

Scientific Computation Homework 6

Michael Nameika

October 2022

Exercise 9.1

Consider the following method for solving the heat equation $u_t = u_{xx}$:

$$U_i^{n+2} = U_i^n + \frac{2k}{h^2}(U_{i-1}^{n+1} - 2U_i^{n+1} + U_{i+1}^{n+1}).$$

- (a) Determine the order of accuracy of this method (in both space and time).

To begin, let us rearrange the method into the following form:

$$\frac{U_i^{n+2} - U_i^n}{2k} = \frac{U_{i-1}^{n+1} - 2U_i^{n+1} + U_{i+1}^{n+1}}{h^2}$$

Writing the above equation in exact form, we have

$$\frac{u_i^{n+2} - u_i^n}{2k} = \frac{u_{i-1}^{n+1} - 2u_i^{n+1} + u_{i+1}^{n+1}}{h^2}$$

Let us begin by Taylor expanding the left hand side. Notice

$$u_i^{n+2} = u_i^n + 2k \frac{\partial u_i^n}{\partial t} + 2k^2 \frac{\partial^2 u_i^n}{\partial t^2} + \frac{4}{3} k^3 \frac{\partial^3 u_i^n}{\partial t^3} + \mathcal{O}(k^4)$$

Then

$$\frac{u_i^{n+2} - u_i^n}{2k} = \frac{\partial u_i^n}{\partial t} + k \frac{\partial^2 u_i^n}{\partial t^2} + \frac{2}{3} k^2 \frac{\partial^3 u_i^n}{\partial t^3} + \mathcal{O}(k^4)$$

Now let us Taylor expand the right hand side:

$$\begin{aligned} u_{i-1}^{n+1} &= u_i^{n+1} - h \frac{\partial u_i^{n+1}}{\partial x} + \frac{h^2}{2} \frac{\partial^2 u_i^{n+1}}{\partial x^2} - \frac{h^3}{3!} \frac{\partial^3 u_i^{n+1}}{\partial x^3} + \frac{h^4}{4!} \frac{\partial^4 u_i^{n+1}}{\partial x^4} - \frac{h^5}{5!} \frac{\partial^5 u_i^{n+1}}{\partial x^5} + \mathcal{O}(h^6) \\ u_{i+1}^{n+1} &= u_i^{n+1} + h \frac{\partial u_i^{n+1}}{\partial x} + \frac{h^2}{2} \frac{\partial^2 u_i^{n+1}}{\partial x^2} + \frac{h^3}{3!} \frac{\partial^3 u_i^{n+1}}{\partial x^3} + \frac{h^4}{4!} \frac{\partial^4 u_i^{n+1}}{\partial x^4} + \frac{h^5}{5!} \frac{\partial^5 u_i^{n+1}}{\partial x^5} + \mathcal{O}(h^6) \end{aligned}$$

Then

$$\frac{u_{i-1}^{n+1} - 2u_i^{n+1} + u_{i+1}^{n+1}}{h^2} = \frac{\partial^2 u_i^{n+1}}{\partial x^2} + \frac{2}{4!} h^2 \frac{\partial^4 u_i^{n+1}}{\partial x^4} + \mathcal{O}(h^4)$$

And so our local truncation error takes the following form:

$$\tau(x, t) = \frac{\partial u_i^n}{\partial t} + k \frac{\partial^2 u_i^n}{\partial t^2} + \frac{2}{3} k^2 \frac{\partial^3 u_i^n}{\partial t^3} - \frac{\partial^2 u_i^{n+1}}{\partial x^2} + \mathcal{O}(h^2) + \mathcal{O}(k^3)$$

To simplify our local truncation error any further, we must Taylor expand u_i^{n+1} :

$$u_i^{n+1} = u_i^n + k \frac{\partial u_i^n}{\partial t} + \frac{k^2}{2} \frac{\partial^2 u_i^n}{\partial t^2} + \frac{k^3}{3!} \frac{\partial^3 u_i^n}{\partial t^3} + \mathcal{O}(k^4)$$

