, n\Delta t) \).

Applying the finite difference or finite element approximation to (28) yields a simple algorithm.

For simplicity, we begin with the case that \( g_0 = g_1 = 0 \).
The difficulty with this simple algorithm is that it is \emph{unstable} unless $\Delta t$ is sufficiently small.

The simplest implicit time discretization method for the heat equation uses the backward (or implicit) Euler difference method. It takes the form

\[
\begin{align*}
    u^{n+1}(x) &= u^n(x) + \Delta t \frac{\partial^2 u^{n+1}}{\partial x^2}(x, t) \quad \forall x \in [0, 1], \\
    u^0(x) &= u_0(x) \quad \forall x \in [0, 1] \\
    u^n(0) &= g_0(t) \quad \text{and} \quad u^n(1) = g_1(t) \quad \forall n > 0
\end{align*}
\]

(29)

where $u^n(x)$ again denotes an approximation to $u(x, n\Delta t)$. 

\[
\begin{align*}
    u^{n+1}(x) &= u^n(x) + \Delta t \frac{\partial^2 u^{n+1}}{\partial x^2}(x, t) \quad \forall x \in [0, 1], \\
    u^0(x) &= u_0(x) \quad \forall x \in [0, 1] \\
    u^n(0) &= g_0(t) \quad \text{and} \quad u^n(1) = g_1(t) \quad \forall n > 0
\end{align*}
\]
Applying the finite difference or finite element approximation to (29) yields now a system of equations to be solved at each time step.

For simplicity, we begin with the case that \( g_0 = g_1 = 0 \).

This algorithm is \textit{stable} for all \( dt \), but now we have to solve a system of equations instead of just multiplying by a matrix.

Note however that the system to be solved is just the same as in the ODE boundary value problems studied earlier, so the same family of techniques can be used.
Explicit Euler time stepping method is written in variational form as

\[
(u^{n+1}, v)_{L^2(\Omega)} = (u^n, v)_{L^2(\Omega)} \\
+ \Delta t \left( F(v) - a(u^n, v) \right) \quad \forall v \in V.
\]  

Solving for \( u^{n+1} \) requires inverting the mass matrix.

Implicit Euler time stepping method is written in variational form as

\[
(u^{n+1}, v)_{L^2(\Omega)} + \Delta t \ a(u^{n+1}, v) = (u^n, v)_{L^2(\Omega)} \\
+ \Delta t \ F(v) \quad \forall v \in V.
\]
Solving for $u^{n+1}$ requires inverting linear combination of stiffness and the mass matrices. This is now in the familiar form: find $u^{n+1} \in V$ such that

$$a_{\Delta t}(u^{n+1}, v) = F_{\Delta t}^n(v) \quad \forall v \in V,$$

where

$$a_{\Delta t}(v, w) = \int_{\Omega} vw + \Delta t \, v'w' \, dx,$$

and

$$F_{\Delta t}^n(v) = (u^n, v)_{L^2(\Omega)} + \Delta t F(v) \quad \forall v, w \in V.$$
A popular way to achieve increased accuracy in time-dependent problems is to use a **backwards differentiation formula** (BDF)

\[
\frac{du}{dt}(t_n) \approx \frac{1}{\Delta t} \sum_{i=0}^{k} a_n u_{n-i},
\]

where the coefficients \( \{a_i : i = 0, \ldots k\} \) are given in Table 2.

<table>
<thead>
<tr>
<th>(k)</th>
<th>(a_0)</th>
<th>(a_1)</th>
<th>(a_2)</th>
<th>(a_3)</th>
<th>(a_4)</th>
<th>(a_5)</th>
<th>(a_6)</th>
<th>(a_7)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3/2</td>
<td>-2</td>
<td>1/2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>11/6</td>
<td>-3</td>
<td>3/2</td>
<td>-1/3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>25/12</td>
<td>-4</td>
<td>6/2</td>
<td>-4/3</td>
<td>1/4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>137/60</td>
<td>-5</td>
<td>10/2</td>
<td>-10/3</td>
<td>5/4</td>
<td>-1/5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>49/20</td>
<td>-6</td>
<td>15/2</td>
<td>-20/3</td>
<td>15/4</td>
<td>-6/5</td>
<td>1/6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>363/140</td>
<td>-7</td>
<td>21/2</td>
<td>-35/3</td>
<td>35/4</td>
<td>-21/5</td>
<td>7/6</td>
<td>-1/7</td>
</tr>
</tbody>
</table>

**Table 2**: Coefficients of the BDF schemes of degree \(k\).

The BDF for \(k = 1\) is the same as implicit Euler.
The BDF formulæ satisfy [3]

\[
\sum_{i=0}^{k} a_i u_{n-i} = \sum_{j=1}^{k} \frac{(-1)^j}{j!} \Delta^j u_n,
\]  

(34)

where \( \Delta u_n \) is the sequence whose \( n \)-th entry is \( u_n - u_{n-1} \).

The higher powers are defined by induction:

\( \Delta^{j+1} u_n = \Delta (\Delta^j u_n) \).

For example, \( \Delta^2 u_n = u_n - 2u_{n-1} + u_{n-2} \), and in general \( \Delta^j \) has coefficients given from Pascal’s triangle.
We thus see that $a_0 \neq 0$ for all $k \geq 1$; $a_0 = \sum_{i=1}^{k} 1/i$.

Similarly, $a_1 = -k$.

For $j \geq 2$, $ja_j$ is an integer conforming to Pascal’s triangle.

Given this simple definition of the general case of BDF, it is hard to imagine what could go wrong regarding stability.

Unfortunately, the BDF method of order $k = 7$ is unconditionally unstable and hence cannot be used.

Exercise: explore the use of BDF schemes.
The backwards heat equation

The heat equation is reversible with respect to time, in the sense that if we let time run backwards we get an equation that takes the final values to the initial values.

More precisely, let $u(x, t)$ be the solution to (5) for $0 \leq t \leq T$. Let $v(x, t) := u(x, T - t)$. Then $v$ solves the backwards heat equation

$$\frac{\partial v}{\partial t}(x, t) + \frac{\partial^2 v}{\partial x^2}(x, t) = 0 \quad \forall x \in [0, 1], \ t > 0$$

$$v(x, 0) = v_0(x) = u(x, T) \quad \forall x \in [0, 1]$$

$$v(0, t) = g_0(T - t), \quad v(1, t) = g_1(T - t) \quad \forall t > 0$$

and $v(x, T)$ will be the same as the initial data $u_0$ for (5).
With nonsmooth initial data, unreliable results occur.

Figure 4: Two solutions of the backwards heat equation with hat-function initial data at time $t = 0.001$, computed with piecewise linears with 50 mesh points (uniform mesh). (left) One time step with $\Delta t = 0.001$. (right) Two time steps with $\Delta t = 0.0005$. 
Although (35) has a well-defined solution in many cases, it is not well posed in the usual sense.

It only has a solution starting from solutions of the heat equation.

Moreover, such solutions may exist only for a short time, and then blow up.

Thus great care must be used in attempting to solve the backwards heat equation.

