Solving 1-D Heat Equation and Burger's Equation using Finite Difference Method

1 1-D Heat Equation

We solve the 1-D heat equation

$$u_t(x,t) = vu_xx(x,t) + f(x,t) \qquad\qquad \forall [x,t] \in [0,1] \times [0,1]$$
$$u(0,t) = u(1,t) = 0 \qquad\qquad \forall t \in [0,1]$$
$$u(x,0) = u_0(x) \qquad\qquad \forall x \in [0,1]$$

by the implicit Euler's method.

First, we partition the interval of x using $m + 1$ equally spaced nodes and the interval of t using $n + 1$ equally spaced nodes. At time t_{j+1} we approximate u_t and u_{xx} using the 2-point backward finite difference formula and the 3-point centered finite difference formula, respectively. This gives

$$\frac{u_i^{j+1} - u_i^j}{\Delta t} - \nu \frac{u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}}{(\Delta x)^2} = f(x_i, t_{j+1})$$

$$u_{i-1}^{j+1}\left(-\frac{\nu \Delta t}{(\Delta x)^2}\right) + u_i^{j+1}\left(1 + \frac{2\nu \Delta t}{(\Delta t)^2}\right) + u_{i+1}^{j+1}\left(-\frac{\nu \Delta t}{(\Delta x)^2}\right) = u_i^j + \Delta t f(x_i, t_{j+1})$$

for $i = 0, 1, 2, ..., m$ and $j = 0, 1, 2, ..., n$.

Let $\alpha = -\frac{\nu \Delta t}{(\Delta x)^2}$ and $\gamma = 1 - 2\alpha$, the above system of equations can be written as

$$\begin{pmatrix} \alpha & 1-2\alpha & \alpha & & & \\ & \alpha & 1-2\alpha & \alpha & & \\ & & \ddots & \ddots & \ddots & \\ & & & \alpha & 1-2\alpha & \alpha \\ & & & & \alpha & 1-2\alpha & \alpha \end{pmatrix} \begin{pmatrix} u_0^{j+1} \\ u_1^{j+1} \\ \vdots \\ u_{m-1}^{j+1} \\ u_m^{j+1} \end{pmatrix} = \begin{pmatrix} u_0^j + \Delta t f(x_0, t_{j+1}) \\ u_1^j + \Delta t f(x_1, t_{j+1}) \\ \vdots \\ u_{m-1}^j + \Delta t f(x_{m-1}, t_{j+1}) \\ u_m^j + \Delta t f(x_m, t_{j+1}) \end{pmatrix}$$

To enforce Dirichlet boundary conditions at x_0 we replace the first row of the coefficient matrix with all zeros and one at the first position, and the first entry in the right hand side vector with the boundary condition. To enforce Dirichlet boundary conditions at x_m we replace the last row of the coefficient matrix with all zeros and one at the last position, and the last entry in the right hand side vector with the boundary condition. We eliminate the boundary conditions by substituting the values of u_0^{j+1} and u_m^{j+1} into the second and penultimate equations, respectively. This gives an $(m-1) \times (m-1)$ system of equations

$$\begin{pmatrix} 1-2\alpha & \alpha & 0 & & & \\ \alpha & 1-2\alpha & \alpha & & & \\ & \ddots & \ddots & \ddots & & \\ & & \alpha & 1-2\alpha & \alpha \\ & & & \alpha & 1-2\alpha \end{pmatrix} \begin{pmatrix} u_1^{j+1} \\ u_2^{j+1} \\ \vdots \\ u_{m-2}^{j+1} \\ u_{m-1}^{j+1} \end{pmatrix} = \begin{pmatrix} u_1^j + \Delta t f(x_1, t_{j+1}) - \alpha BC1(t_{j+1}) \\ u_2^j + \Delta t f(x_2, t_{j+1}) \\ \vdots \\ u_{m-2}^j + \Delta t f(x_{m-2}, t_{j+1}) - \alpha BC2(t_{j+1}) \\ u_{m-1}^j + \Delta t f(x_{m-1}, t_{j+1}) \end{pmatrix}$$

where $BC1$ and $BC2$ are the boundary conditions at x_0 and x_m, respectively.

At each time t_{j+1} we solve the above system to get the solution vector at time t_{j+1}, for $j = 0, 1, 2, ..., n$. This is implemented in `heateq.m`.

The method is implicit and unconditionally stable. For $\Delta t = 0.25$ and $\Delta x = 0.2$, the approximation is within two accurate digits from the exact solution.

1.1   Validation problem

Here we construct the validation problem. Let the solution $u(x, t) = e^{-t} \sin(\pi x)$. Note that this solution satisfies the specified boundary conditions. Then

$$u(x, 0) = \sin(\pi x)$$

$$u_t = -\sin(\pi x)e^{-t} \text{ and}$$

$$u_x x = -(\pi)^2 \sin(\pi x)$$

Substituting in the original heat equation gives

$$f(x,t) = -\sin(\pi x)e^{-t} + \nu e^{-t}\sin(\pi x)$$

The constructed problem

$$u_t - \nu u_x x = -\sin(\pi x)e^{-t} + \nu e^{-t}\sin(\pi x) \qquad \forall x \in [0,1] \text{ and} \qquad \forall t \in [0,1]$$

$$u(0,t) = u(1,t) = 0 \qquad \forall t \in [0,1]$$

$$u(x,0) = \sin(\pi x) \qquad \forall x \in [0,1]$$

has exact solution $u(x,t) = e^{-t}\sin(\pi x)$.

