

Homework 2

Ningyu Ma
ME614 Spring 2017

March 10, 2017

Problem 1.

(a) For n_{max} , I find it in python using the command,

```
while np.isfinite(pi_star * np.pi)
```

where $\hat{\pi}^*$ and π are in float16, float32 and float64 format respectively. I find that the n_{max} for float16, float32 and float64 are 7, 76, 619 respectively.

From Figure 1 to 3, we can see that the round off error increase with bigger n and it is larger for less precise format and smaller for better precise format.

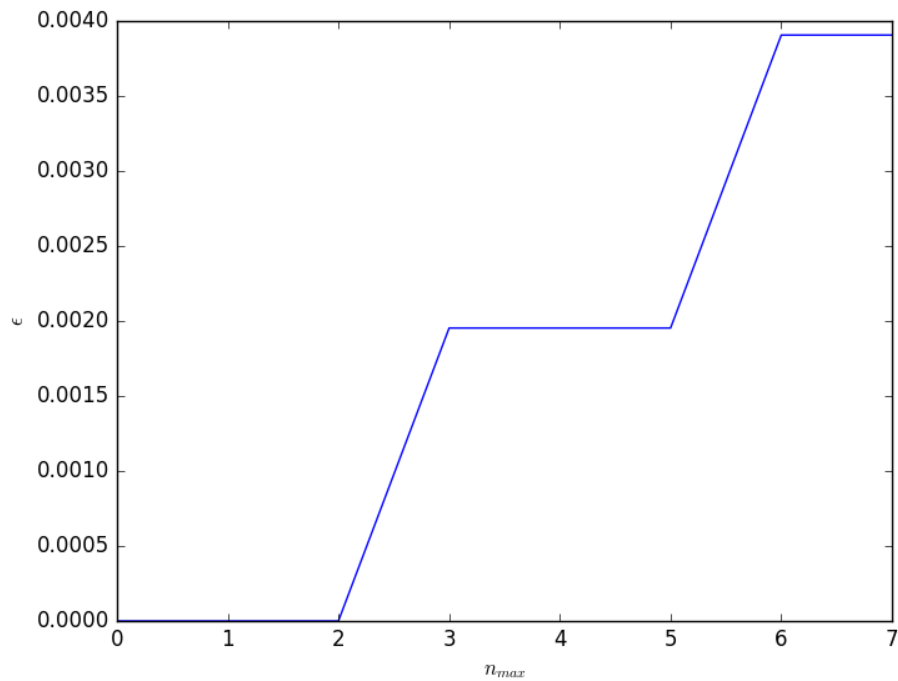


Figure 1: Round off Error ϵ vs. n for float16

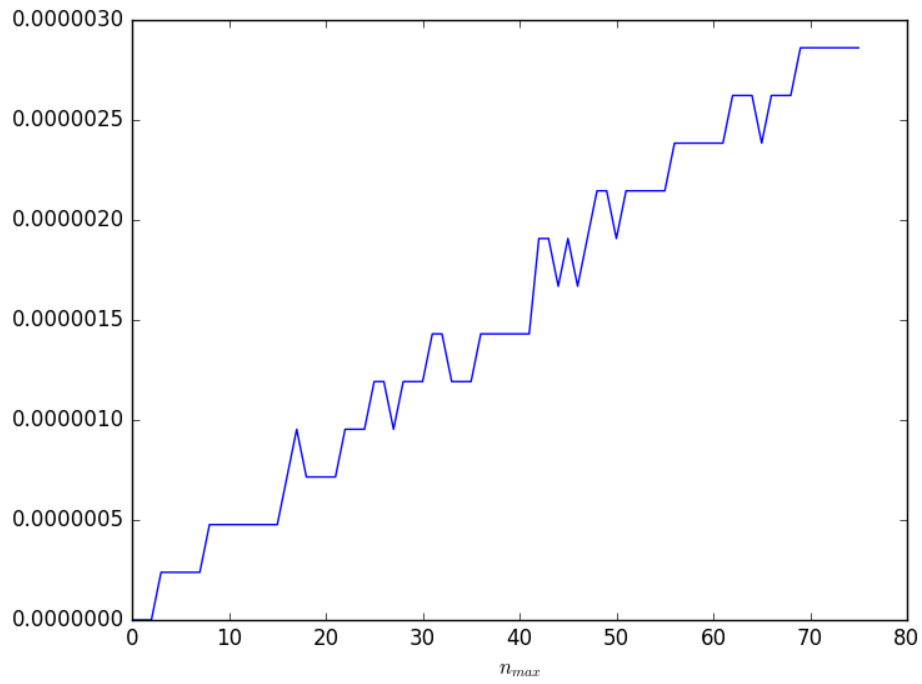


Figure 2: Round off Error ϵ vs. n for float32

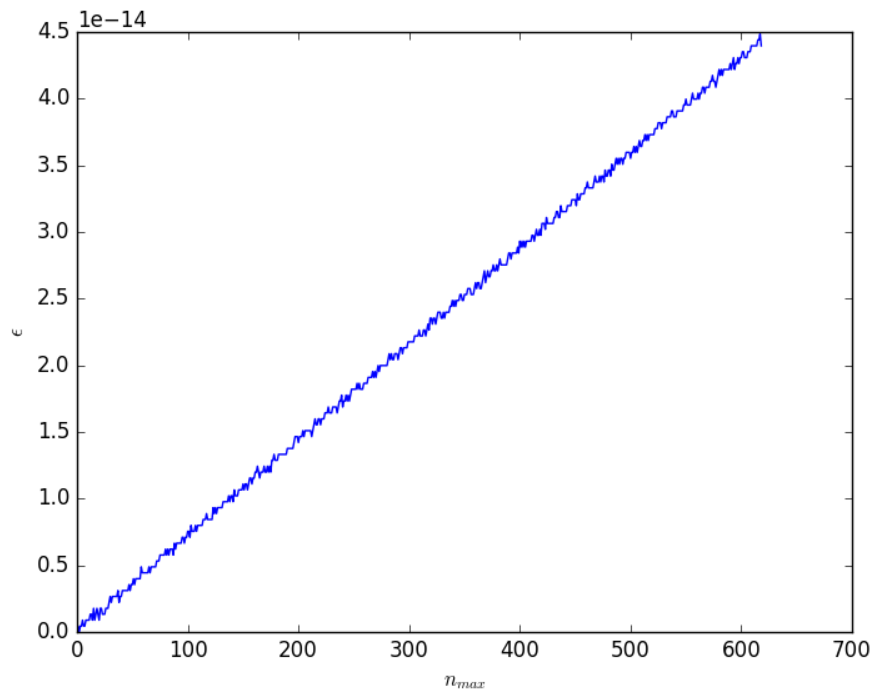


Figure 3: Round off Error ϵ vs. n for float64

Problem 2.

