Physics 598 Homework 9

Cunwei Fan

Problem 2

This problem is to solve the linear diffusion equation with

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \quad , \quad x \geq 0, t \geq 0$$

with boundary condition:

$$u(0,t)_{,x} = u(0,t) \text{ and } u(\infty, t) = 1$$

and intial condition:

$$u(x,0) = 1$$

Explicit Scheme

The first method is to use explicit scheme to do the integration over time and space. Then write the continous partial differential equation into the discrete form:

$$u_j^{n+1} = u_j^n + \frac{\Delta t}{(\Delta x)^2}(u_{j+1}^n - 2u_j^n + u_{j-1}^n)$$

To implement the boundary condition, I used a forward differntial :

$$\frac{u_1 - u_0}{\Delta x} = u_0 \quad \Rightarrow \quad u_0 = \frac{u_1}{1 + \Delta x}$$

In this integration, I used $\Delta t = 0.2\frac{(\Delta x)^2}{2}$ This implementation, with $\Delta x = 0.1$ makes the following result:

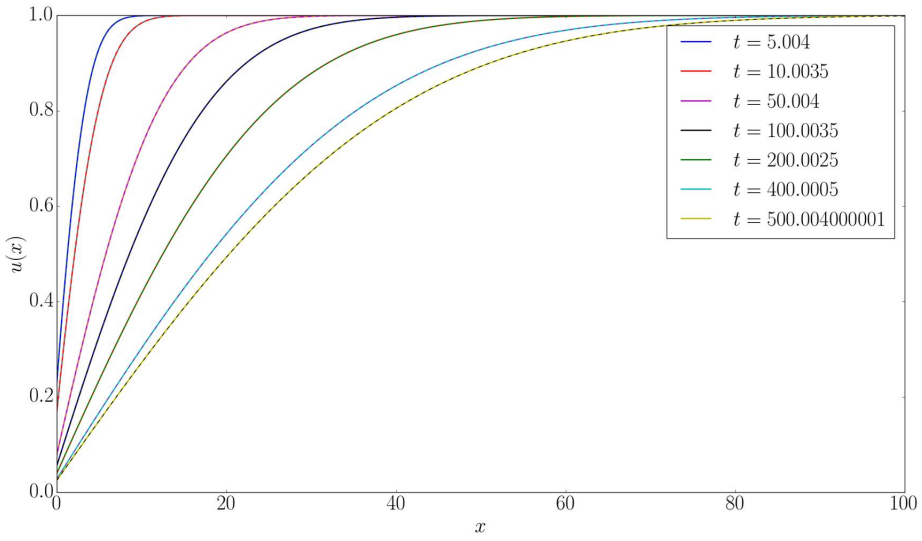


Figure 1: The plot of the numerical solution and the analytic solution

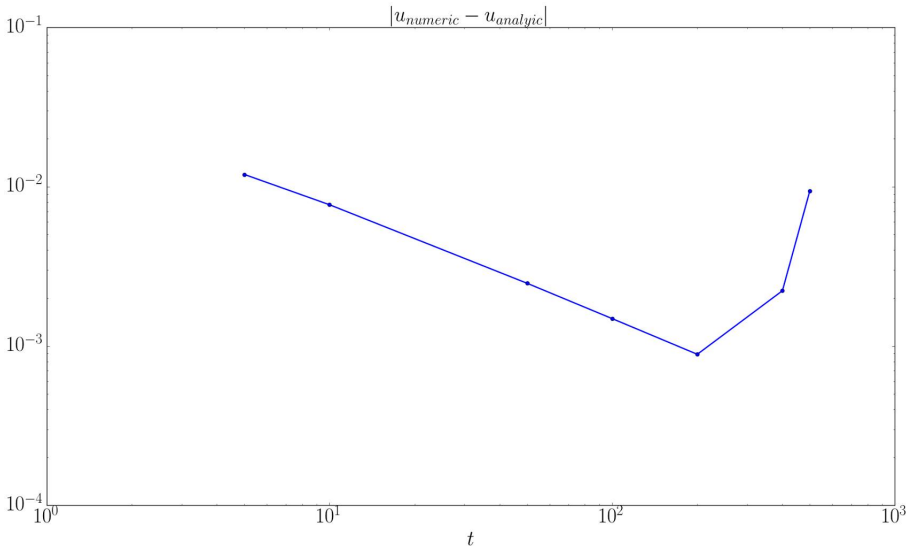


Figure 2: The plot of the residual error with time

The residual error is calculated as

$$\int |u(x) - u_{analytic}(x)|dx$$

To check that there is no bug in the code, there are two ways. The first is to check that the convergent rate is right. To do this, I plotted the error, $2^k(u_k(x) - u_{analytic})$ with space mesh, for different Δx. The Δx stars from 0.2 and goes on by dividing a factor of 2.

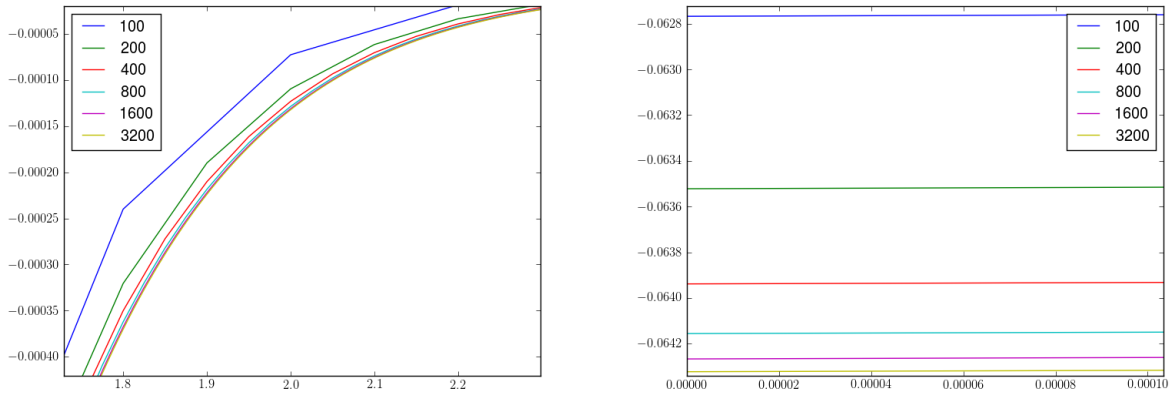


Figure 3: Left: the error plot for different number of points with x axis; Right: the zoom near at $x = 0.0$ error plot

As can be seen in the plot, the difference between each regriding is aprroximately decreasing with factor of 2, which confirms a first order convergence according to the following analysis.