Using this in our equation for the local truncation error leads us to

$$\tau(x, t) = \frac{\partial u_i^n}{\partial t} + k \frac{\partial^2 u_i^n}{\partial t^2} + \frac{2}{3} k^2 \frac{\partial^3 u_i^n}{\partial t^3} - \frac{\partial^2 u_i^n}{\partial x^2} - k \frac{\partial^2}{\partial x^2} \left(\frac{\partial u_i^n}{\partial t} \right) - \frac{k^2}{2} \frac{\partial^2}{\partial x^2} \left(\frac{\partial^2 u_i^n}{\partial t^2} \right) - \frac{k^3}{3!} \frac{\partial^2}{\partial x^2} \left(\frac{\partial^3 u_i^n}{\partial t^3} \right) + \mathcal{O}(k^4 + h^2)$$

From our differential equation, we have $\frac{\partial u_i^n}{\partial t} = \frac{\partial^2 u_i^n}{\partial x^2}$, so our local truncation error becomes

$$\begin{aligned} \tau(x, t) &= \frac{\partial u_i^n}{\partial t} + k \frac{\partial^2 u_i^n}{\partial t^2} + \frac{2}{3} k^2 \frac{\partial^3 u_i^n}{\partial t^3} - \frac{\partial u_i^n}{\partial t} - k \frac{\partial^2 u_i^n}{\partial t^2} - \frac{k^2}{2} \frac{\partial^3 u_i^n}{\partial t^3} - \frac{k^3}{3!} \frac{\partial^4 u_i^n}{\partial t^4} + \mathcal{O}(k^4 + h^2) \\ &= \frac{k^2}{6} \frac{\partial^3 u_i^n}{\partial t^3} + \mathcal{O}(k^3 + h^2) \end{aligned}$$

So

$$\tau(x, t) = \mathcal{O}(k^2 + h^2)$$

That is, this method is second order accurate in both space and time.

- (b) Suppose we take $k = \alpha h^2$ for some fixed $\alpha > 0$ and refine the grid. For what values of α (if any) will this method be Lax-Richtmyer stable and hence convergent?

Hint: Consider the MOL interpretation and the stability region of the time-discretization being used. We require $|\lambda| \leq 1$ for stability. Additionally, we have $|1 - k/(2h^2)\lambda| \leq 1$ which leads us to

$$\begin{aligned} \frac{k}{2h^2} &\leq 2 \\ \alpha &\leq 4 \end{aligned}$$

- (c) Is this a useful method?

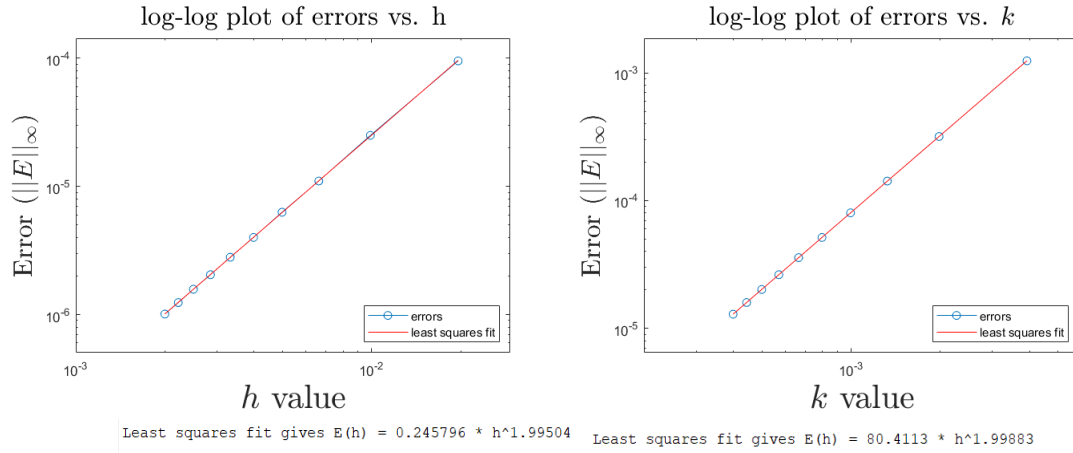
It's second order accurate in both space and time, so why not?