(a) for $u(x, t) = c_1 e^{(-\omega_1^2 \alpha t)} \sin(\omega_1(x - ct) - \gamma_1) - c_2 e^{(-\omega_2^2 \alpha t)} \cos(\omega_2(x - ct) - \gamma_2)$, if we plug it in to the advection/diffusion equation, we can get:

$$\frac{\partial u}{\partial t} = c_1(-\omega_1^2 \alpha) e^{(-\omega_1^2 \alpha t)} \sin(\omega_1(x - ct) - \gamma_1) + c_1(-\omega_1 c) e^{(-\omega_1^2 \alpha t)} \cos(\omega_1(x - ct) - \gamma_1) - c_2(-\omega_2^2 \alpha) e^{(-\omega_2^2 \alpha t)} \cos(\omega_2(x - ct) - \gamma_2) + c_2(-\omega_2 c) e^{(-\omega_2^2 \alpha t)} \sin(\omega_2(x - ct) - \gamma_2)$$

$$\frac{\partial u}{\partial x} = c_1 \omega_1 e^{(-\omega_1^2 \alpha t)} \cos(\omega_1(x - ct) - \gamma_1) + c_2 \omega_2 e^{(-\omega_2^2 \alpha t)} \sin(\omega_2(x - ct) - \gamma_2)$$

$$\frac{\partial^2 u}{\partial x^2} = -c_1 \omega_1^2 e^{(-\omega_1^2 \alpha t)} \sin(\omega_1(x - ct) - \gamma_1) + c_2 \omega_2^2 e^{(-\omega_2^2 \alpha t)} \cos(\omega_2(x - ct) - \gamma_2)$$

And we can see that $\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} - \alpha \frac{\partial^2 u}{\partial x^2} = 0$

(b) Figure 4 to 7, we can see the results from python code. We can see that the results are indistinguishable between each scheme combination when N is large.

So I calculated the root-square-mean for each scheme combination w.r.t. the Analytical solution.

From the result of python code, the best scheme is 2nd-order-central & Crank-Nicolson for all four cases and least performing one is 1st-order-upwind & Explicit-Euler scheme.

