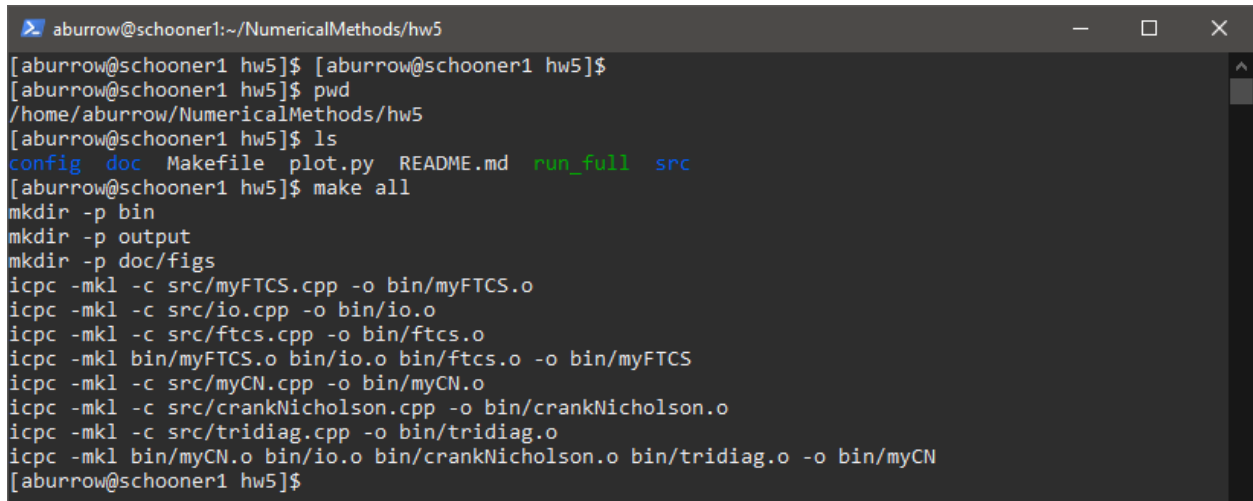


Assignment 5: Results/Description

Problem 1

For this problem I have written two programs (`myFTCS` and `myCN`) to solve the following problem. These may be compiled all together with a `make all` as shown below.



```
aburrow@schooner1:~/NumericalMethods/hw5
[aburrow@schooner1 hw5]$ [aburrow@schooner1 hw5]$
[aburrow@schooner1 hw5]$ pwd
/home/aburrow/NumericalMethods/hw5
[aburrow@schooner1 hw5]$ ls
config doc Makefile plot.py README.md run_full src
[aburrow@schooner1 hw5]$ make all
mkdir -p bin
mkdir -p output
mkdir -p doc/figs
icpc -mkl -c src/myFTCS.cpp -o bin/myFTCS.o
icpc -mkl -c src/io.cpp -o bin/io.o
icpc -mkl -c src/ftcs.cpp -o bin/ftcs.o
icpc -mkl bin/myFTCS.o bin/io.o bin/ftcs.o -o bin/myFTCS
icpc -mkl -c src/myCN.cpp -o bin/myCN.o
icpc -mkl -c src/crankNicholson.cpp -o bin/crankNicholson.o
icpc -mkl -c src/tridiag.cpp -o bin/tridiag.o
icpc -mkl bin/myCN.o bin/io.o bin/crankNicholson.o bin/tridiag.o -o bin/myCN
[aburrow@schooner1 hw5]$
```

This generates executables `myFTCS` and `myCN` in the “./bin” directory, which may be run one at a time. There is also a `plot.py` in the root which may be run to generate plots from the output of the executables. However, to run all these programs at once, I have also included a `run_full` bash script that may be executed for convenience.

In this problem, I use (a) the FTCS method and (b) the Crank-Nicholson method to solve the diffusion equation

$$\frac{\partial n}{\partial t} = \frac{\partial^2 n}{\partial x^2}.$$

(a)

Below I run the program that demonstrates solving our differential equation using the FTCS method:

```
aburrow@schooner1:~/NumericalMethods/hw5
[aburrow@schooner1 hw5]$ [aburrow@schooner1 hw5]$
[aburrow@schooner1 hw5]$ pwd
/home/aburrow/NumericalMethods/hw5
[aburrow@schooner1 hw5]$ ls
bin  config  doc  Makefile  output  plot.py  README.md  run_full  src
[aburrow@schooner1 hw5]$ ./bin/myFTCS
Reading from parameter file: ./config/params
D: 1
dt: 0.02
dx: 0.2
x min: -20
x max: 20
Left bound: 0
Right bound: 0
Evolution time: 20
Using r = 0.5
Calculating solution...
Writing to ./output/ftcs.dat
[aburrow@schooner1 hw5]$
```