Exercise 9.2

- (a) The m-file `heat_CN.m` solves the heat equation $u_t = \kappa u_{xx}$ using the Crank-Nicolson method. Run this code, and by changing the number of grid points, confirm that it is second-order accurate. (Observe how the error at some fixed time such as $T = 1$ behaves as k and h go to zero with a fixed relation between k and h , such as $k = 4h$.)

You might want to use the function `error_table.m` to print out this table and estimate the order of accuracy, and `error_loglog.m` to produce a log-log plot of the error vs. h . See `bvp_2.m` for an example of how these are used

Running `heat_CN.m` for decreasing values of h , we find the following:



Which shows us that the Crank-Nicolson method is second order accurate in both space and time.

- (b) Modify `heat_CN.m` to produce a new m-file `heat_trbdf2.m` that implements the TR-BDF2 method on the same problem. Test it to confirm that it is also second order accurate. Explain how you determined the proper boundary conditions in each stage of this Runge-Kutta method.

Recall that the TR-BDF2 method has the following form:

$$U^* = U^n + \frac{\Delta t}{4}(f(U^n) + f(U^*))$$

$$U^{n+1} = \frac{1}{3}(4U^* - U^n + f(U^{n+1}))$$

Applying this method to $u_{tt} = \kappa u_{xx}$, we will find the following for U^* :

$$U^* = U^n + \frac{\Delta t}{2} \left(\frac{f(U^n) + f(U^*)}{2} \right)$$

Which shows that U^* is a half step increment in time, and using the method of lines (MOL) to rewrite as discrete system, we have

$$u_i^* = u_i^n + \kappa \frac{\Delta t}{2\Delta x^2} \left(\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{2} + \frac{u_{i+1}^{n+1/2} - 2u_i^{n+1/2} + u_{i-1}^{n+1/2}}{2} \right)$$

and for $i = 1$,

$$u_1^* = u_1^n + \kappa \frac{\Delta t}{2\Delta x^2} \left(\frac{u_2^n - 2u_1^n + u_0^n}{2} + \frac{u_2^{n+1/2} - 2u_1^{n+1/2} + u_0^{n+1/2}}{2} \right)$$

Writing this step in matrix notation, we have

$$A^*U^* = A_RU^n + G_1^n$$

where

$$A^* = \begin{bmatrix} 1+2r & -r & & & \\ -r & 1+2r & -r & & \\ & \ddots & \ddots & \ddots & \\ & & -r & 1+2r & -r \\ & & & -r & 1+2r \end{bmatrix}$$

$$A_R = \begin{bmatrix} 1-2r & r & & & \\ r & 1-2r & r & & \\ & \ddots & \ddots & \ddots & \\ & & r & 1-2r & r \\ & & & r & 1-2r \end{bmatrix}$$

$$G_1^n = \begin{bmatrix} g_0^n + g_0^{n+1/2} \\ 0 \\ \vdots \\ 0 \\ g_1^n + g_1^{n+1/2} \end{bmatrix}$$