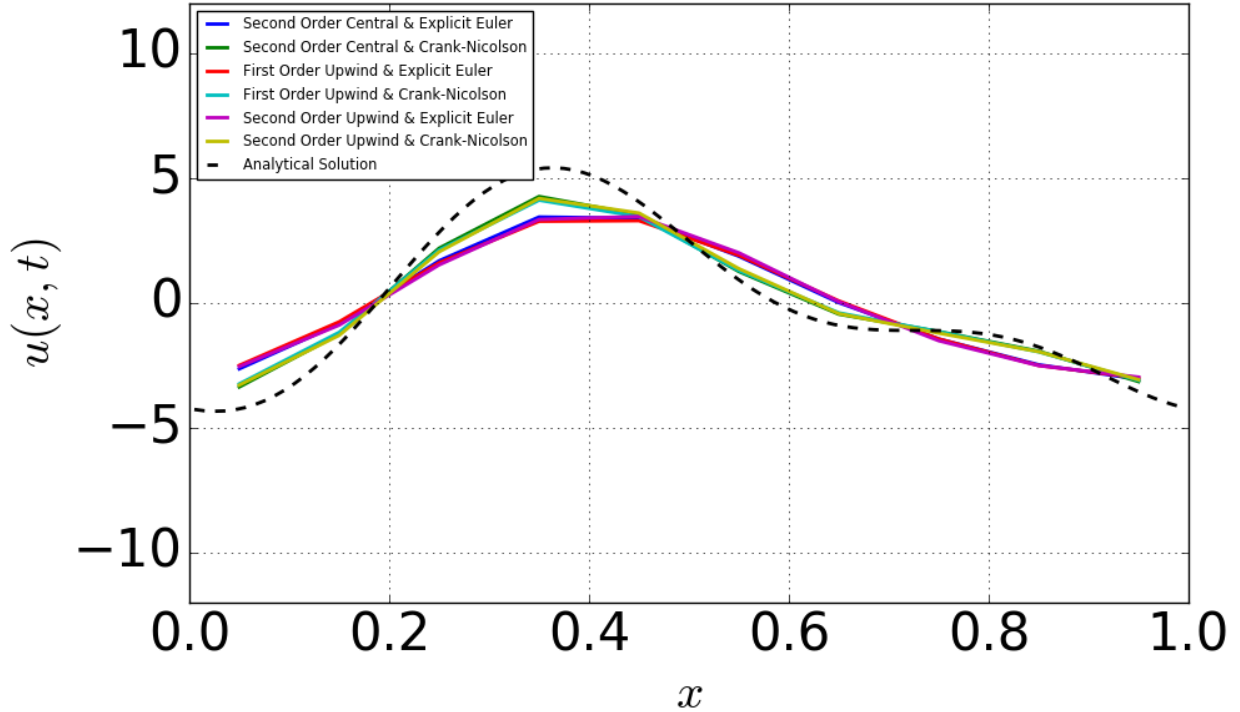


Figure 4: Numerical Solution at $t = t_f$ with $N = 10$

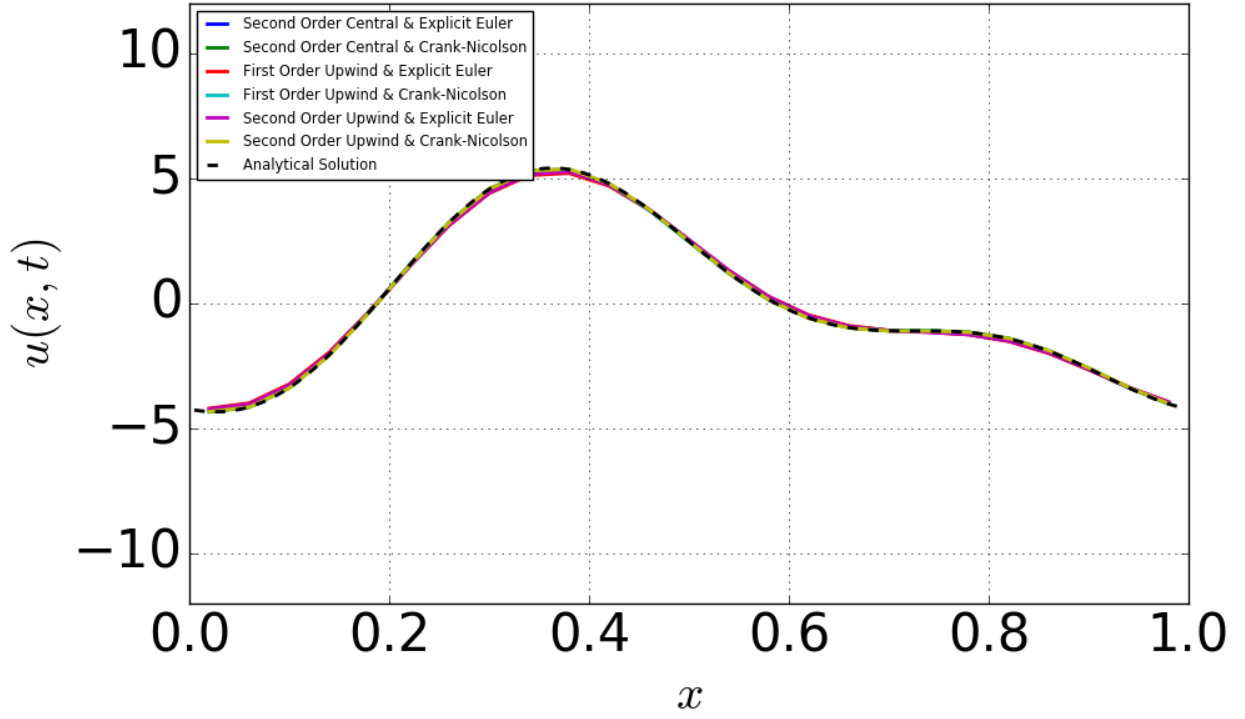


Figure 5: Numerical Solution at $t = t_f$ with $N = 25$

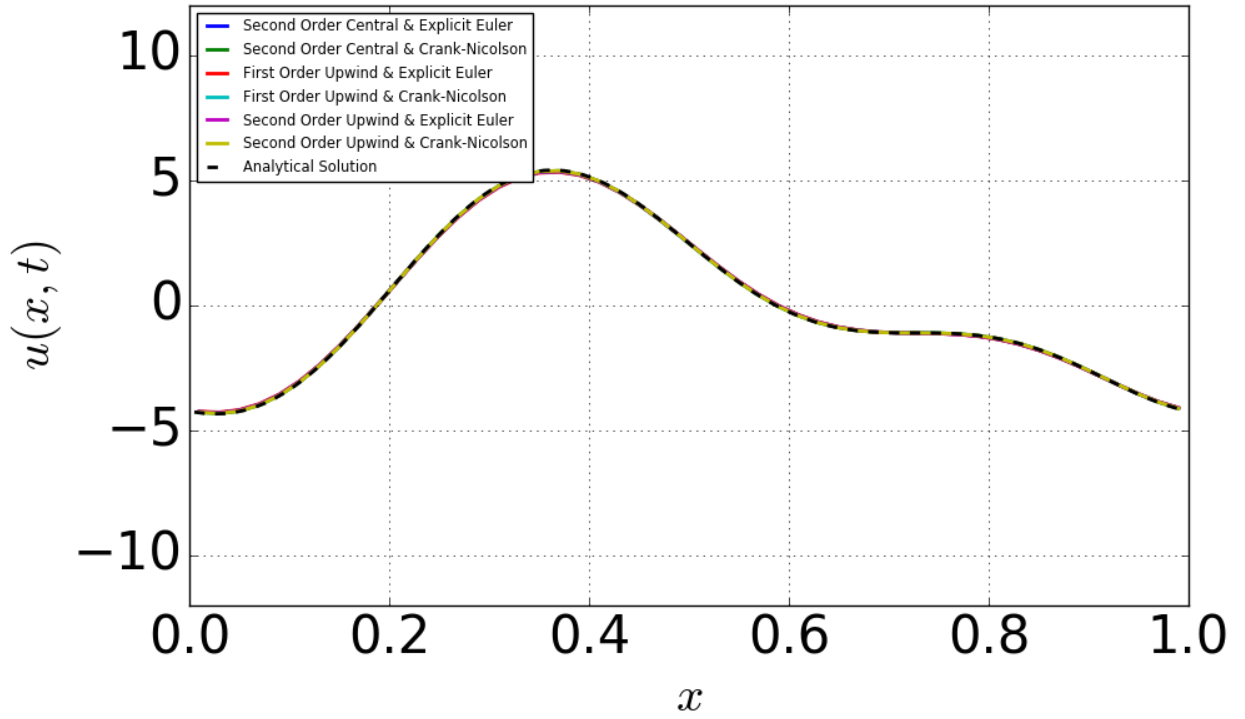


Figure 6: Numerical Solution at $t = t_f$ with $N = 50$

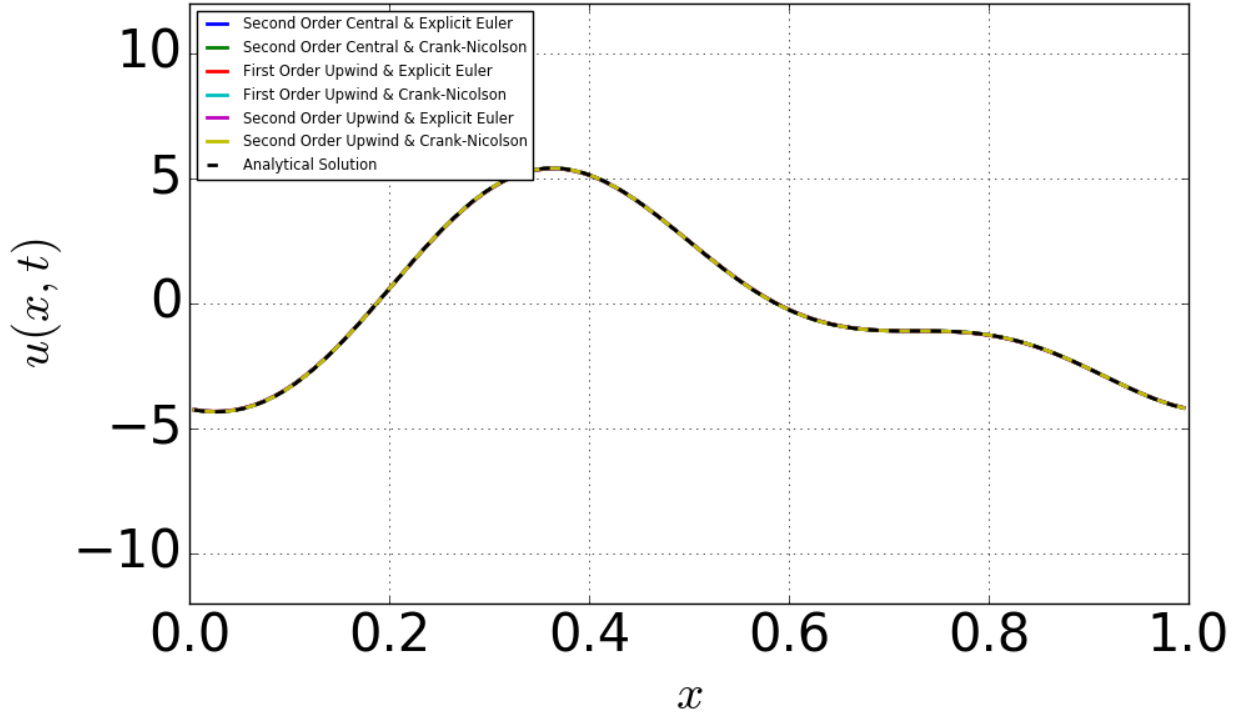


Figure 7: Numerical Solution at $t = t_f$ with $N = 100$

(c)

From Figure 8, which is the case of fixed small Δt with changing Δx , we can see that for the two