If the convergent rate is first order, then we have

$$\epsilon_0 = u_0(x) - u(x) = E_1 h + E_2 h^2$$

then we decrease the h by a factor of 2, we have

$$\epsilon_1 = u_1(x) - u(x) = E_1\frac{h}{2} + \frac{1}{4}E_2h^2 \quad \Rightarrow \quad \epsilon_1 = 2^1(u_1(x) - u(x)) = E_1h + \frac{1}{2}E_2h^2$$

Similarly, we have

$$\epsilon_2 = 2^2(u_2(x) - u(x)) = E_1h + \frac{1}{4}E_2h^2$$

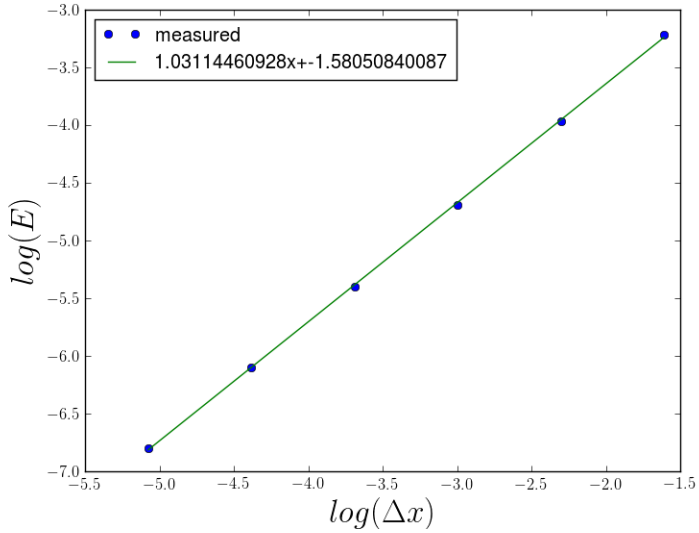
$$\epsilon_3 = 2^3(u_2(x) - u(x)) = E_1h + \frac{1}{8}E_2h^2$$

$$\epsilon_4 = 2^4(u_2(x) - u(x)) = E_1h + \frac{1}{16}E_2h^2$$

$$\dots$$

Therefore, the difference between each line is $\frac{1}{2}E_2h^2, \frac{1}{4}E_2h^2, \frac{1}{8}E_2h^2$. Thus, the difference between each line is decreasing with a factor of 2 for the first order convergence. This effect is shown in the previous plot.

Also, another way To seed the order of convergence is to use the log log plot to desrcibe the residual errors.



From the slop of the fitting line, it is shown clearly that the slope is 1 and it is first order convergence in space.

Another way to see check the code, the self consistence check is also useful and in this case, it is the conservation of energy check. The "energy" mentioned here is

$$F = \int_0^\infty udx$$

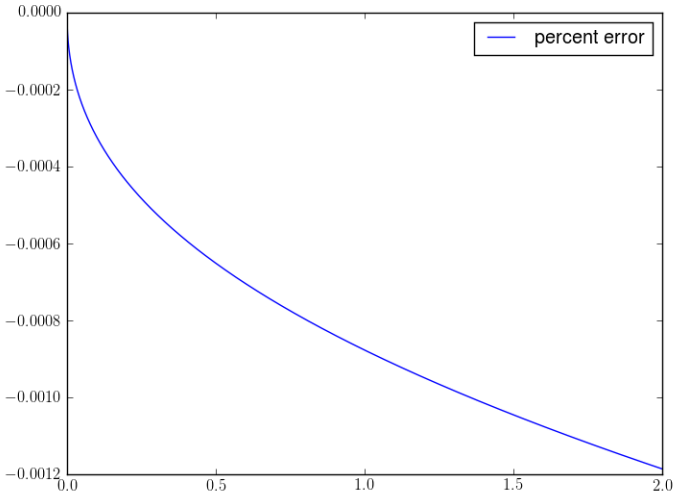


Figure 4: The percent error of calculated F and the analytical $F_{analytic}$

Implicit Scheme

Another way to solve the problem is to use the implicit scheme to integrate.

$$u_j^{n+1} = u_j^n + \frac{\Delta t}{(\Delta x)^2}(u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1})$$

with the definition $\alpha = \frac{\Delta t}{(\Delta x)^2}$ can write the equation in the tridiagonal matrix form:

$$-\alpha u_{j+1}^{n+1} + (2\alpha + 1)u_j^{n+1} - \alpha u_{j-1}^{n+1} = u_j^n \quad \text{for} \quad 1 \leq j \leq N-1$$

For the two end points, use

$$u_0 = \frac{u_1}{1 + \Delta x}, u_N \equiv 1$$

Thus, plug this into the tridiagonal matrix, the u_j^n has to be modifed for $j = 0, N$ with this condition. This implementation, with $\Delta x = 0.1$ makes the following result:

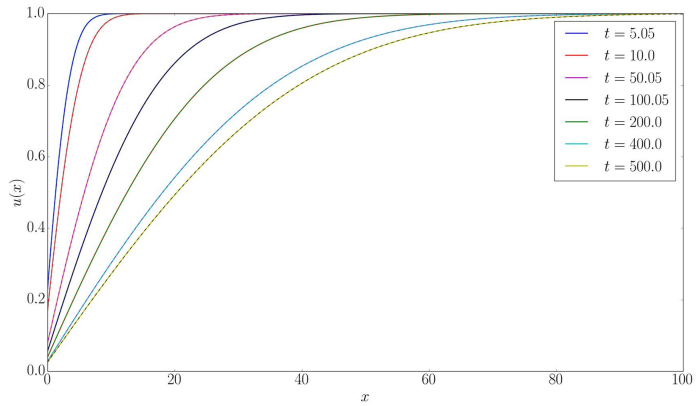


Figure 5: The plot of the numerical solution and the analytic solution

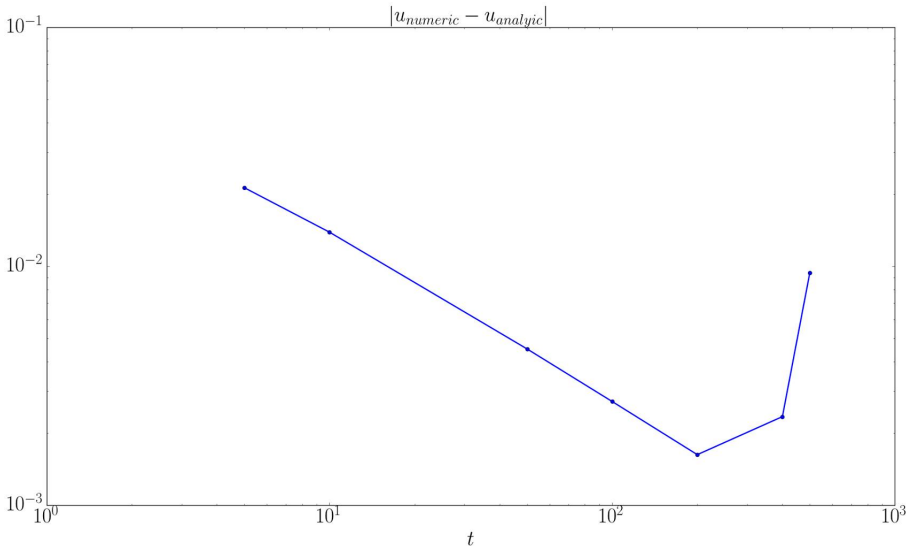


Figure 6: The plot of the residual error with time

From the error plot, it is clear that the scheme is accurate enough for the times covered.

As before, the convergence test is necessary for this case. The Δx stars from 0.2 and goes on by dividing a factor of 2.

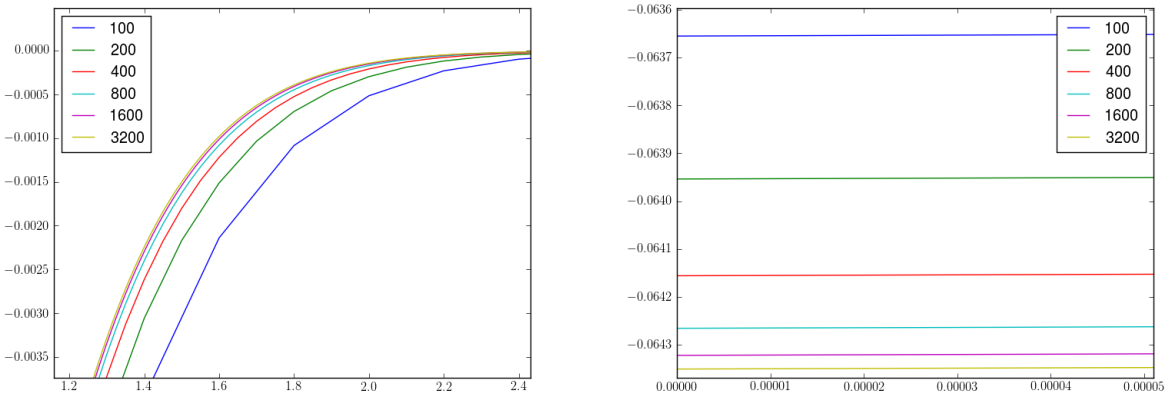


Figure 7: Left: the error plot for different number of points with x axis; Right: the zoom near at $x = 0.0$ error plot

The energy check for this case is similar to the explicit scheme,

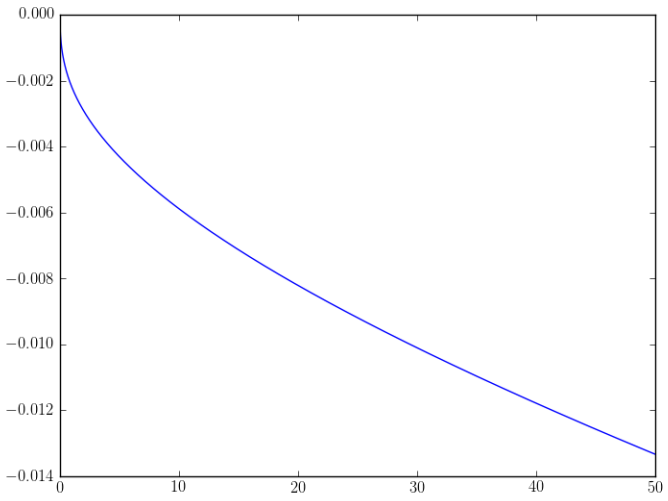


Figure 8: The percent error of calculated F and the analytical $F_{analytic}$

Monte Carlo Scheme

In the monte carlo scheme, I sampled 10000 particles uniformly distributed in the interval $[0, 20]$ at initial time. Then, each particle is moved with the rule:

$$\Delta x_i = q_i \sqrt{2\Delta t}$$

To fulfill the boundary condition, I add a ghost bin on the left side of the first physical bin $[0, \Delta x]$. Then, at each time, I bin all the particles and find the difference between the ghost bin and the first bin and then compare it with the boundary setted difference. Specifically, the difference between the ghost bin B_0 and the first physical bin B_1 shoud be

$$\frac{B_1 - B_0}{\Delta x} = \frac{B_1 + B_0}{2} \quad \Rightarrow \quad \Delta B_0 = \frac{\Delta x (B_1 + B_0)}{2}$$

Thus, exchange particle in B_1 and B_0 so that $B_1 - B_0 = \Delta B_0$

For the right end, I set that whenever paticles cross the right boundary, just reflect them back. Also, at each time I bin the particles, I will divide the $bin(t)$ by $bin(0)$. With this implementation, I can produce a plot as:

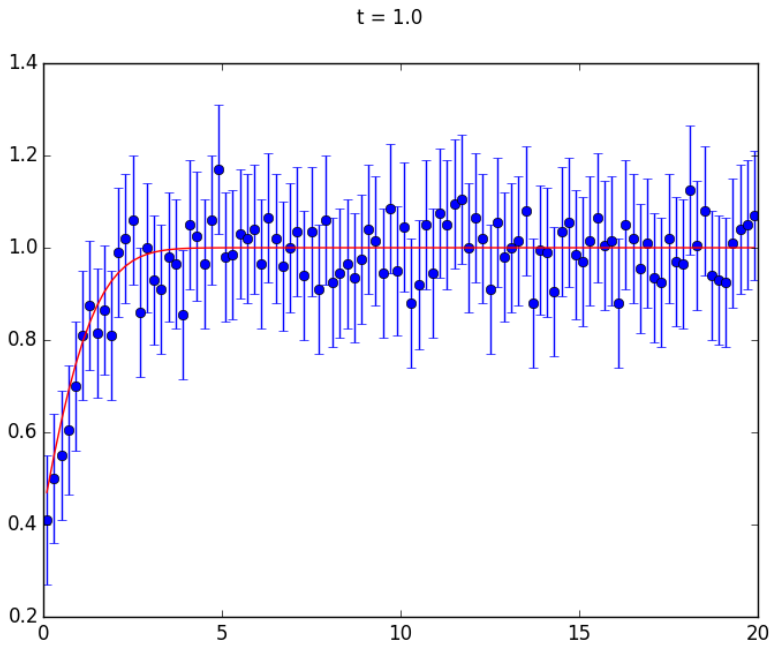


Figure 9: Blue dots are the numerical monte carlo result and the red line is the analytic solution

