Introduction

In this document, we will be precribing the template to solve the Kuramoto-Sivashinsky (K-S) equation. In this instance, the stencil in space will be fourth (4th) order, and for the time integration, we will be using the fourth (4th) order Runge-Kutta method.

The K-S equation is:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + \alpha\frac{\partial^2 u}{\partial x^2} + \beta\frac{\partial^3 u}{\partial x^3} + \gamma\frac{\partial^4 u}{\partial x^4} = 0$$

In [1]:
```
import sys
import os
import time
import numpy as np
import matplotlib.pyplot as plt

# Add the directory containing your module to sys.path
module_path = os.path.abspath(os.path.join('..', r"A:\Users\mtthl\Documents\Education\ME5653_C
sys.path.append(module_path)

from distributedObjects import *
from distributedFunctions import *
```

Spatial Stencil

The spatial stencil to be 4th order, and thus take four (4) points plus the sampling point, thus as below for interior points.

-X-------X-------X--------X-------X-

i-2_____i-1_____i______i+1____i+2

For the boundary points, the stencil will be:

X-------X-------X-------X-------X-

i______i+1___i+2____i+3____i+4

-X-------X-------X-------X-------X

i-4_____i-3____i-2_____i-1_____i

Thus, the formulation for the spatial stencil in the 1st derivative becomes

In [2]:
```
first_interior_gradient = numericalGradient( 1 , ( 2 , 2 ) )
first_interior_gradient.coeffs
```

```
Out[2]: array([ 8.33333333e-02, -6.66666667e-01,  1.11022302e-16,  6.66666667e-01,
               -8.33333333e-02])
```

```
In [3]: first_LHS_gradient = numericalGradient( 1 , ( 0 , 4 ) )
        first_LHS_gradient.coeffs
```

```
Out[3]: array([-2.08333333,  4.        , -3.        ,  1.33333333, -0.25      ])
```

```
In [4]: first_RHS_gradient = numericalGradient( 1 , ( 4 , 0 ) )
        first_RHS_gradient.coeffs
```

```
Out[4]: array([ 0.25      , -1.33333333,  3.        , -4.        ,  2.08333333])
```

Interior 1st Derivative - 4th Order

$$\frac{\partial \phi}{\partial x} = \frac{\frac{1}{12}\phi_{i-2} - \frac{1}{6}\phi_{i-1} + \frac{1}{6}\phi_{i+1} - \frac{1}{12}\phi_{i+2}}{\Delta x}$$

Boundary LHS 1st Derivative - 4th Order

$$\frac{\partial \phi}{\partial x} = \frac{\frac{25}{12}\phi_i + 4\phi_{i+1} - 3\phi_{i+2} + \frac{4}{3}\phi_{i+3} - \frac{1}{4}\phi_{i+4}}{\Delta x}$$

Boundary RHS 1st Derivative - 4th Order

$$\frac{\partial \phi}{\partial x} = \frac{\frac{1}{4}\phi_{i-4} - \frac{4}{3}\phi_{i-3} + 3\phi_{i-2} - 4\phi_{i-1} + \frac{25}{12}\phi_i}{\Delta x}$$

The 2nd derivative becomes

In [5]:
```
second_interior_gradient = numericalGradient( 2 , ( 2 , 2 ) )
second_interior_gradient.coeffs
```

Out[5]:
```
array([-0.08333333,  1.33333333, -2.5       ,  1.33333333, -0.08333333])
```

In [7]:
```
second_LHS_gradient = numericalGradient( 2 , ( 0 , 4 ) )
second_LHS_gradient.coeffs
```

Out[7]:
```
array([ 2.91666667, -8.66666667,  9.5       , -4.66666667,  0.91666667])
```

In [9]:
```
second_RHS_gradient = numericalGradient( 2 , ( 4 , 0 ) )
second_RHS_gradient.coeffs
```

Out[9]: array([0.91666667, -4.66666667, 9.5   , -8.66666667, 2.91666667])

Interior 2nd Derivative - 4th Order

$$\frac{\partial^2 \phi}{\partial x^2} = \frac{-\frac{1}{12}\phi_{i-2} + \frac{4}{3}\phi_{i-1} - \frac{5}{2}\phi_i + \frac{4}{3}\phi_{i+1} - \frac{1}{12}\phi_{i+2}}{\Delta x^2}$$

Boundary LHS 2nd Derivative - 4th Order

$$\frac{\partial^2 \phi}{\partial x^2} = \frac{\frac{35}{12}\phi_i - \frac{26}{3}\phi_{i+1} + \frac{19}{2}\phi_{i+2} - \frac{14}{3}\phi_{i+3} + \frac{11}{12}\phi_{i+4}}{\Delta x^2}$$