second order spatial discretization scheme, the order of accuracy is second order, and for the one first order upwind scheme, the order of accuracy is one.

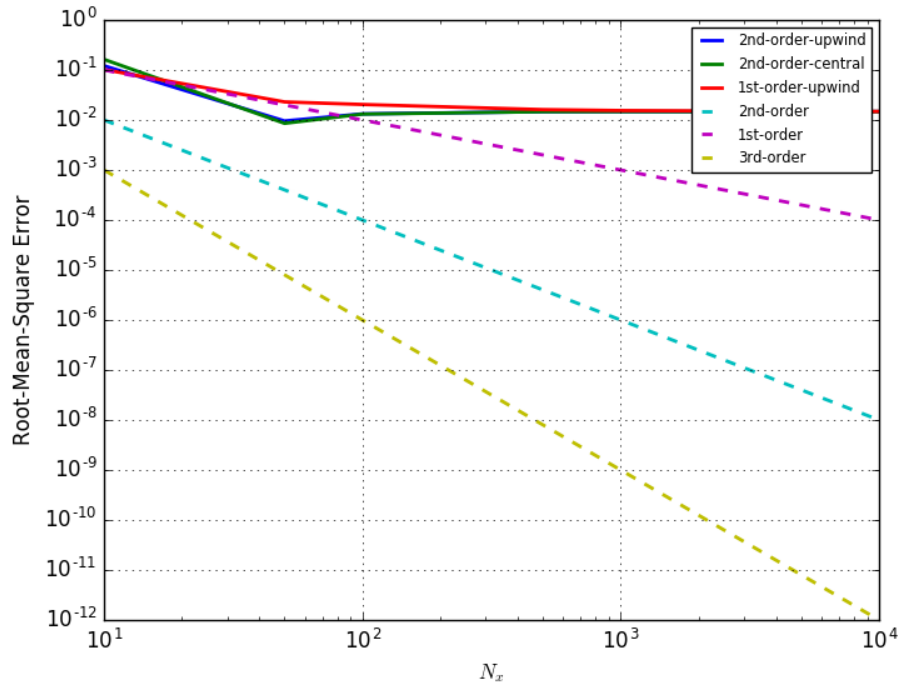


Figure 8: Root-Mean-Square Error at $t = T_f$ vs. Number of Points N

From Figure 9, which is the case of fixed small Δx with changing Δt , we can see that for the for all of those three schemes, the order of accuracy are all second order approximately. Because the order of accuracy of Crank-Nicolson is two.