For the convergent test, the result looks as:

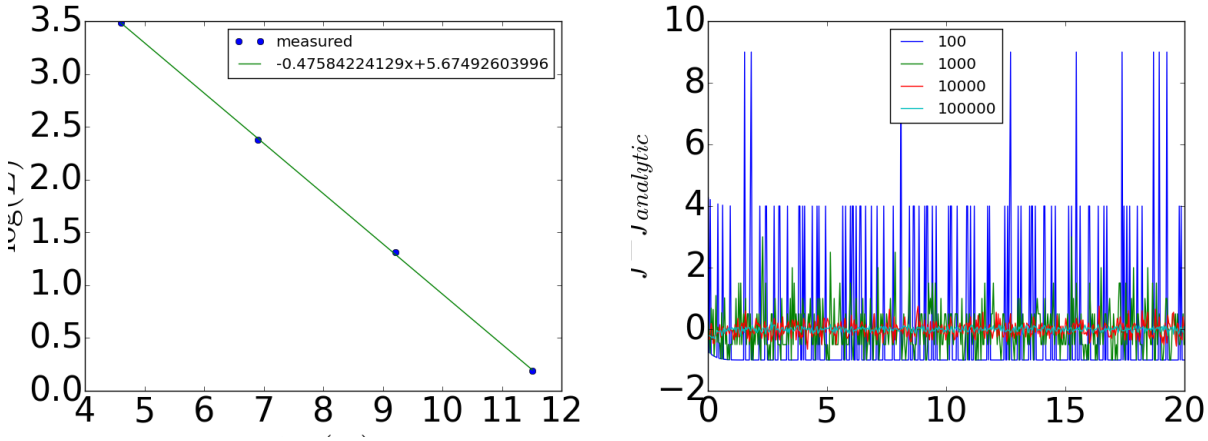


Figure 10: The convergent test is used with 100, 1000, 10000 and 100000 particles

From the plot, it is seen that the slope for the plot is nearlly 0.5 and therefore the monte carlo scheme converges in order of $\frac{1}{\sqrt{N}}$.

Also, the energy check is given the following plot.

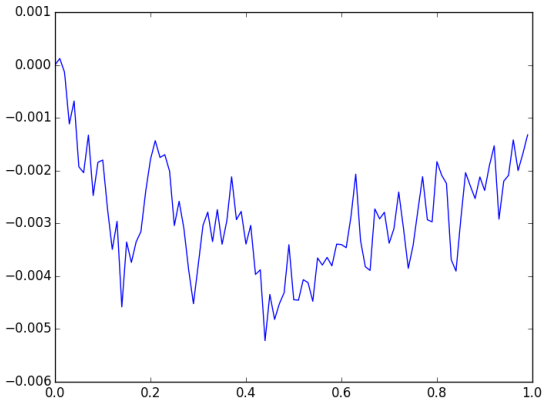


Figure 11: The energy check

Problem 3

In this problem, it is required to solve the non-linear diffusion equation is

$$\frac{\partial u}{\partial t} = \frac{\partial^2}{\partial x^2} u^3$$

with boundary condition:

$$u(-1, t) = -1 \text{ and } u(1, t) = 1$$

and intial condition:

$$u(x, 0) = x$$

Explicit Scheme

Write the diffusioin equation as

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(3u^2 \frac{\partial}{\partial x} u\right)$$

And I used the discrete form of this differential equation as

$$u_j^{n+1} = u_j^n + \frac{\Delta t}{(\Delta x)^2}(u^3)_{j+1}^n - 2(u^3)_j^n + (u^3)_{j-1}^n$$

The stability condition for this question is

$$\Delta t << \frac{(\Delta x)^2}{2(3u(x)^2)}$$

and I used a factor of 0.4 for this conditioin.

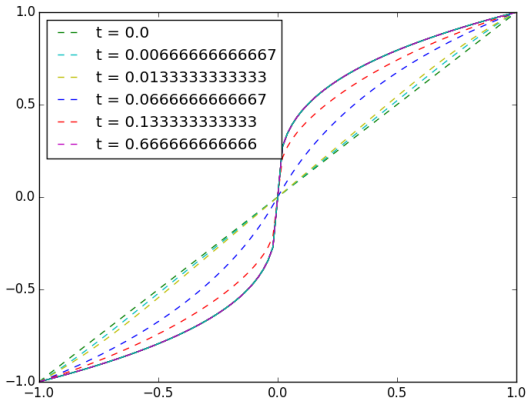


Figure 12: The plot of the numerical solution and the analytic solution at $t \to \infty$

To check the accuracy of this code, I use a definite time at $t = 1.0$ for which the solution comes to the final steady state. Below are the convergence test result.

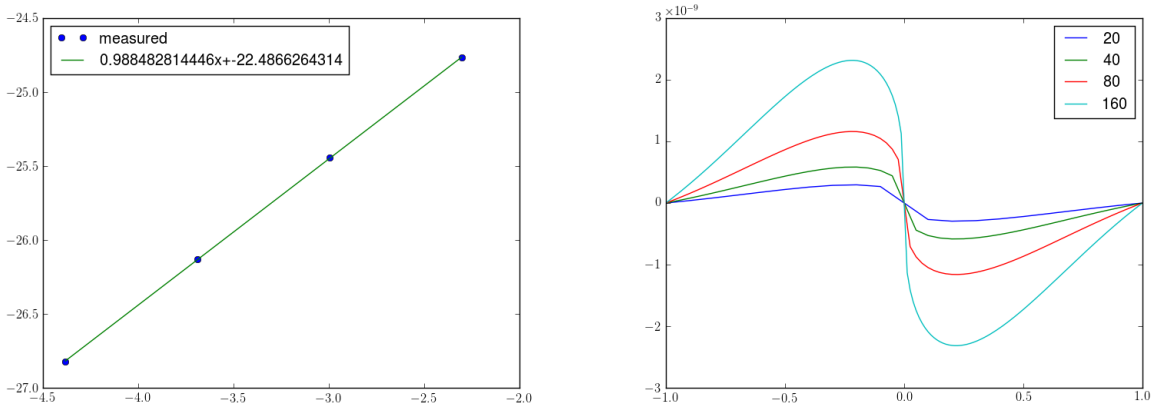


Figure 13: Right: the error plot for different number of points with x axis; Left: the zoom near at $x = 0.0$ error plot

As can be seen in the plots, the slope of log error residual vs $\log(\Delta x)$ is nearlly one. It is obvious the scheme has first order convergence. Secondly, the difference between two consecutive curves decreases with factor 2. It also showes that the scheme has first order convergence.

