Finite Difference Method V.S. Spectral Method in Stationary Burger's Equation and Troesch Equation

第26組：

B05201013 陳昱銘

B04703045 蔡宜庭

B06901093 李青芳

Summary

1. Problem
   - Burger's Equation
   - Troesch Equation
2. Method & Implementation
   - Finite Difference
   - Spectral Method
3. Results & Analysis

1. Problem

Burger's Equation

Solve differential equation

$$au'' + buu' + cu = 0, \; -1 \le x \le 1$$

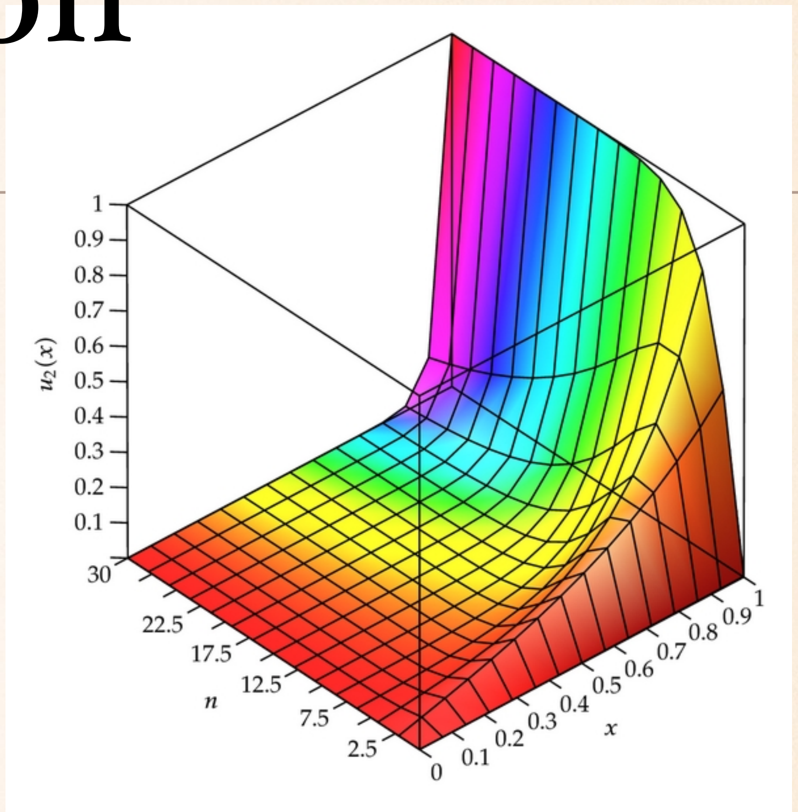
with boundary conditions

$$u(-1) = \alpha, \;\; u(1) = \beta$$

Choose real parameters a, c, α, β

with $b = 0$ and $b \neq 0$ (nonlinear term)