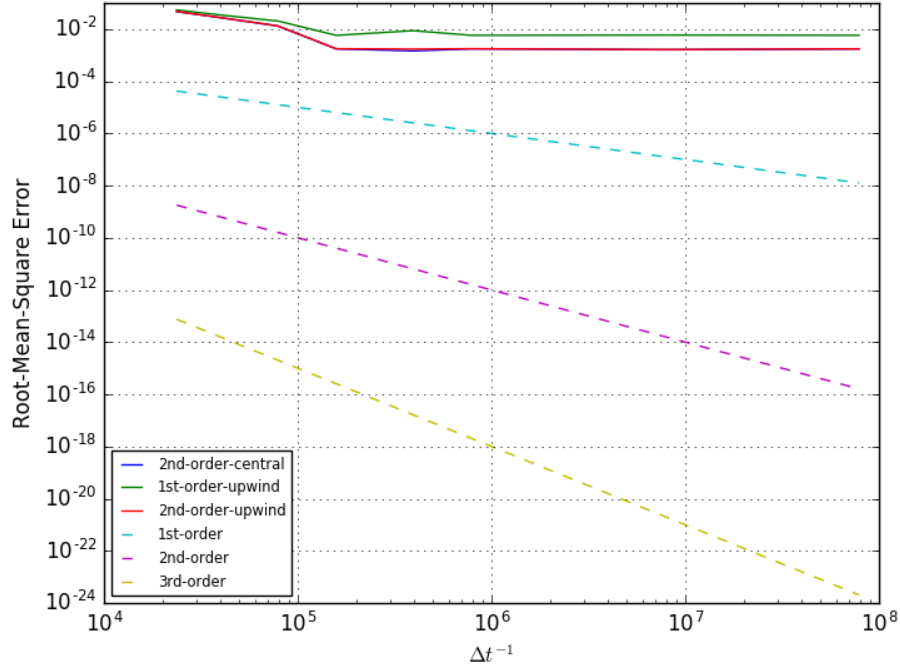


Figure 9: Root-Mean-Square Error at $t = T_f$ vs. δt

(d) Please see Figure 10 to 15 for the spy of \mathbf{A} and \mathbf{B} .

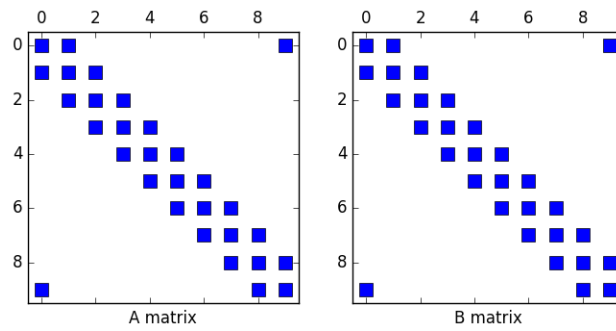


Figure 10: Spy Plot of \mathbf{A} and \mathbf{B} for 1st-order-upwind & Crank-Nicolson

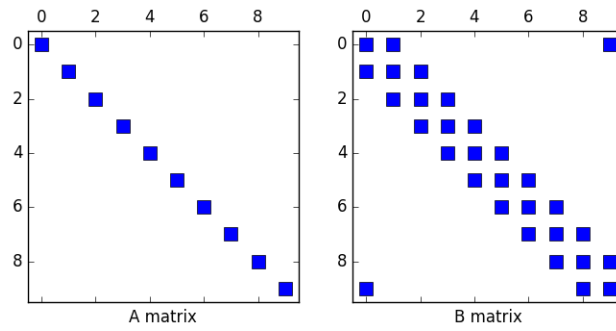


Figure 11: Spy Plot of \mathbf{A} and \mathbf{B} for 1st-order-upwind & Explicit-Euler

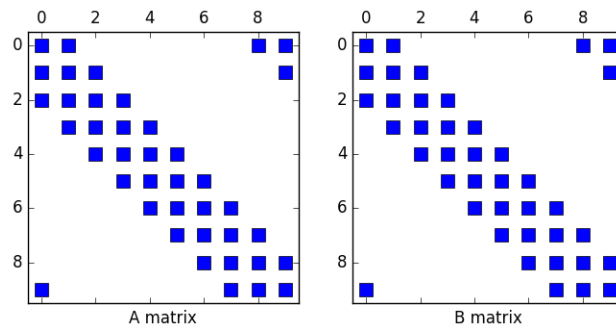


Figure 12: Spy Plot of \mathbf{A} and \mathbf{B} for 2nd-order-upwind & Crank-Nicolson

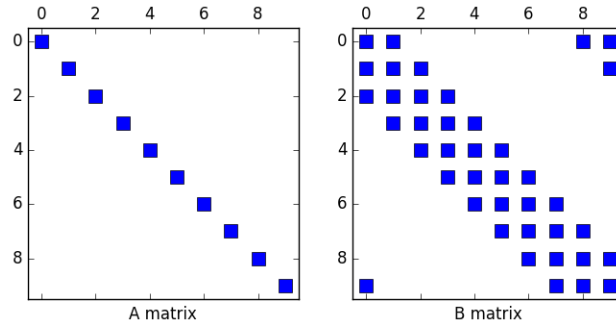


Figure 13: Spy Plot of **A** and **B** for 2nd-order-upwind & Explicit-Euler

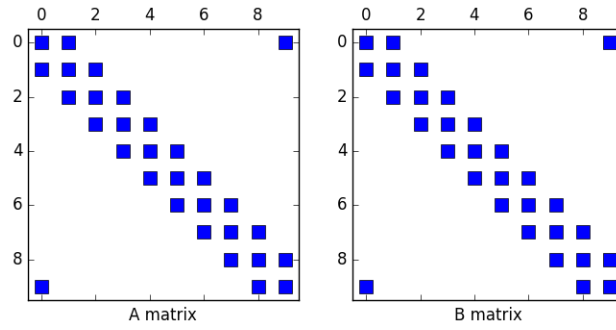


Figure 14: Spy Plot of **A** and **B** for 2nd-order-central & Crank-Nicolson

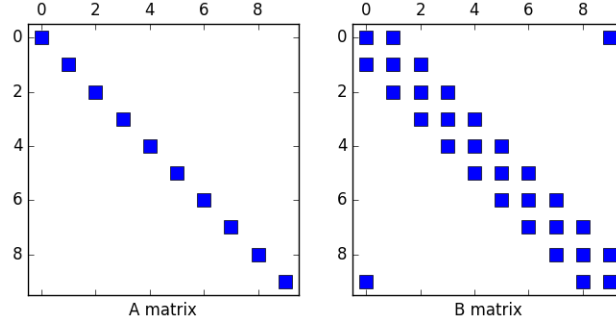


Figure 15: Spy Plot of **A** and **B** for 2nd-order-central & Explicit-Euler

(e) From Figure 16 to 18, we can see from the contour plots that the green region is the stable region, In green region, for Explicit Euler, the maximum λ is exactly 1 with all the other λ s below 1. And 2nd-order-central scheme has the largest stable region.

