Heat equation in 2D

Problem

Consider the following PDE:

$$\begin{aligned}
u_t - \Delta u &= f(\mathbf{x},t) = 2\pi\left[4\pi\cos(2\pi t) - \sin(2\pi t)\right]\sin(2\pi x)\sin(2\pi y), && \forall \mathbf{x} \in \Omega = (0,1)^2, && \text{(1a)}\\
u(\mathbf{x},0) &= u_0(\mathbf{x}) = \sin(2\pi x)\sin(2\pi y), && \forall \mathbf{x} \in \Omega = (0,1)^2, && \text{(1b)}\\
u(\mathbf{x},t) &= 0 && \forall \mathbf{x} \in \partial\Omega. && \text{(1c)}
\end{aligned}$$

Our objective is to obtain the solution at $t = 1$. The exact solution for this problem is given by

$$u = \cos(2\pi t)\sin(2\pi x)\sin(2\pi y).$$

Due to the stiff nature of the PDE, we use backward Euler to integrate the equation in time. By doing this, however, the overall accuracy is limited to first-order.

Weak formulation

Recall that $\Delta u = \nabla \cdot \nabla u$. Then, the weak formulation is given by

$$\int_\Omega u_t \varphi d\mathbf{x} + \int_\Omega \nabla u \cdot \nabla \varphi d\mathbf{x} + \underbrace{\int_{\partial\Omega} (\nabla u \cdot \mathbf{n})\varphi d\mathbf{x}}_{=0} = \int_\Omega f(\mathbf{x},t)\varphi d\mathbf{x}, \qquad \forall \varphi \in V, \tag{2}$$

where $V = \{v \in H^1(\Omega) : v = 0 \text{ in } \partial\Omega\}$.

Time discretization via backward Euler

Considering a discrete counterpart of (2) and using backward Euler yields

$$\int_\Omega \left(\frac{u_h^{n+1} - u_h^n}{\Delta t}\right)\varphi d\mathbf{x} + \int_\Omega \nabla u_h^{n+1} \cdot \nabla \varphi d\mathbf{x} = \int_\Omega f(\mathbf{x}, t^{n+1})\varphi.$$

Using the fact that $u_h(\mathbf{x}) = \sum_j U_j \varphi(\mathbf{x})$, we get

$$\sum_j U_j^{n+1} \underbrace{\int_\Omega (\varphi_i\varphi_j + \Delta t \nabla\varphi_i \cdot \nabla\varphi_j)\, d\mathbf{x}}_{= M_{ij} + \Delta t S_{ij}} = \underbrace{\int_\Omega \left[u_h^n + \Delta t f(\mathbf{x}, t^{n+1})\right]\varphi_i d\mathbf{x}}_{= r_i^{n+1}}$$

Therefore, we get

$$(M + \Delta t S)U^{n+1} = R^{n+1}.$$

An alternative implementation

Let's try to improve the performance of the implementation by avoiding FE loops at every time step. In the left hand side, the matrix $M + \Delta t S$ does not change, assuming Δt is constant. Even if Δt changes, we can compute M and S independently and add them together appropriately. For the right hand side, we can first let

$$f_h(\mathbf{x}, t^{n+1}) = \sum_j F_j^{n+1} \varphi_j, \quad \text{with} \quad F_j^{n+1} = f(\mathbf{x}_j, t^{n+1}),$$

and then perform the following approximation:

$$\int_\Omega f(\mathbf{x}, t^{n+1}) \varphi_i d\mathbf{x} \approx \int_\Omega f_h(\mathbf{x}, t^{n+1}) \varphi_i d\mathbf{x}.$$

Therefore, we get

$$\begin{aligned} r_i^{n+1} &= \int_\Omega \left[u_h^n + \Delta t f(\mathbf{x}, t^{n+1}) \right] \varphi_i d\mathbf{x} \approx \int_\Omega \left[u_h^n + \Delta t f_h(\mathbf{x}, t^{n+1}) \right] \varphi_i d\mathbf{x} \\ &= \sum_j \left(U_j^n + \Delta t F_j^{n+1} \right) \int_\Omega \varphi_i \varphi_j d\mathbf{x}. \end{aligned}$$

Finally, we obtain

$$(M + \Delta t S) U^{n+1} \approx M(U^n + \Delta t F^{n+1}).$$

Remark: we can only do this alternative implementation if the basis functions are interpolatory. By doing this we commit an error of $\mathcal{O}(h^{p+1})$, where p is the degree of the polynomial.

Numerical results