and

$$r = \frac{\kappa \Delta t}{4\Delta x^2}$$

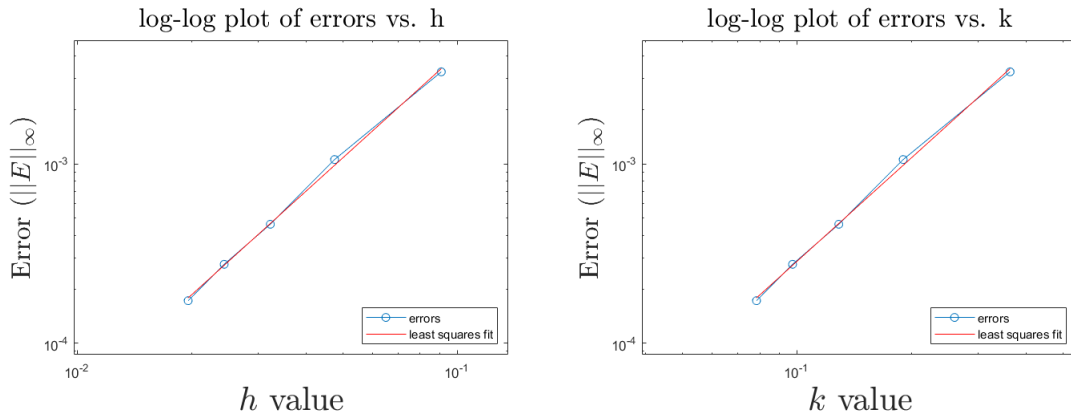
Now, to for the implicit step, we can see, similar to the work above, that

$$A_L U^{n+1} = \frac{4}{3} U^* - \frac{1}{3} U^n$$

where

$$A_L = \begin{bmatrix} 1+4/3r & -4/3r & & & \\ -4/3r & 1+4/3r & -4/3r & & \\ & \ddots & \ddots & \ddots & \\ & & -4/3r & 1+4/3r & -4/3r \\ & & & -4/3r & 1+4/3r \end{bmatrix}$$

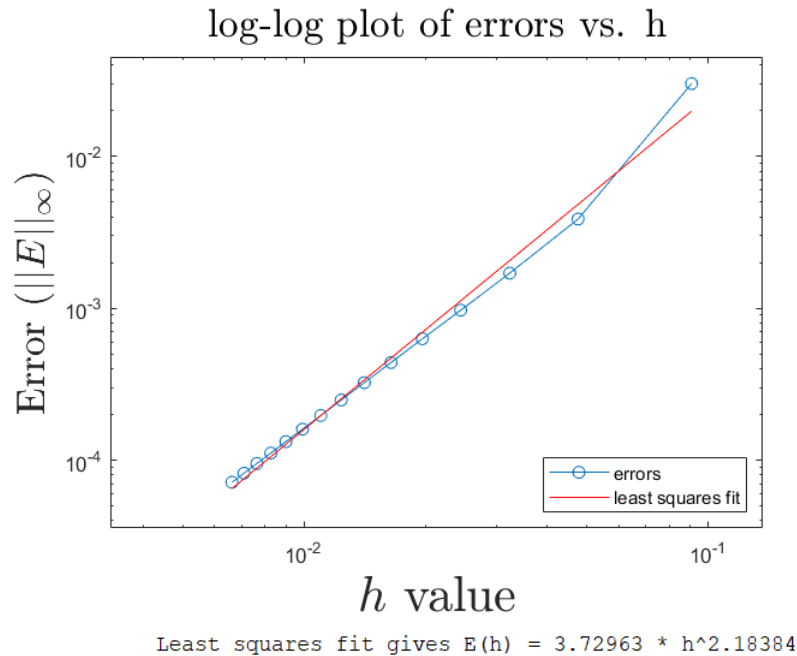
Implementing this method by modifying `heat_CN.m` (see attached `.m` file), we find the following for the order of accuracy:



So we can see the TR-BDF2 method is second order accurate in both space and time.

- (c) Modify `heat_CN.m` to produce a new m-file `heat_FE.m` that implements the forward Euler explicit method on the same problem. Test it to confirm that it is $\mathcal{O}(h^2)$ accurate as $h \rightarrow 0$ provided when $k = 24h^2$ is used, which is within the stability limit for $\kappa = 0.02$. Note how many more time steps are required than with Crank-Nicolson or TR-BDF2, especially on finer grids.

