We propose to develop a user-friendly software to solve a specific heat equation Ut = Uxx at certain user-specified conditions, including dimensions of the equation, numerical methods, boundary conditions and order of accuracy etc. One of the applications of this software is that it can serve as a helpful demo tool in engineering PED class for the lecturer and the students.

**One-dimensional heat equation**

**1) numerical schemes**

The most straightforward part is one-dimensional problem. A series of numerical methods will be available for the user to choose from, including Forward Euler method, Backward Euler method, Crank-Nicolson method and Dufort-Frankel method.

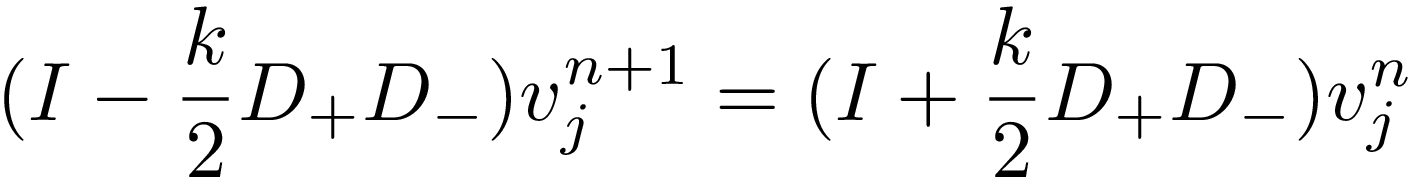
Forward Euler method:



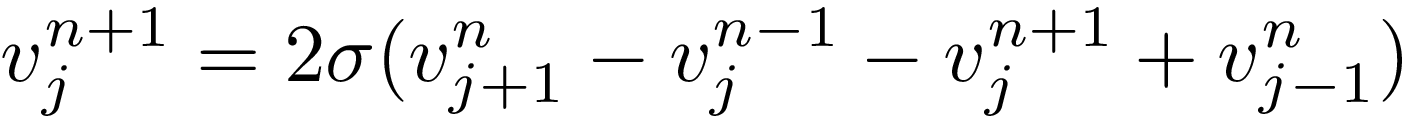
Backward Euler method:



Crank-Nicolson method:



Dufort-Frankel method:



**2) Initial conditions**

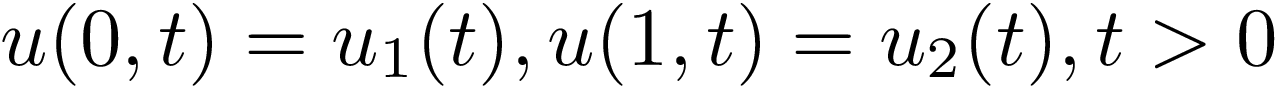
The initial condition of heat equation is the initial temperature distribution at t = 0, thus, the initial condition is given as:



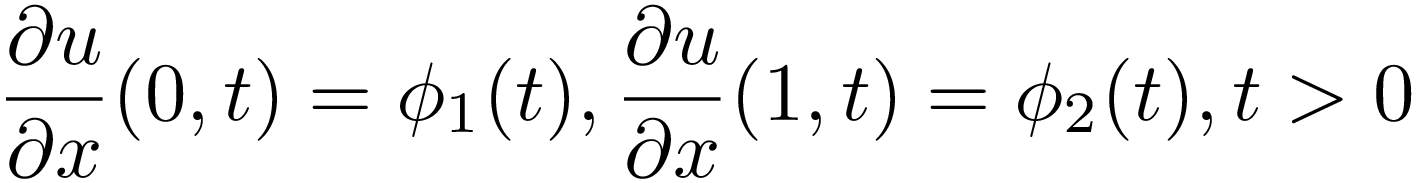
**3) boundary conditions**

Another feature is that the software is able to solve the heat equation with certain kinds of boundaries conditions, including I) temperature prescribed at ends, II) insulated boundary conditions and III) mixed boundary conditions.

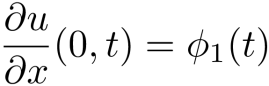
I) temperature prescribed at ends:



II) insulated boundary condition:



III) mixed boundary conditions: a equation involving both first kind and second kind of boundary conditions:

**4) discretization of spatial and time domains and CFL conditions**

The user needs to give input from interface for the space step h and time step k for the discretization of the spatial and time domains. The number of nodes can be given as N = 1/h, and the number of time steps is M = time/k, where time is User specified time segment.

By giving the spatial and the time step, the corresponding CFL (Courant-Friedrichs-Lewy) condition for the specified numerical scheme would be output for the user to decide whether to proceed on or not. Basically, the CFL condition is a necessary condition for the convergence of a finite difference method for a hyperbolic PDE, that is, the numerical domain of dependence contains the mathematical domain of dependence.

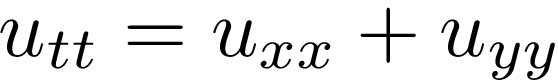
**5) validation of numerical algorithms**

In order to validate our codes, results from our software will be compared and validated by those calculated from some built-in algorithms and solver from PDE libraries.

**Two-Dimensional heat equation**

Once our results in one-dimensional equation are validated, we can then move to a more complex problem, two-dimensional version of heat equation.

2-D heat equation:

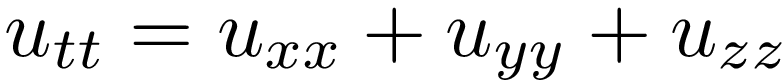


We can apply the same numerical schemes to this two-dimensional problem to solve for numerical solution.

The computing time in 2-D heat equation could be large. By applying CCV, we may probably observe a good speed up by multi-threading, which is another feature of our software, that is to show the computing speed-up by ccv multi-threading.

**Three-Dimensional heat equation**

The 3-D heat equation:



Considering the time constraint, we may or may not be able to the software to three-dimensional heat equation. The algorithm and the numerical scheme should be identical, but the computing power would be badly needed for 3-D problem.

**Qt implementaion**

Qt, a cross-platform framework, which is usually used as a graphical toolkit, will be used as the graphical interface between the users and the software.

A graphical interface will be generated for the user to decide what the dimension is for the problem: 1-D or 2-D, what numerical scheme to apply and what kind of boundary and initial conditions will be used to solve the heat equation. Another input from the user will be the size of grid for the discretization of the spatial and time domains, which are denoted as k and h, respectively.

Alternatively, as for graphical interface, matlab is also a good tool for the generating user-friendly interface. Below is a figure generated by matlab GUI, showing the dialog for the users to input spatial and time step sizes and choose the dimension of the problem. If not specified, the program will take the default input parameters for the software.