Implicit Scheme

To use the implicit scheme, we first change the discrete time differential in the explicit scheme equal to the $n + 1$ functions, i.e.

$$u_j^{n+1} = u_j^n + \frac{\Delta t}{(\Delta x)^2}(u^3)_{j+1}^{n+1} - 2(u^3)_j^{n+1} + (u^3)_{j-1}^{n+1}$$

However, to solve this equation, need to linearize it.

To linearize the equations, first notice,

$$(u^3)_j^{n+1} = (u^3)_j^n + 3(u^2)_j^n(u_j^{n+1} - u_j^n)$$

Also, by defining a new variable, $w_j = u_j^n + 1 - u_j^n$, the equation can be written in the form:

$$w_j - \frac{3\Delta t}{(\Delta x)^2}[(u^2)_{j+1}^n W_{j+1} - 2(u^2)_j^n W_j + (u^2)_{j-1}^n W_{j-1}] = \frac{\Delta t}{(\Delta x)^2}[(u^3)_{j+1}^n - (u^3)_j^n + (u^3)_{j-1}^n]$$

Notice that the RHS is known and the LHS has know coefficients for unknown functions $\{W_j\}$. Then use the tridiagonal matrix to solve w_j and add it to the known u_j^n to find u_j^{n+1}.

Also, the boundary contion sets $u_0 = -1$ and $u_N = 1$. Plug this into the above equation at $n = 1$ and $n = N - 1$, it can be shown that the equation becomes,

$$w_2 - \frac{3\Delta t}{(\Delta x)^2}[(u^2)_2^n W_2 - 2(u^2)_1^n W_1 + W_0] = \frac{\Delta t}{(\Delta x)^2}[(u^3)_2^n - (u^3)_1^n - 1]$$

Therefore, in the matrix equation, the first entry of the right hand side is no longer simply plug in $n = 1$ but also need to move the $w_0 = 0$ term to the right hand side and since w_0 is zero in this case(u_0 does not change with time), the first entry of the right hand side is

$$\frac{\Delta t}{(\Delta x)^2}[(u^3)_2^n - (u^3)_1^n - 1]$$

Since the implicit scheme is absolutely stable, the Δt can be chosen aribitraryly. However, to compare, I set Δt the same as in the explicit scheme. Using this implementation, the implicit scheme gives result as

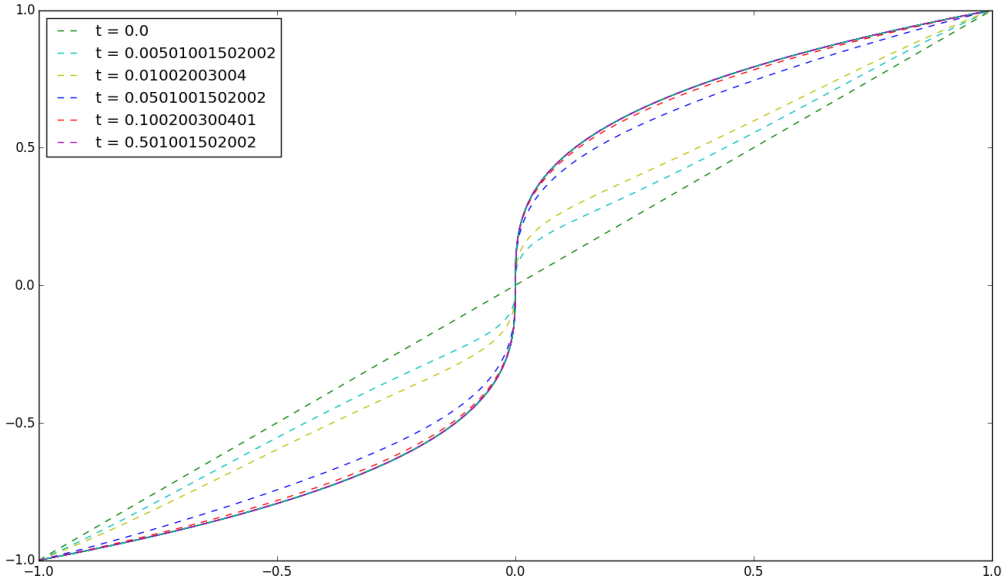


Figure 14: The plot of the numerical solution and the analytic solution at $t \to \infty$

As before, to test for the code, the convergent test is done for this scheme. To check the accuracy of this code, I use a definite time at $t = 1.0$ for which the solution comes to the final steady state. Below are the convergence test result.

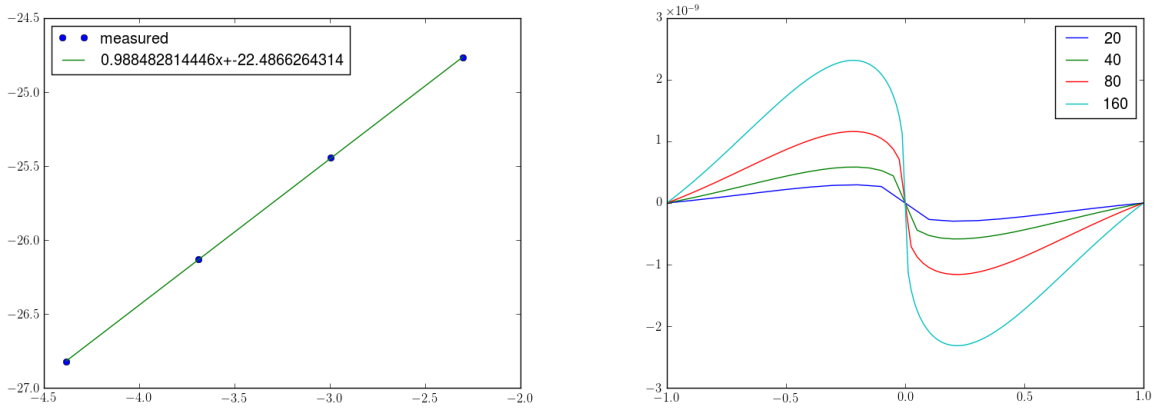


Figure 15: Right: the error plot for different number of points with x axis; Left: the zoom near at $x = 0.0$ error plot

As can be seen in the plots, the slope of log error residual vs $\log(\Delta x)$ is nearlly one. It is obvious the scheme has first order convergence. Secondly, the difference between two consecutive curves decreases with factor 2. It also showes that the scheme has first order convergence.