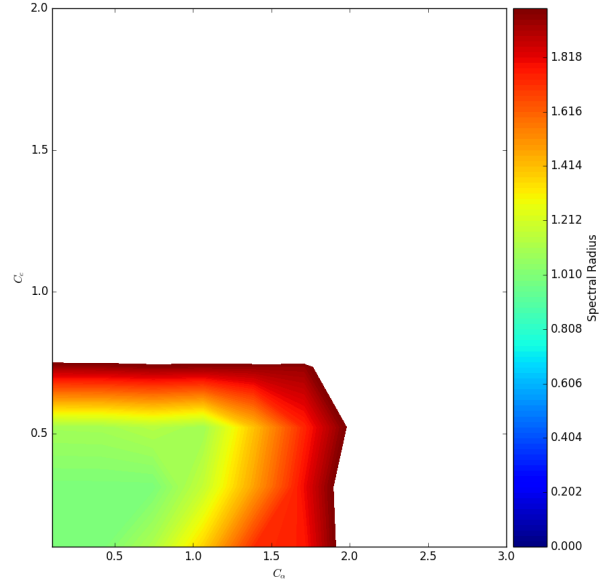


Figure 16: Flooded Iso-contour of Spectral Radius for 2nd-order-central

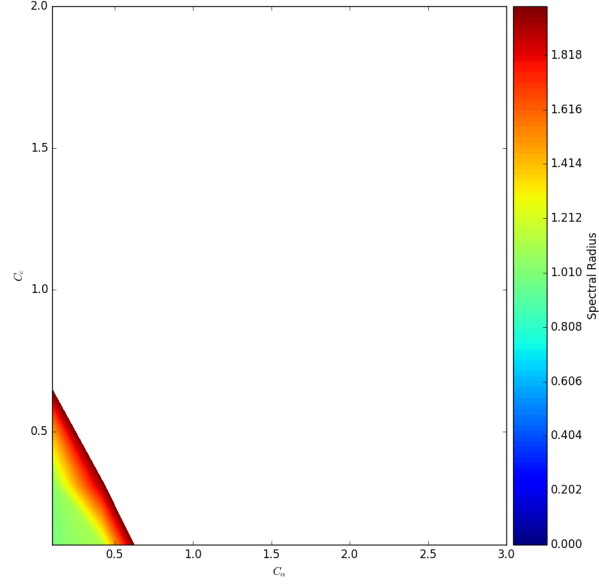


Figure 17: Flooded Iso-contour of Spectral Radius for 2nd-order-upwind

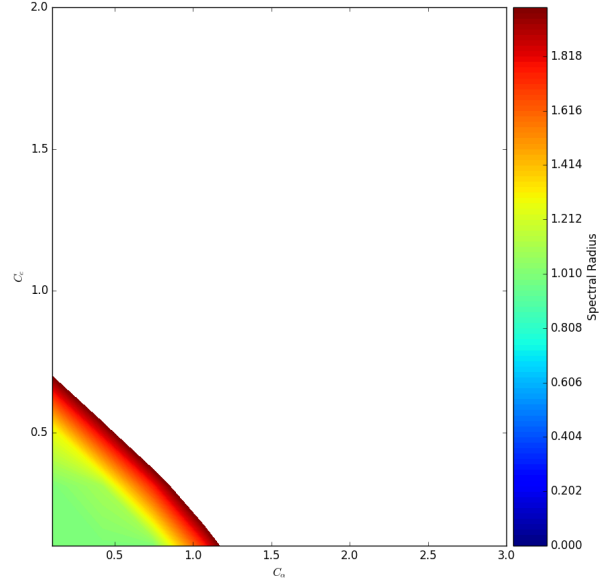


Figure 18: Flooded Iso-contour of Spectral Radius for 1st-order-upwind

Problem 3.

(a) From Figure 19, we can see the evolution from transient states to steady state which is the line with Final Time label, we can see that the solutions $u(x, t)$ all converge into a point on $u = 0$ at steady state, where that point can be defined as δ by $u(\delta) = 0.01a$.

For this original parameter combination, they are $L = 1$, $c = 10$, $\alpha = 2$, $\beta = 10$, $\omega = 100000$, $a = 1$. And we can get that the $\delta = 0.5102$ by set a tolerance of 1×10^{-6} and calculate the different between each δ for each transient state, and got the steady state when the δ difference is within the tolerance.

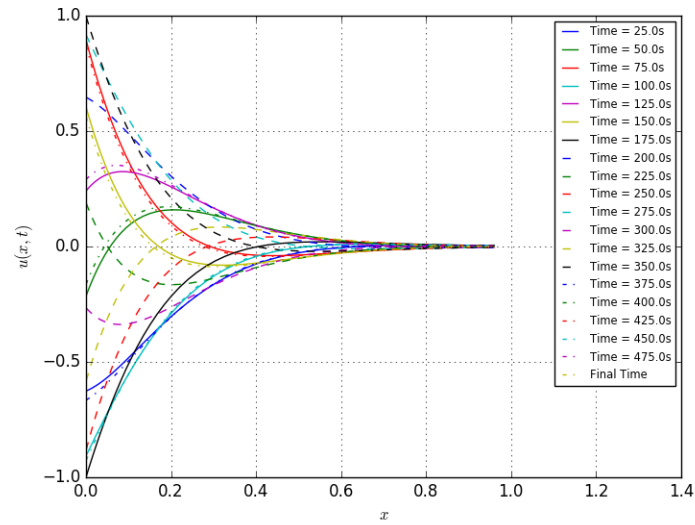


Figure 19: Transient and Steady State Solution

i) Change c from 10 to 20 and 30. c is the advection speed, so larger c will cause larger convection, which means the solution is moving faster and has bigger amplitude(convection input).