Boundary RHS 2nd Derivative - 4th Order

$$\frac{\partial^2 \phi}{\partial x^2} = \frac{\frac{11}{12}\phi_{i-4} - \frac{14}{3}\phi_{i-3} + \frac{19}{2}\phi_{i-2} - \frac{26}{3}\phi_{i-1} + \frac{35}{12}\phi_i}{\Delta x^2}$$

Even though we will not be using the 3rd derivative, here it is

In [10]:
```
third_interior_gradient = numericalGradient( 3 , ( 2 , 2 ) )
third_interior_gradient.coeffs
```

Out[10]: array([-0.5, 1. , 0. , -1. , 0.5])

In [11]:
```
third_LHS_gradient = numericalGradient( 3 , ( 0 , 4 ) )
third_LHS_gradient.coeffs
```

Out[11]: array([-2.5,   9. , -12. ,   7. , -1.5])

In [12]:
```
third_RHS_gradient = numericalGradient( 3 , ( 4 , 0 ) )
third_RHS_gradient.coeffs
```

Out[12]: array([1.5, -7. , 12. , -9. , 2.5])

Interior 3rd Derivative - 4th Order

$$\frac{\partial^3 \phi}{\partial x^3} = \frac{\frac{1}{2}(\phi_{i+2} - \phi_{i-2}) - (\phi_{i+1} - \phi_{i-1})}{\Delta x^3}$$

Boundary LHS 3rd Derivative - 4th Order

$$\frac{\partial^3 \phi}{\partial x^3} = \frac{-\frac{5}{2}\phi_i + 9\phi_{i+1} - 12\phi_{i+2} + 7\phi_{i+3} - \frac{3}{2}\phi_{i+4}}{\Delta x^3}$$

Boundary RHS 3rd Derivative - 4th Order

$$\frac{\partial^3 \phi}{\partial x^3} = \frac{\frac{3}{2}\phi_{i-4} - 7\phi_{i-3} + 12\phi_{i-2} - 9\phi_{i-1} + \frac{5}{2}\phi_i}{\Delta x^3}$$

Finally, the fourth (4th) derivative will be

In [13]:
```
fourth_interior_gradient = numericalGradient( 4 , ( 2 , 2 ) )
fourth_interior_gradient.coeffs
```

Out[13]: array([1., -4., 6., -4., 1.])

In [14]:
```
fourth_LHS_gradient = numericalGradient( 4 , ( 0 , 4 ) )
fourth_LHS_gradient.coeffs
```

Out[14]: array([1., -4., 6., -4., 1.])

In [15]:
```
third_RHS_gradient = numericalGradient( 4 , ( 4 , 0 ) )
third_RHS_gradient.coeffs
```

Out[15]: array([1., -4., 6., -4., 1.])

Interior 4th Derivative - 4th Order

$$\frac{\partial^4 \phi}{\partial x^4} = \frac{\phi_{i-2} - 4\phi_{i-1} + 6\phi_i - 4\phi_{i+1} + \phi_{i+2}}{\Delta x^4}$$

Boundary LHS 4th Derivative - 4th Order

$$\frac{\partial^4 \phi}{\partial x^4} = \frac{\phi_i - 4\phi_{i+1} + 6\phi_{i+2} - 4\phi_{i+3} + \phi_{i+4}}{\Delta x^4}$$

Boundary RHS 4th Derivative - 4th Order

$$\frac{\partial^4 \phi}{\partial x^4} = \frac{\phi_{i-4} - 4\phi_{i-3} + 6\phi_{i-2} - 4\phi_{i-1} + \phi_i}{\Delta x^4}$$

Time Integration Method

As mentioned before, we will be using the Runge-Kutta 4th order time integration scheme (RK4). I chose this one because it seems to be more widely used, and thus would like to get some experience with it.

As formulated in [1], the RK4 method comes down to a central equation as follows:

$$\phi^{n+1} = \phi^n + \frac{\Delta t}{6}\left(R^n + 2R^{(1)} + 2R^{(2)} + R^{(3)}\right)$$

where R is the time derivative linear operator that is a function of u and ϕ. All values in the parenthesis in superscript represent a virtual step between the time steps. The process to find the R-values is:

1. $\phi^{(1)} = \phi^n + \frac{\Delta t}{2} R^n$
2. $\phi^{(2)} = \phi^n + \frac{\Delta t}{2} R^{(1)}$
3. $\phi^{(3)} = \phi^n + \Delta t R^{(2)}$

The $R^{(3)}$ value comes from the time derivative value that corresponds to $\phi^{(3)}$.

Works Cited

1. Anderson, D. A., Tannehill, J. C., Pletcher, R. H., Munipalli, R., and Shankar, V. (2021). Computational Fluid Mechanics and Heat Transfer. 4th Edition. CRC Press.