Monte Carlo Scheme

A big challenge for monte carlo scheme is to represent the negative function values. To represent the negative function values, I introduced negative particles. That is, I have positive particles and negative particles and when I bin them, the negative particles contribute to -1 and positive particles contribute to 1. To fulfill the boundary conditions, I multiply a factor to the monte carlo simulated function so that the end points are 1 and -1.

Since, the equation can be seen as a diffusion equation with diffusion constant $3u^2$. The monte carlo particles are going to move with the following rule:

$$x_i^{n+1} = q_i \sqrt{2(3u^2)\Delta t}$$

Using the same Δt used in explicit scheme, the implementation gives the following result

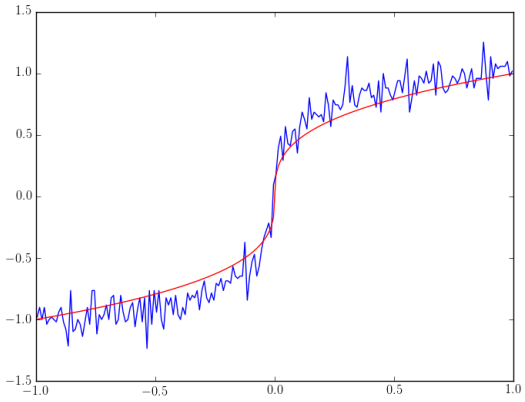
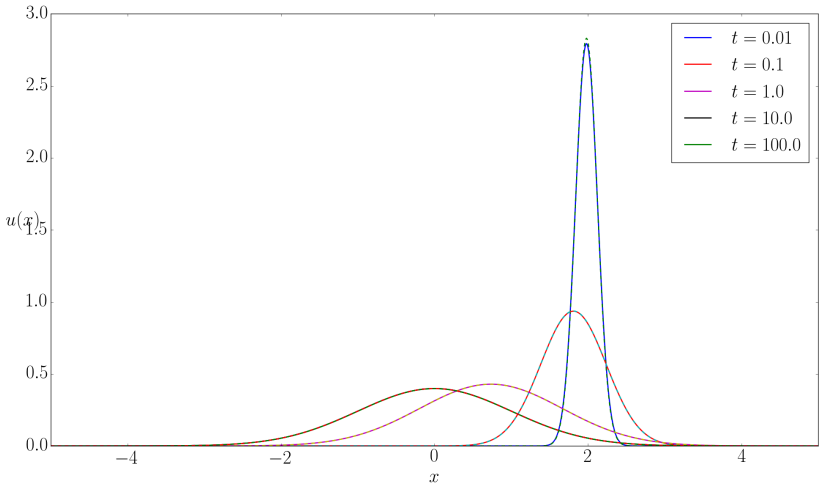


Figure 16: The plot of the numerical solution and the analytic solution at $t \to \infty$



Problem 4

This problem is to solve the Fokker-Plank equation:

$$\frac{\partial f}{\partial t} = \frac{\partial}{\partial v}(vf) + \frac{\partial^2 f}{\partial v^2}, \qquad -\infty \le v \le \infty, \qquad t \ge 0,$$

subject to to boundary conditions

$$f(-\infty, t) = f(\infty, t) = 0,$$

and initial conditions

$$f(v, 0) = \delta(v - 2).$$

Explicit Scheme

In the explicit scheme, the finite difference method is used as

$$\frac{\partial f}{\partial t} = \frac{\partial}{\partial v}(vf) + \frac{\partial^2 f}{\partial v^2},$$

obtaining

$$\frac{f_j^{n+1} - f_j^n}{\Delta t} = \frac{v_{j+1} f_{j+1}^n - v_{j-1} f_{j-1}^n}{2\Delta v} + \frac{f_{j+1}^n - 2f_j^n + f_{j-1}^n}{(\Delta v)^2}.$$

Written more explicitly,

$$f_j^{n+1} = f_j^n + \frac{\Delta t}{(\Delta v)^2}\left[(1 + \frac{1}{2} v_{j+1}\Delta v) f_{j+1}^n - 2f_j^n + (1 - \frac{1}{2} v_{j-1}\Delta v) f_{j-1}^n\right].$$

Here are the results of this implementation:

To test the code, as before, the convergent test is used.

The first image above uses the $2^k(u_k(x) - u(x))$ as before, and since the consecutive spacing is equal, the error must be of higher orders. In fact, if plot with $4^k(u_k(x) - u(x))$, you will find all lines are overlaped, which means that the difference is of higher orders than 2.