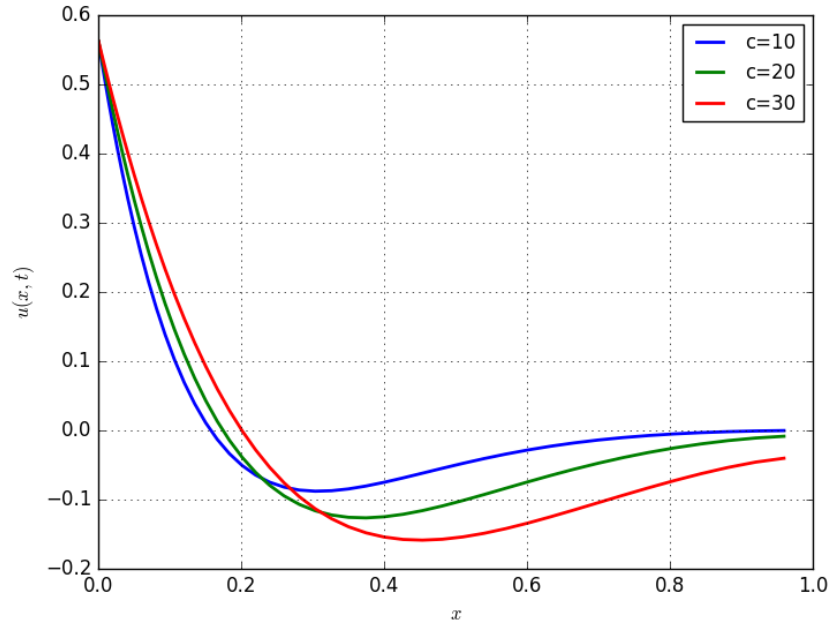


Figure 20: Showcase of changing c

ii) Change α from 2 to 5 and 10. α is the diffusive coefficient, so larger α will cause more obvious diffusive phenomenon and smaller amplitude.

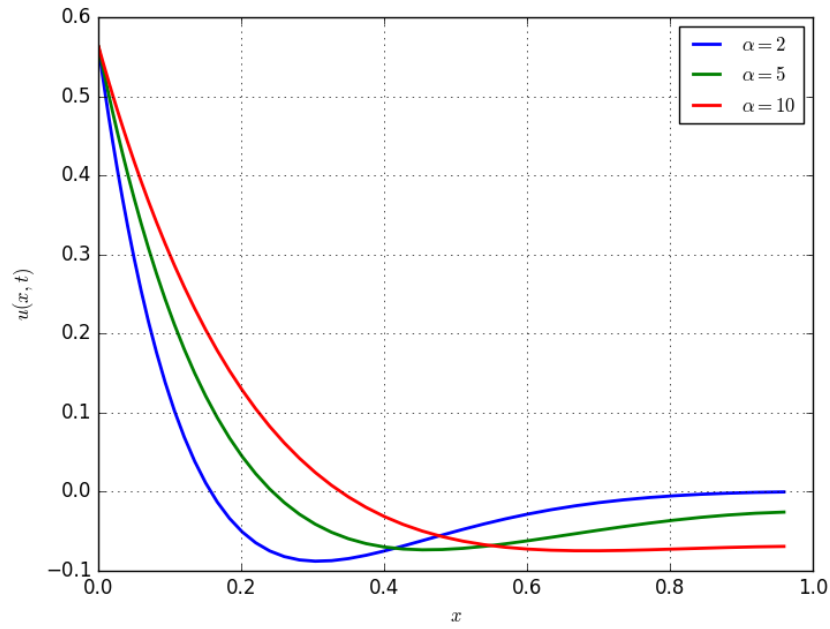


Figure 21: Showcase of changing α

iii) Change β from 10 to 100 and 1000. β is the damping coefficient. It controls how fast the curve reach the steady state, so larger β will damp faster and have smaller δ .

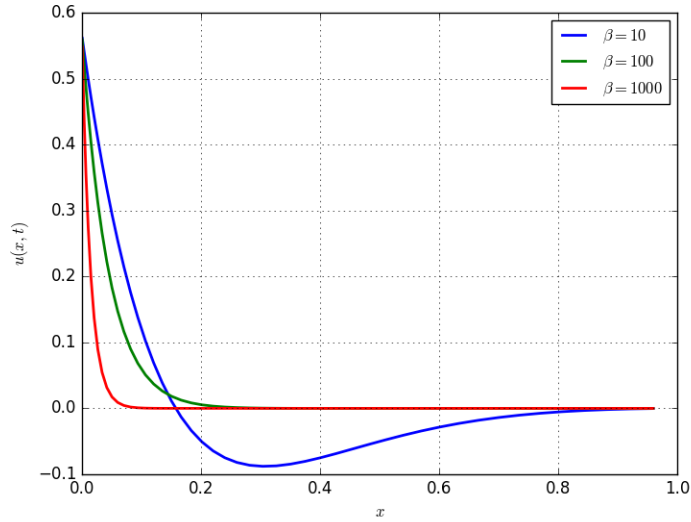


Figure 22: Showcase of changing β

iv) Change ω from 100000 to 10000 and 1000. ω is the angular frequency of the boundary condition. Larger ω will have smaller period and larger frequency.

