

1.

- Make a graph of the boundary of the absolute stability region for the Runge-Kutta RK4 method on page 24.
- Apply the RK4 method to $u' = 30u(1 - u)$ with $u(0) = 0.1$ on the interval $0 \leq t \leq 2$. Use 14, 17 and 25 steps. For each run graph the numerical solution and the exact solution on the same plot.
- Explain the previous sequence of graphs in terms of the ODE and the plot from part a. Your answer should contain a quantitative explanation for why the transition occurs at the value of h you observe.

2. The θ -method for finite difference solutions of ODEs is

$$u_{n+1} = u_n + h[\theta f_{n+1} + (1 - \theta)f_n]$$

where $\theta \in [0, 1]$.

- Compute the order of convergence of this method. Hint: it depends on θ .
- Numerically graph the boundary of the absolute stability region for $\theta = 1/4$.
- Show that the method is A -stable if and only if $\theta \geq 1/2$.
- Bonus: Analytically compute the boundary of the absolute stability region for $\theta = 1/4$.

3. Newton's method can be used to solve

$$f(x) = 0$$

where $x \in \mathbb{R}^n$ and $f(x) \in \mathbb{R}^n$. Starting from an initial guess x_k ,

$$x_{k+1} = x_k - Df(x_k)^{-1}f(x_k).$$

Here, $Df(x)$ is the Jacobian matrix

$$Df_{ij} = \frac{\partial f_i}{\partial x_j}$$

Implement Newton's method for systems. Your function should take as arguments f , Df and x_0 (an initial guess). It should terminate whenever either

- $|f(x)|_\infty$ is less than a specified tolerance
- $|f(x)|_\infty$ is less than a specified fraction of $|f(x_0)|_\infty$

These tolerances should be specified with optional arguments as used in your language of choice.

Test your code against TBA.

4. The energy for the heat equation $u_t = u_{xx}$ for $0 \leq x \leq 1$ is

$$E(t) = \frac{1}{2} \int_0^1 (u_x(x, t))^2 dx.$$

- a) Assuming that at $x = 0$ and at $x = 1$ u satisfies either a homogeneous Dirichlet condition or a homogeneous Neumann condition, show that

$$\frac{d}{dt}E(t) \leq 0.$$

Hint: Take a time derivative, use the PDE, and integrate by parts.

- b) Conclude that the only solution of $u_t = u_{xx}$ with $u = 0$ at $t = 0$, and at $x = 0$ and $x = 1$ is the zero solution.

5. The backwards heat equation reads

$$u_t = -u_{xx},$$

so all that differs is a sign on the right-hand side. But this sign makes all the difference.

We will work with this equation for $0 \leq x \leq 1$ and $0 \leq t \leq 1$, and with homogeneous Dirichlet boundary conditions, so $u = 0$ at $x = 0$ and $x = 1$.

- a) Show that

$$v(t) = \sin(k\pi x)e^{k^2\pi^2 t}$$

is a solution of the PDE and the boundary conditions.

- b) For each $\epsilon > 0$, find a solution of the PDE and boundary conditions that satisfies $|u(0, x)| < \epsilon$ at each x , but $|u(1, x)| \geq 1$ at some x .
- c) Suppose you wish to find the solution u of the backwards heat equation with initial condition u_0 . But you don't know u_0 exactly, you know \hat{u}_0 , and that $|u_0(x) - \hat{u}_0(x)| < 10^{-47}$ at every x . So you solve the backwards heat equation for \hat{u} instead. Find an L such that $|u(x, 1) - \hat{u}(x, 1)| < L$ for all x , or explain why no such L exists.

6. Implement the explicit method for solving the heat equation with right-hand side function

$$u_t = u_{xx} + f$$

on $0 \leq x \leq 1$ and $0 \leq t \leq T$. Your function should have the following signature:

`forcedheat(f,u0,N,M)`

where

- $f(x, t)$ is a function and provides the desired forcing term
- $u_0(x)$ is a function and provides the desired initial condition.
- $N + 1$ is the number of interior spatial steps
- M is the number of time steps

It should return (x, t, u) where x is an array of grid coordinates that includes 0 and 1, t is a vector of t coordinates that includes 0 and T , and where u is an $(N + 2) \times (M + 1)$ matrix where column j encodes the solution at time t_j .

Test your code as follows

- Compute what f is if the solution is $u(t, x) = \sin(t)x(1 - x)$.
- Now, working on $0 \leq x \leq 1$ and $0 \leq t \leq 2\pi$ compute solutions with this forcing term and compare your solution with the exact solution. By working with various grid sizes, confirm that your code has the expected order of convergence.