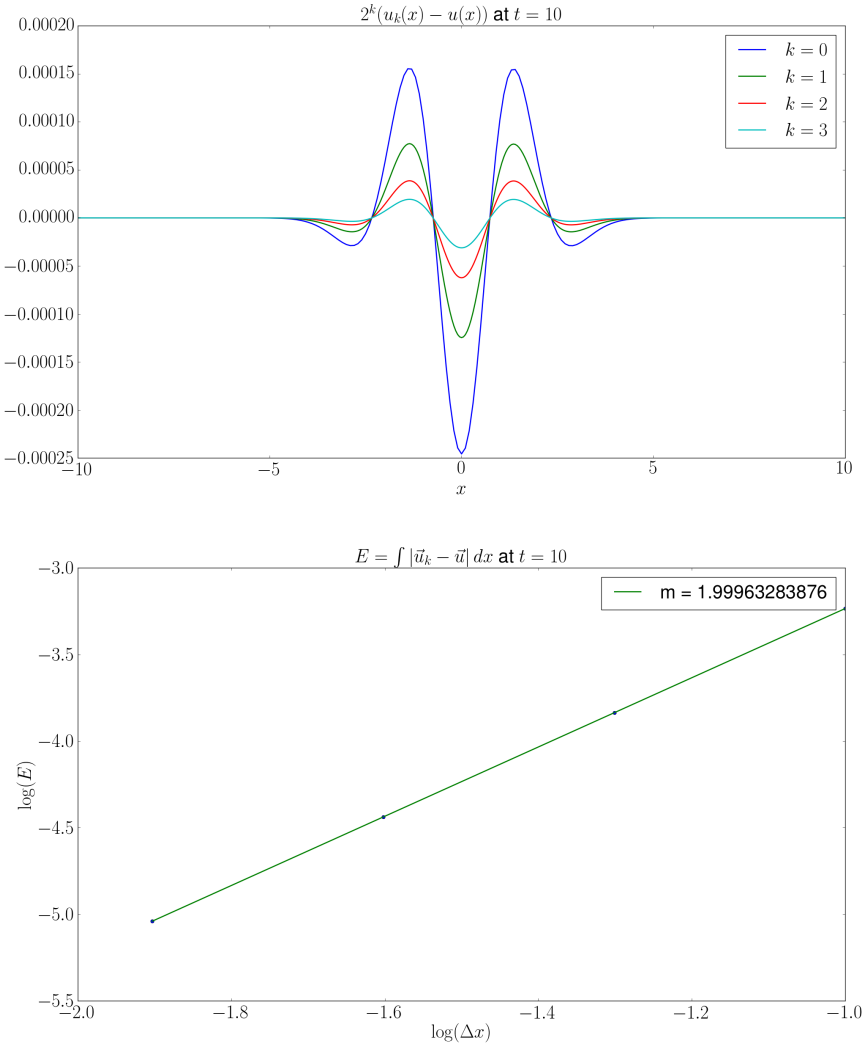
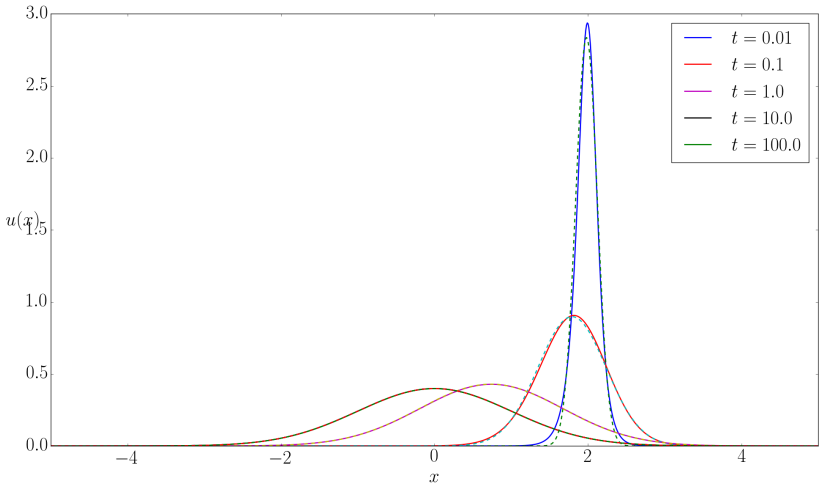


Figure 17: The first image shows that the difference between the consecutive is equal and therefore it must be of higher order convergence, the scond image shows that the error residual has slope 2 respect the Δx which showes that the convergence is second order



Implicit Shceme

From the implicit formulation

$$f_j^{n+1} = f_j^n + \frac{\Delta t}{(\Delta v)^2}\left[(1+\frac{1}{2}v_{j+1}\Delta v)f_{j+1}^{n+1} - 2f_j^{n+1} + (1-\frac{1}{2}v_{j-1}\Delta v)f_{j-1}^{n+1}\right].$$

Rewrite as

$$f_j^n = -\left[\frac{\Delta t}{(\Delta v)^2}(1-\frac{1}{2}v_{j-1}\Delta v)\right]f_{j-1}^{n+1} + \left[1+2\frac{\Delta t}{(\Delta v)^2}\right]f_j^{n+1} - \left[\frac{\Delta t}{(\Delta v)^2}(1+\frac{1}{2}v_{j+1}\Delta v)\right]f_{j+1}^{n+1}$$

Here are the results of this implementation:
To test the code, as before, the convergent test is used.
Thus, this scheme has second order convergence.

Monte Carlo Scheme

The Fokker-Plank equation for Monte Carlo is

$$\frac{\partial f}{\partial t} = \frac{\partial}{\partial v}\left(f\left\langle \Delta v\right\rangle\right) + \frac{1}{2}\frac{\partial^2}{\partial v^2}\left(f\left\langle (\Delta v)^2\right\rangle\right)$$

Comparing to the problem, we have

$$\left\langle \Delta v\right\rangle = v, \qquad \left\langle (\Delta v)^2\right\rangle = 2$$

Each particle then evolves as

$$\begin{aligned} v_j^{n+1} &= v_j^n + \Delta v\Delta t + q_j\sqrt{\left\langle (\Delta v)^2\right\rangle \Delta t} \\ &= v_j^n + v_j^n\Delta t + q_j\sqrt{2\Delta t} \end{aligned}$$

where q_j is sampled with probability $p(q_j) = (2\pi)^{-1/2}e^{-(q_j)^2/2}$
This implementation gives the follwing results:
To test the code, as before, the convergent test is used.

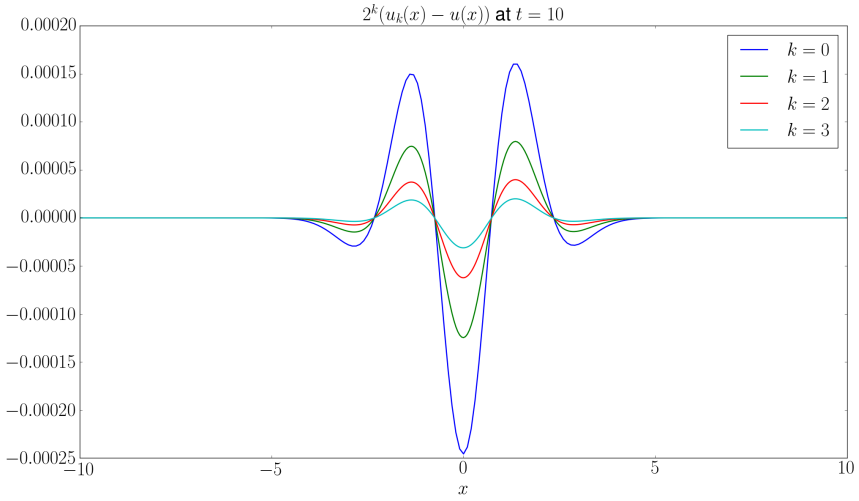


Figure 18: The image shows that the difference between the consecutive is equal and therefore it must be of higher order convergence