1.2 Convergence rates

We determine numerically the order of convergence for the local (truncation) error. The local (truncation) error is the error made at the discretization using Δx. The local (truncation) error at a fixed time t^* in x is defined as

$$\epsilon = \max_{1 \le j \le m} |u(x_j) - u_j| = C\Delta x^p = O(\Delta x^p)$$

where p is the order of convergence. p is determined as follows

$$\log_{10}\epsilon = \log_{10} C\Delta x^p = \log_{10} C + p\log_{10}\Delta t$$

We compute p over several grids with Δx decreasing by a factor of $1/2$ at each grid. Then p is

$$p = \frac{log_{10}\epsilon_{\Delta x} - \log_{10}\epsilon_{\Delta x/2}}{\log_{10}\Delta x - \log_{10}(\Delta x/2)} = \frac{log_{10}\epsilon_{\Delta x} - \log_{10}\epsilon_{\Delta x/2}}{\log_{10} 2}$$

We use $t^* = t_{final} = 0$ to eliminate the need of determining exact index $j+1$ for t and simplify the code. For the error, we use the infinity norm of the error vector at time $t^* = 1$.

Convergence in t is determined analogously.

Convergence analysis is carried out in `convHeateq.m`.

2 Burger equation

Here we want to solve the 1-D Burger's equation

$$u_t(x,t) = \nu u_x x(x,t) - u(x,t)u_x(x,t) + f(x,t) \qquad \forall [x,t] \in [0,1] \times [0,1]$$

$$u(0,t) = u(1,t) = 0 \qquad \forall t \in [0,1]$$

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

using backward Euler's method.

We partition the interval of x using $m + 1$ equally spaced nodes and the interval of t using $n + 1$ equally spaced nodes.

At time t_{j+1} we approximate u_t, u_x, and u_{xx} using the 2-point backward finite difference formula, 2-point centered finite difference formula, and the 3-point centered finite difference formula, respectively. This gives

$$\frac{u_i^{j+1} - u_i^j}{\Delta t} + u_i^{j+1} \frac{u_{i+1}^{j+1} - u_{i-1}^{j+1}}{2\Delta x} - \nu \frac{u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}}{(\Delta x)^2} = f(x_i, t_{j+1})$$

Letting $\alpha = \frac{\Delta t}{2\Delta x}$ and $\gamma = \frac{\nu \Delta t}{(\Delta x)^2}$, and grouping like terms, gives

$$u_{i-1}^{j+1} \left(-\alpha u_i^{j+1} - \gamma\right) + u_i^{j+1} (1 + \gamma) + u_{i+1}^{j+1} \left(\alpha u_i^{j+1} - \gamma\right) - u_i^j - \Delta t f(x_i, t_{j+1}) = 0$$

for $i = 0, 1, 2, ..., m$ and $j = 0, 1, 2, ..., n$.

To enforce Dirichlet boundary conditions at x_0 we replace the first equation by $u_0^{j+1} - BC1 = 0$; to enforce Dirichlet boundary conditions at x_m we replace the last equation by $u_m^{j+1} - BC2 = 0$, where $BC1$ and $BC2$ are the boundary conditions at x_0 and x_m, respectively.

The above system of equation is non-linear. For more robust approximations, we use Matlab's built-in `fsolve`. At each time t_{j+1} we solve the above system to get the solution vector at time t_{j+1}, for $j = 0, 1, 2, ..., n$. This is implemented in `iburger.m`.

The method used in the code is implicit and unconditionally stable. For $\Delta t = 0.25$ and $\Delta x = 0.2$, the approximation is within 2 accurate digits from the exact solution.

Convergence analysis is similar to that of the 1-D Heat equation and is carried out in `convIburger.m`.

2.1 Validation problem

Here we construct the validation problem. Let the solution $u(x, t) = e^{-t} \sin(\pi x)$. Note that this solution satisfies the specified boundary conditions. Then

$$u(x, 0) = \sin(\pi x)$$
$$u_t = -\sin(\pi x)e^{-t}$$
$$u_x = \pi e^{-t} \cos(\pi x)$$
$$u_{xx} = -(\pi)^2 \sin(\pi x)$$

Substituting in the original Burger's equation gives

$$f(x, t) = -e^{-t} \sin(\pi x) + \pi e^{-2t} \sin(\pi x) \cos(\pi x) + \nu \pi^2 e^{-t} \sin(\pi x)$$

The constructed problem

$$u_t + uu_x - \nu u_{xx} = -e^{-t}\sin(\pi x) + \pi e^{-2t}\sin(\pi x)\cos(\pi x) + \nu\pi^2 e^{-t}\sin(\pi x) \quad \forall x \in [0,1] \text{ and } \forall t \in [0,1]$$

$$u(0,t) = u(1,t) = 0 \qquad\qquad\qquad\qquad\qquad \forall t \in [0,1]$$

$$u(x,0) = \sin(\pi x) \qquad\qquad\qquad\qquad\qquad \forall x \in [0,1]$$

has exact solution $u(x,t) = e^{-t}\sin(\pi x)$.