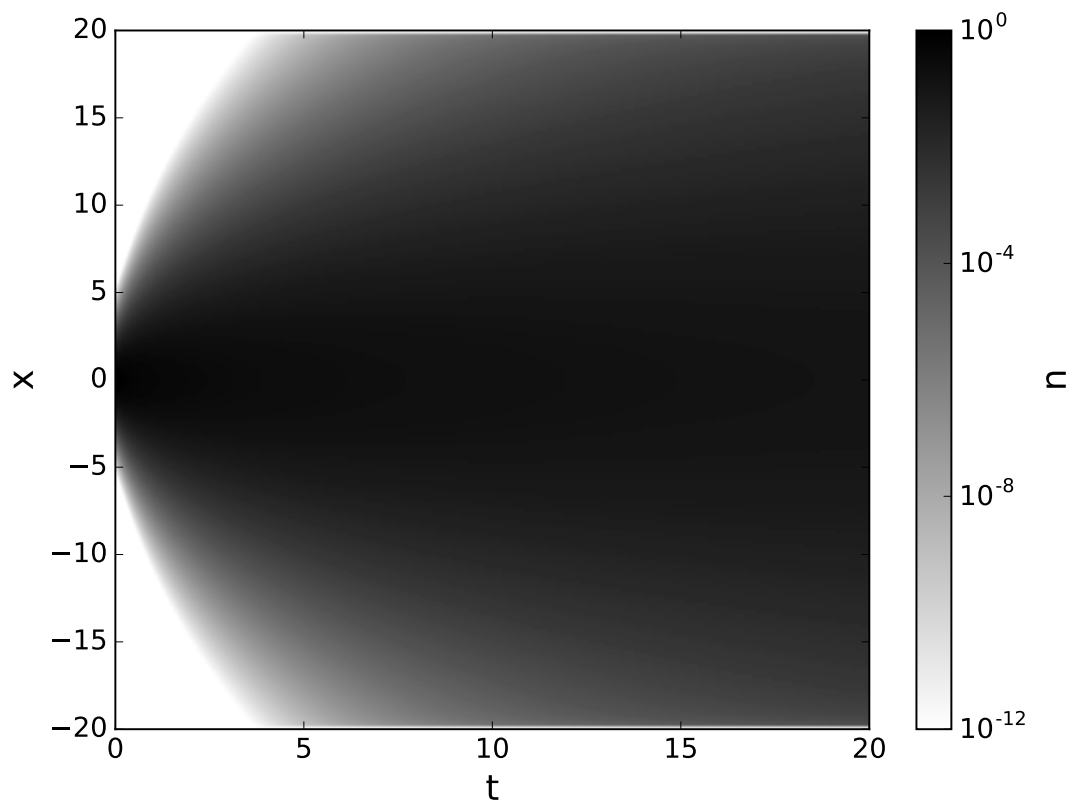


Figure 1: Solution using the FTCS method.

(b)

Below I run the program that demonstrates solving our differential equation using the Crank-Nicholson method:

```
aburrow@schooner1:~/NumericalMethods/hw5
[aburrow@schooner1 hw5]$ [aburrow@schooner1 hw5]$ pwd
/home/aburrow/NumericalMethods/hw5
[aburrow@schooner1 hw5]$ ls
bin config doc Makefile output plot.py README.md run_full src
[aburrow@schooner1 hw5]$ ./bin/myCN
Reading from parameter file: ./config/params
D: 1
dt: 0.02
dx: 0.2
x min: -20
x max: 20
Left bound: 0
Right bound: 0
Evolution time: 20
Using r = 0.5
Calculating solution...
Writing to ./output/crankNicholson.dat
[aburrow@schooner1 hw5]$
```

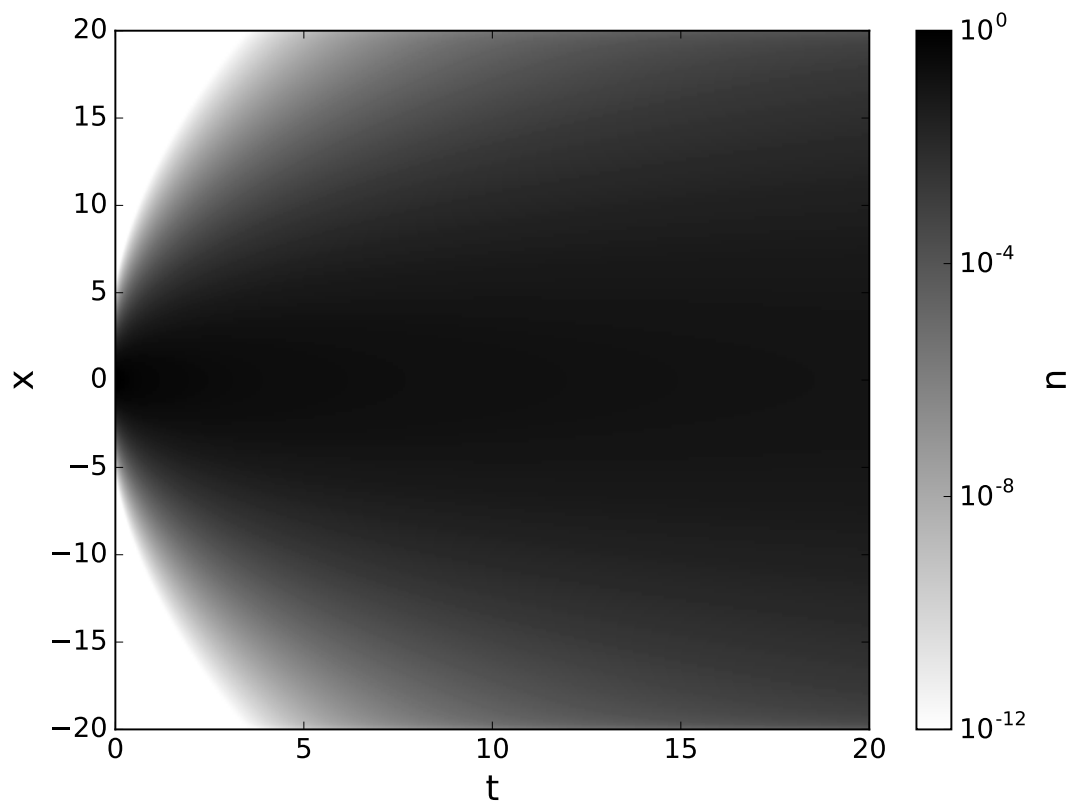


Figure 2: Solution using the Crank-Nicholson method.