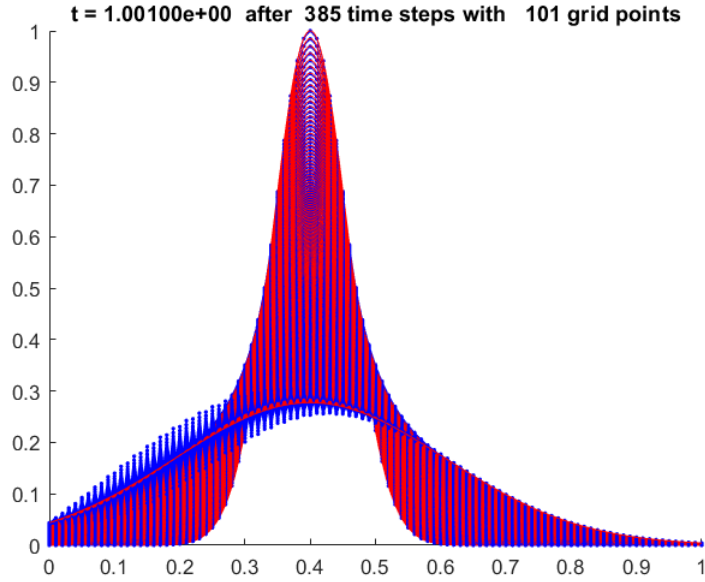
Modifying `heat_CN.m` to implement the Forward-Euler method, and using $k = 24h^2$, we find the following for smaller values of h :



which shows us that the method is approximately second order accurate in space.

- (d) Test `heat_FE.m` with $k = 26h^2$, for which it should be unstable. Note that the instability does not become apparent until about time 1.6 for the parameter values $\kappa = 0.02$, $m = 39$, $\beta = 150$. Explain why the instability takes several undred time steps to appear, and why it appears as a sawtooth oscillation.

Changing $k = 26h^2$, we find the following plot for the solutions in time using the Forward-Euler method:



Notice how the solutions at later time steps have this 'sawtooth' instability. Curiously, using $m = 39$ did not show this instability and the instabilities did not show up until about $t = 0.9$.

Exercise 9.3

- (a) Modify `heat_CN.m` to solve the heat equation for $-1 \leq x \leq 1$ with step function initial data

$$u(x, 0) = \begin{cases} 1 & \text{if } x < 0 \\ 0 & \text{if } x \geq 0. \end{cases}$$

With appropriate Dirichlet boundary conditions, the exact solution is

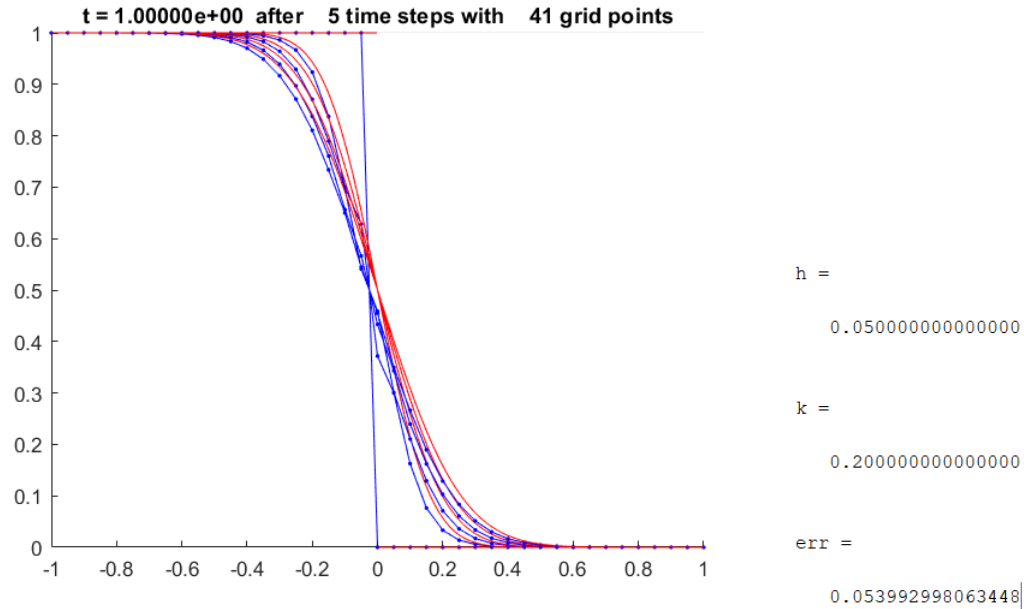
$$u(x, t) = \frac{1}{2} \operatorname{erfc}(x/\sqrt{4\kappa t}),$$

