Exercises

Day 1

Exercise #1a

- Write the FORTRAN program:

```
PROGRAM test
IMPLICIT NONE
INTEGER :: i,j,k,n
PRINT*,'Enter j n'
READ(*,*) j,n
DO i=1,n
    IF (i.EQ.j) EXIT
    i = i + 10 ! does this compile ?
ENDDO
k = i
PRINT*,'k = ',k
END PROGRAM test
```

- Compile and run the program.
- What value of k do you obtain for `(j,n)` = `(5,0)`, `(5,10)`, and `(12,10)` ?

Hints #1a

- If you are unfamiliar with linux/UNIX then google and read a primer on linux/UNIX, fx.:
  - `www.gbar.dtu.dk/index.php/faq/48-unix-commands`
  - `www.ks.uiuc.edu/Training/Tutorials/Reference/unixprimer.html`
- If you are unfamiliar with linux/UNIX editors then google one of:
  - `vi (avail. on all linux/UNIX systems; 1976).`
  - `emacs` (alternative to 'vi'; 1975).
  - `nedit` (the Nirvana editor).
  - `nano` (1999).
  - gedit (default text editor for GNOME; 1999).

Hints #1a

- To compile a routine/program use (in a terminal):
  - `f90 –free example.f` (will compile and link the program; the executable will be named: a.out).
  - `f90 –free example.f –o foo` (as above but the executable will be named: `foo`).
  - `f90 –free –c example.f` (will only compile the program; not link/create an executable).
  - `f90 example.o sub1.o sub2.o` (will link the program and produce the executable: a.out; we assume `example.f`, `sub1.f`, and `sub2.f` have been compiled)
- Use '`man f90`' to read the online manuals.

Hints #1a

- Executables in linux/UNIX:
  - A program in linux/UNIX does not have a special extension (no '.exe'); normally programs have no extension – so fx. '`ls`' is the program for listing files/directories.
  - `foo` (foo must be in the $PATH environment variable)
  - `./foo` (if the current directory: '.' is not in the $PATH)
- The $PATH is set by executing the command in the shell
  - `export PATH=$PATH:<new path>`
- The $PATH can be set permanently in: the login files: `~/.profile` (ksh shell) or `~/.bashrc` (bash shell) or `~/.tcshrc` (tcsh shell)

Exercise #1b

- Write a FORTRAN program (single main program) to solve the unsteady, two-dimensional diffusion problem:

$$\frac{\partial T}{\partial t} = D \nabla^2 T = D\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right)$$

  Here *T* is the unknown temperature, and *D* the diffusion constant. *t* denotes time and *(x,y)* the two spatial coordinates.
- Solve the problem in a unit square *(Lx,Ly) = (1,1)*, subject to the Dirichlet Boundary Condition (BC):

$$T(x,0,t) = T(x,Ly,t) = T(0,y,t) = T(Lx,y,t) = 1.$$

- The initial condition is $T(x,y,t=0) = 0.$

Exercise #1b

- Discretize the problem using 2nd finite-differences for the Laplacian:

$$\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}\approx\frac{T_{i+1,j}-2\mathrm{T}_{i,j}+T_{i-1,j}}{\Delta x^2}+\frac{T_{i,j+1}-2\mathrm{T}_{i,j}+T_{i,j-1}}{\Delta y^2}$$

on a uniform Cartesian grid:

$$\Delta x=\frac{L_x}{N_x-1}\quad\text{and}\quad\Delta y=\frac{L_y}{N_y-1}$$

where *Nx* and *Ny* denote the number of grid points in the *x* and *y*-directions.

Exercise #1b

- Integrate the solution in time using explicit Euler time integration:

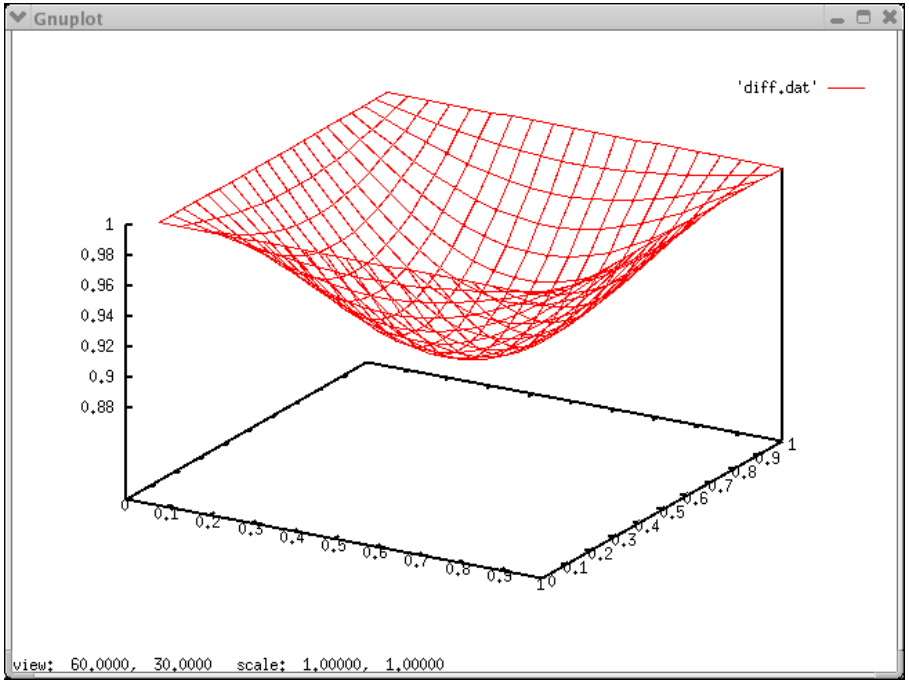
$$\frac{T_{i,j}^{n+1} - T_{i,j}^{n}}{\Delta t} = D\left(\frac{T_{i+1,j}^{n} - 2T_{i,j}^{n} + T_{i-1,j}^{n}}{\Delta x^2} + \frac{T_{i,j+1}^{n} - 2T_{i,j}^{n} + T_{i,j-1}^{n}}{\Delta y^2}\right)$$

Assume D = 1, and select a time step that satisfy the *Fourier* limit:

$$\Delta t < \frac{MIN\left(\Delta x, \Delta y\right)^2}{4D}$$

Exercise #1b

- Perform a simulation using (21,21) grid points, and plot in your reply the field after 200 time steps, that is at t=0.125.
- Attach also the source code.