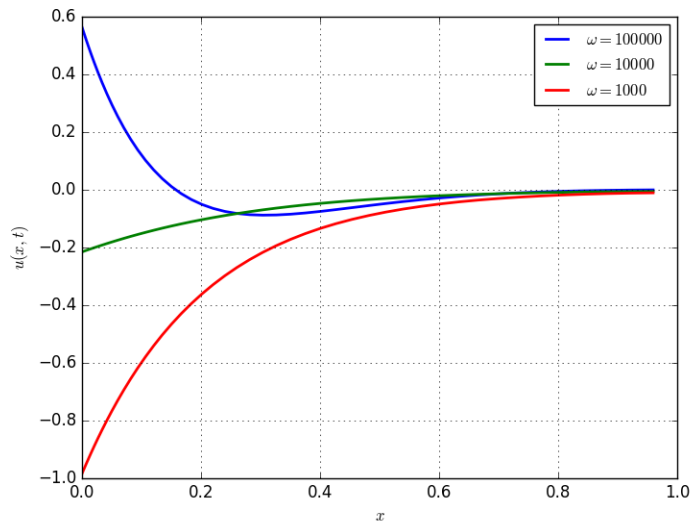


Figure 23: Showcase of changing ω

v) Change a from 1 to 2 and 3. a is the amplitude of the left bound boundary condition. It controls the upper and lower bounds of the solutions. As you can see, solutions are self-similar with bounds limited to a

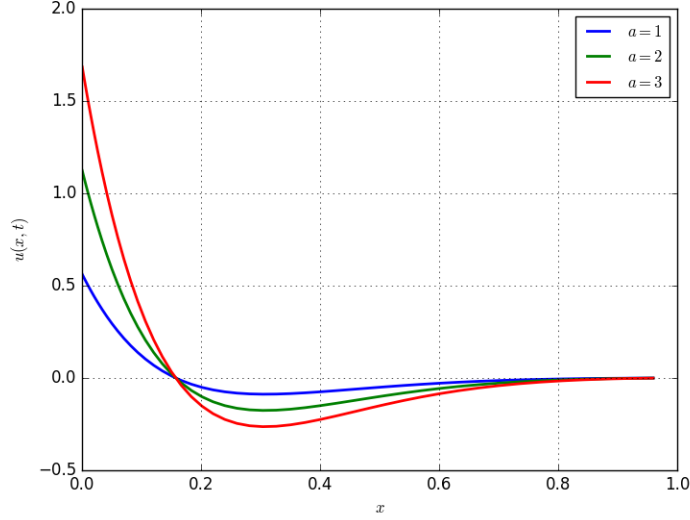


Figure 24: Showcase of changing a

(b) By adopting Π - theorem, we can get 5 Π groups, where $num_{var} - num_{dim} = 7 - 2 = 5$.

The two dimensions are L (length) and T (time).

For the seven parameters, $\delta = [L]$, $c = [\frac{L}{T}]$, $\alpha = [\frac{L^2}{T}]$, $\beta = [\frac{1}{T}]$, $L = [L]$, $a = [\frac{L}{T}]$, $\omega = [\frac{1}{T}]$

First, we want to eliminate $L = [L]$, so $\frac{\delta}{L} = [1]$ (No dimension), $\frac{c}{L} = [\frac{1}{T}]$, $\frac{\alpha}{L^2} = [\frac{1}{T}]$, $\beta = [\frac{1}{T}]$, $\frac{a}{L} = [\frac{1}{T}]$, $\omega = [\frac{1}{T}]$.

Then, we want to eliminate $\omega = [\frac{1}{T}]$, then $\frac{\delta}{L} = [1]$, $\frac{c}{L\omega} = [1]$, $\frac{\alpha}{L^2\omega} = [1]$, $\frac{\beta}{\omega} = [1]$, $\frac{a}{L\omega} = [1]$.

So we can get: $\frac{\delta}{L} = f(\frac{c}{L\omega}, \frac{\alpha}{L^2\omega}, \frac{\beta}{\omega}, \frac{a}{L\omega})$

From Figure 25, we can see that by change L linearly with slope of 2 and keep all other 4 Π groups constant, we can get δ changing linearly with the same slope as L .

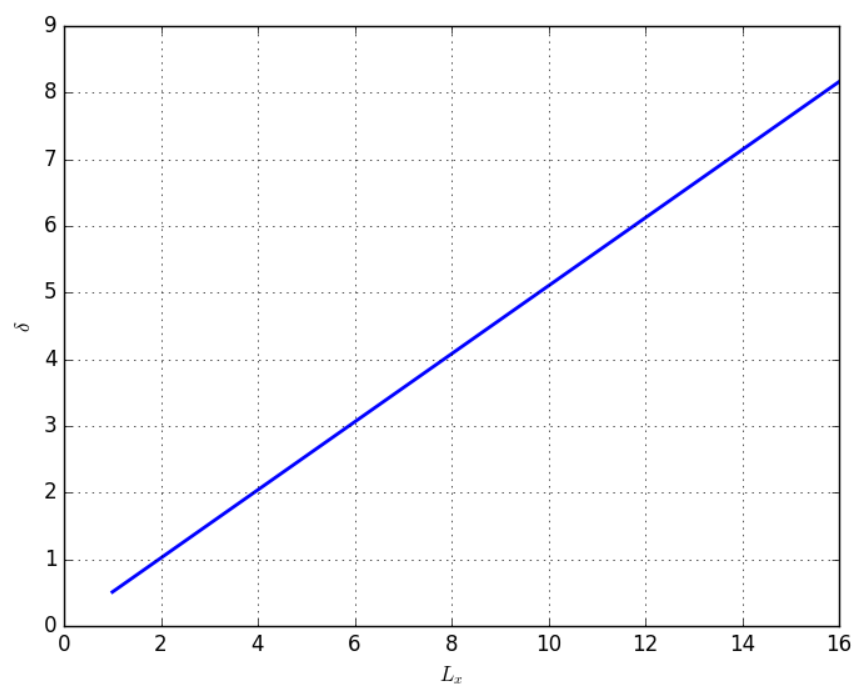


Figure 25: Relationship between δ and L_x with the other Π groups constant