where `erfc` is the complementary error function

$$\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-z^2} dz.$$

- (i) Test this routine $m = 39$ and $k = 4h$. Note that there is an initial rapid transient decay of the high wave numbers that is not captured well with this size time step.

Modifying the `heat_CN.m` (see attached code), we find the following for $m = 39$ and $k = 4h$:

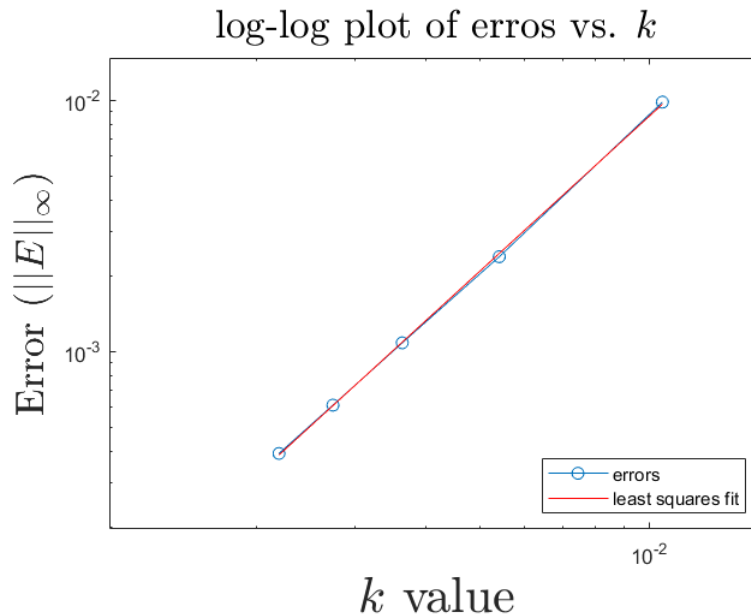


- (ii) How small do you need to take the time step to get reasonable results? For a suitably small time step, explain why you get much better results by using $m = 38$ than $m = 39$. What is the observed order of accuracy as $k \rightarrow 0$ when $k = \alpha h$ with α suitably small and m even?

Presumably, it would be reasonable to assume that $k \leq 1/2$ to get reasonable results since, if $k > 1/2$, we would have only one time step, which would mean only the initial data will be approximated. Having $k \leq 1/2$ will give us at least one approximation that will allow us to see the continuous solution.

We get much better results by using $m = 38$ rather than $m = 39$ since for odd values of m , we will be including the midpoint discontinuity in the approximation, which will lead to the approximation "undershooting" the actual approximation.