Hints #1b

- Coding
  - For the Euler time integration you need two fields: one for the old time step and one for the new.
  - To impose the Dirichlet BCs, simply avoid updating the boundary, thus loop as:

```
DO j=2,Ny-1
    DO i=2,Nx-1
        … compute Laplacian etc.
    ENDDO
ENDDO
```

Hints #1b

- Plotting

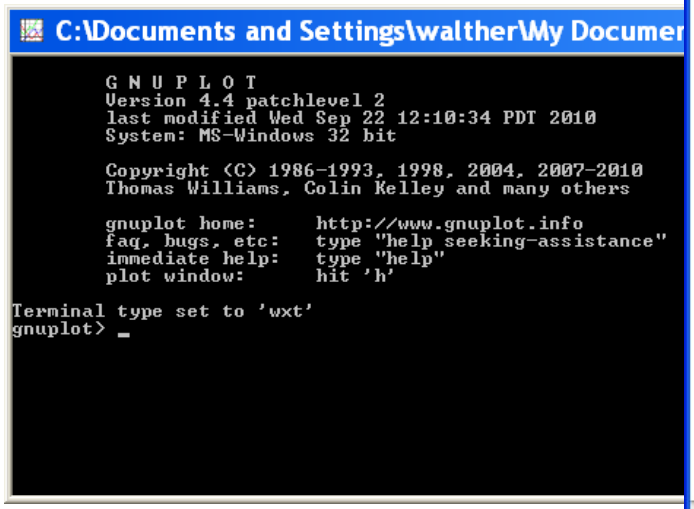
```
OPEN(10,FILE='diff.dat')
DO j=1,Ny
        DO i=1,Nx
                WRITE(10,'(3E12.4)') REAL(i-1)*dx,REAL(j-1)*dy,temp(i,j)
        ENDDO
        WRITE(10,'(A)') ! Will produce a new empty line – and tell gnuplot to lift the pen
ENDDO
CLOSE(10)
```

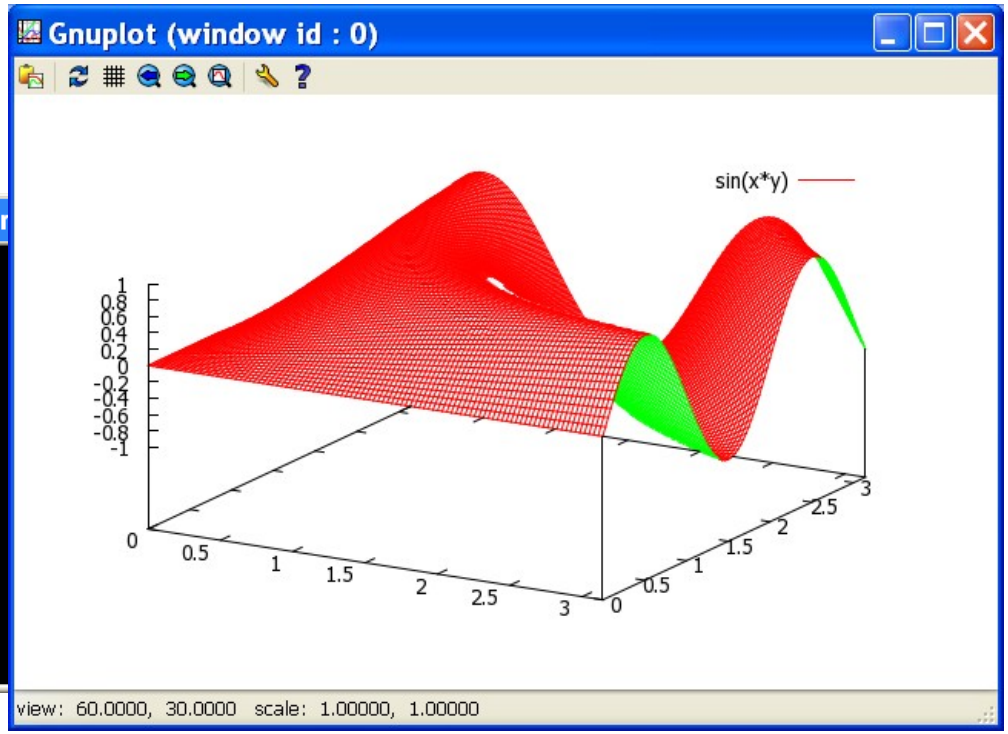
Plot this file (diff.dat) using e.g., gnuplot (in a terminal):

```
gnuplot
sp'diff.dat' u 1:2:3 w l
set term post eps color solid
set out 'diff.ps'
repl
quit
```

Hints #1b

- gnuplot





Online documentation for gnuplot:
www.gnuplot.info/documentation.html
Notice: octave (open source matlab) uses gnuplot for plotting
Another free graphics tool is: xmgrace: plasma-gate.weizmann.ac.il/Grace