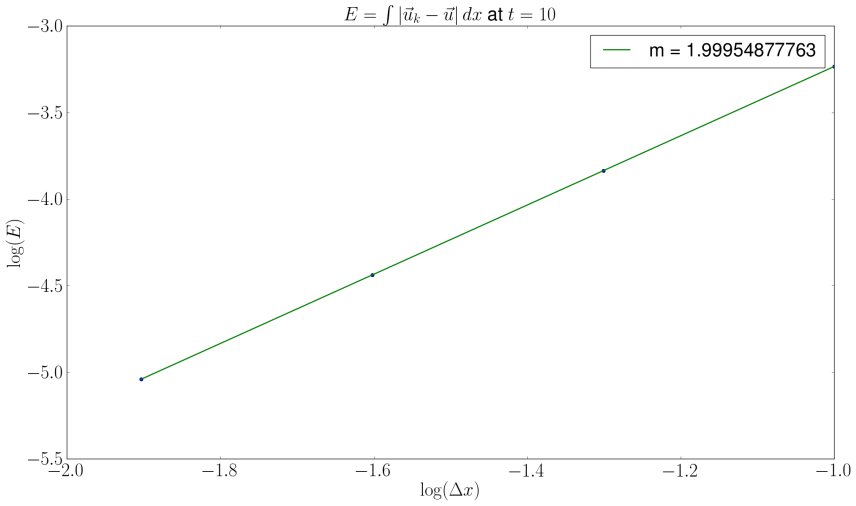
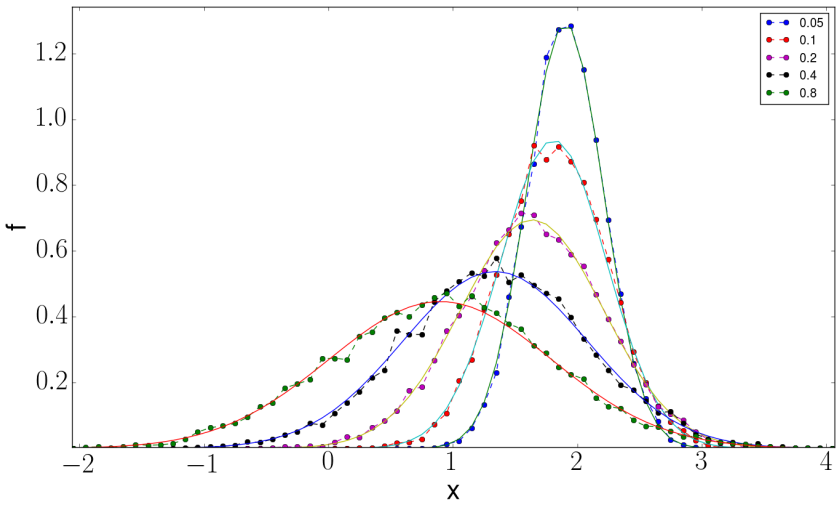
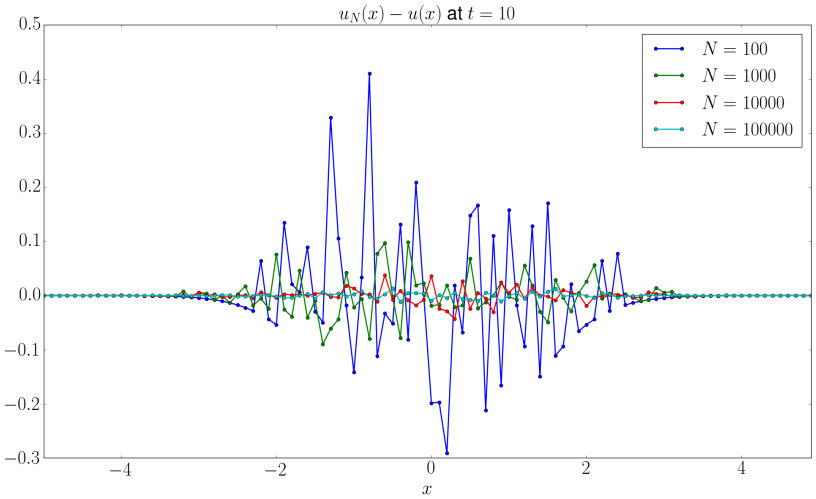


Figure 19: the image shows that the error residual has slope 2 respect the Δx which showes that the convergence is second order


0.05
0.1
0.2
0.4
0.8
f
x


$u_N(x) - u(x)$ at $t = 10$
$N = 100$
$N = 1000$
$N = 10000$
$N = 100000$
x

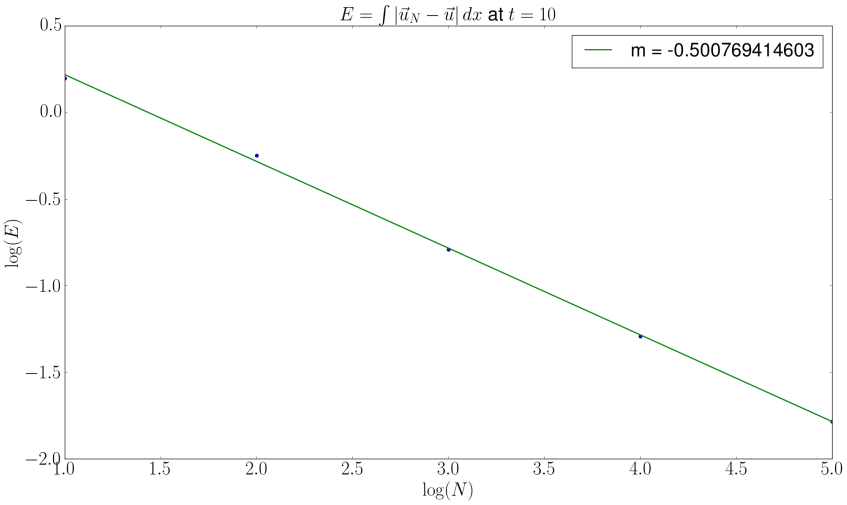


Figure 20: Error$\sim \mathcal{O}(1/\sqrt{N})$ as expected.