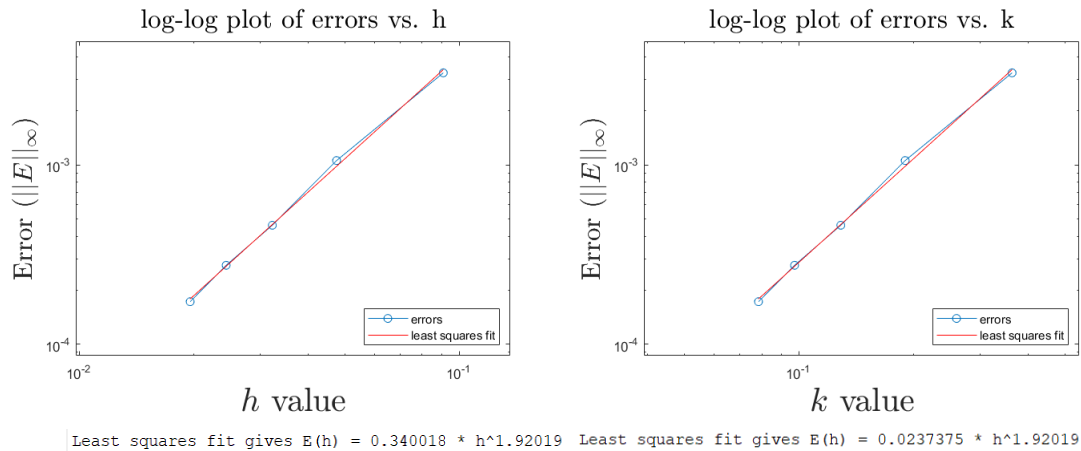
Running the program for $\alpha = 1/10$, we find the following for the order of accuracy in k :



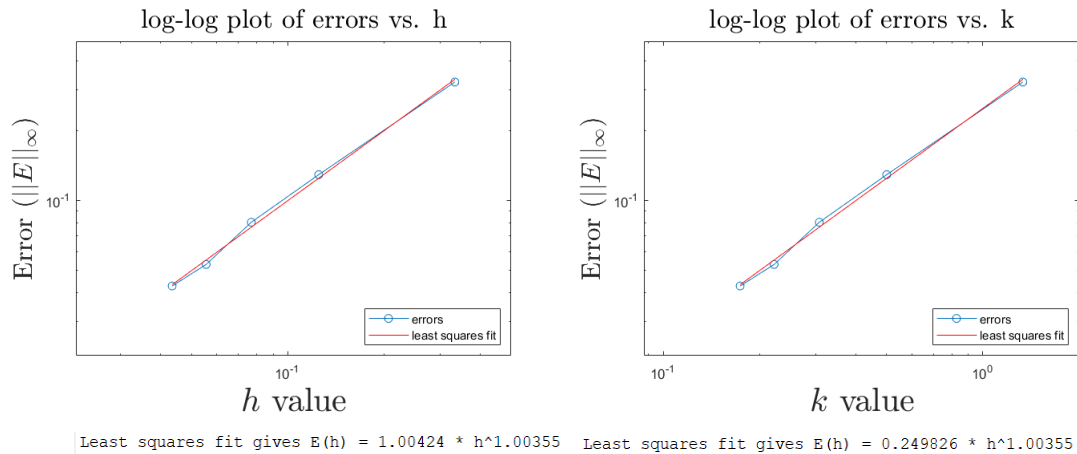
Least squares fit gives $E(h) = 111.319 * h^{2.05363}$

- (b) Modify `heat_trbdf2.m` (see Exercise 9.2) to solve the heat equation for $-1 \leq x \leq 1$ with step function initial data as above. Test this routine using $k = 4h$ and estimate the order of accuracy as $k \rightarrow 0$ with m even. Why does the TR-BDF2 method work better than Crank-Nicolson?

Modifying the `heat_trbdf2.m` we used earlier in the assignment, we find the following for an even number of spacial points:

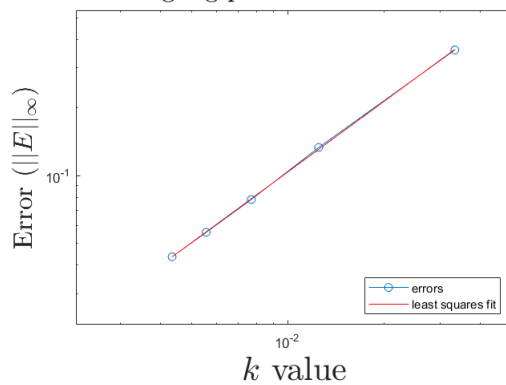


and for an odd number of spacial points:



Which turns out to be the same order that we find for the the Crank-Nicolson method (odd number of spacial points):

log-log plot of errors vs. k



log-log plot of errors vs. h

