Computer Lab Assignment 7

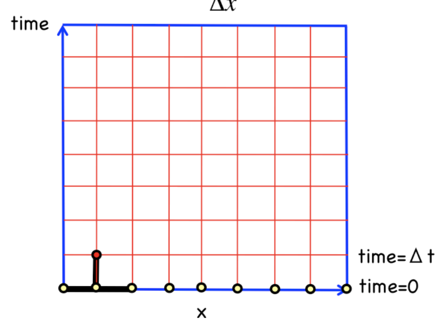
Solving the 1D Heat Equation in Real Time

No Matlab code will be provided for this computer lab. Instead we ask you to build on your files from lab and homework 6, or create it all from scratch. The plan for this lab is to construct a real-time solution for a 1D cooling problem. First review the analytical and the discretized form of the 1D heat equation given in following lecture slide:

Explicit Scheme to Solve the 1D Heat Equation

$$\boxed{T(x,t)} \qquad \frac{\partial T}{\partial t} = k \frac{\partial^2 T}{\partial x^2}$$

$$T(x,t+\Delta t) = \frac{\Delta t\ k}{\Delta x^2}\left[T(x+\Delta x,t) - 2T(x,t) + T(x-\Delta x,t)\right] + T(x,t)$$

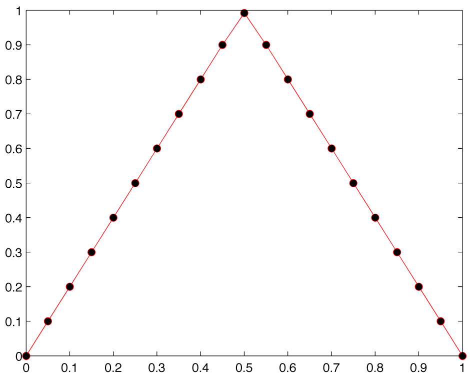


(1) The discretized form needs to be transformed yet again to use the index i that Matlab can handle instead of the variable x. For this exercise, we will use a 1D array `T(i)` to present $T(x,t)$ and `Tnew(i)` to present $T(x,t+\Delta t)$. (One could also introduce an index j to present time but this is not done here to save memory.) Now fill in the blanks:

$T(x,t)$	$\rightarrow$	`T(i)`
$T(x,t+\Delta t)$	$\rightarrow$	`Tnew(i)`
$T(x+\Delta x,t)$	$\rightarrow$	?
$T(x-\Delta x,t)$	$\rightarrow$	?

and write down the resulting equation for `Tnew(i)`:

(2) As next step, we need to prepare the initial conditions, `T(i)`. Discretize the interval x=[0,L] in N sections requiring N+1 points. Set L=1.0 and N=20 and reproduce the following, somewhat artificial, initial temperature distribution:



Issue a plot command that reproduces the picture above. Please make sure that *T(x=0)* and *T(x=1)* are indeed set to 0, and that *T(x=0.5)* equals 1.

(3) Set the heat conduction coefficient *k=1* and assume *T(x=0)= T(x=1)=0* as boundary conditions. Now write a Matlab loop that computes `Tnew(i)` from `T(i)` using the equation from step (1). Assume a time step of 10^{-4}. Then copy `Tnew` to back `T` and plot the temperature distribution for this step.

(4) Now program an additional Matlab loop that repeats step (3) many times. You may not want to plot the temperature distribution at every step but when you do then please add the commands `axis([0 L 0 1]); drawnow; pause;` Run your code for a large number of iterations until less than 1% of the initial thermal energy is left.

- How much time did this approximately take?
- Where did the heat go?
- Why does the heat distribution get smoother and smoother?

Well done if you see the heat disappear! You completed the most time consuming part.

(5) Now we pretend to be a bit impatient and want to run with a larger time step. Please increase Δt step by step until you see a drastic change in the behavior of the resulting solution. What happens? Then carefully determine the *critical* time step, Δt_C, where this change occurs.

(6) Now increase the spatial resolution by cutting Δx in half and determine the new critical time step. Then reduce Δx again by 50% and redetermine the critical time step.

(7) As we will see in lecture, it does not make sense to run $\Delta t > \Delta t_C$. However it is not clear by how much the solution depends on the choice of $\Delta t < \Delta t_C$. So let us design an experiment and assume that the solution will become better and better as we decrease Δt. To compare runs with different Δt, I suggest you plot the temperature at the midpoint, x=0.5, as a function of time. Please make a plot that compares the midpoint temperatures for three time steps that differ by a factor of 10.