Troesch Equation

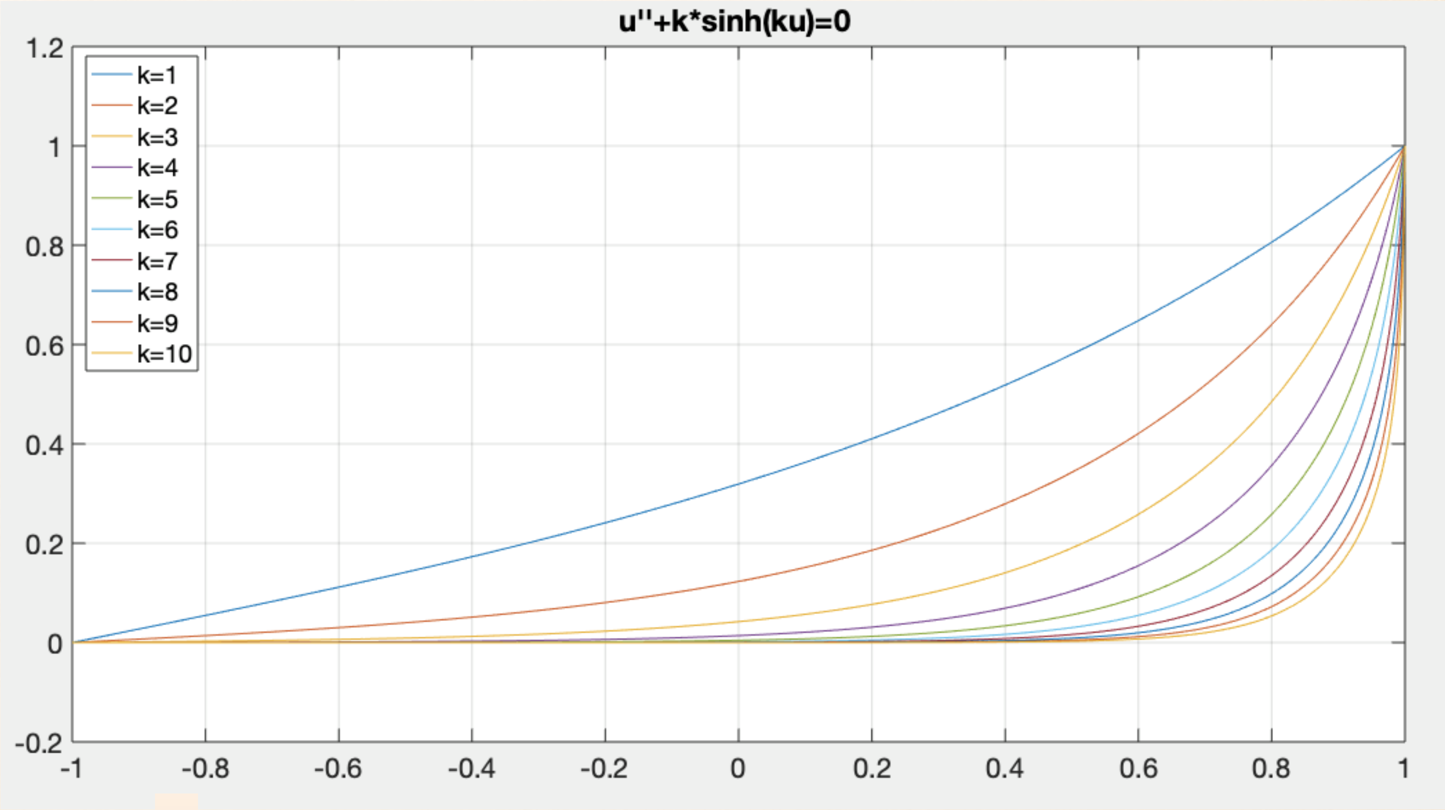


Solve differential equation

$$u'' - ksinh(ku) = 0,\ \ 0 \leq x \leq 1$$

with boundary conditions

$$u(0) = \alpha,\ \ u(1) = \beta$$



2. Method & Implementation

Method

- Chebfun
- Differential matrix method
  1. Compute differential matrices
     - Finite difference
     - Spectral method
       - Uniform distance points
       - Chebyshev points
  2. Solve the system of equations
     - Newton method

Finite Difference

$$u'(x_n) \approx \frac{u(x_{n+1}) - u(x_{n-1})}{2h}$$

First order differential matrix

$$D_1 = \frac{1}{2h} \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ -1 & 0 & 1 & 0 & \dots & 0 \\ 0 & -1 & 0 & 1 & & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & & -1 & 0 & 1 \\ 0 & \dots & & & -1 & 0 \end{bmatrix}$$

Finite Difference

$$u''(x_n) \approx \frac{u(x_{n+1}) - 2u(x_n) + u(x_{n-1})}{h^2}$$

Second order differential matrix

$$D_2 = \frac{1}{h^2}\begin{bmatrix} -2 & 1 & 0 & 0 & \dots & 0 \\ 1 & -2 & 1 & 0 & \dots & 0 \\ 0 & 1 & -2 & 1 & & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & & 1 & -2 & 1 \\ 0 & \dots & & & 1 & -2 \end{bmatrix}$$

Spectral Method

Lagrange interpolation polynomial

$$p_n(x) = \sum_{j=0}^{n} l_j(x) u_j \quad \text{, where} \quad l_j(x) = \prod_{\substack{k=0 \\ k \neq j}}^{n} \frac{x - x_k}{x_j - x_k}$$

Barycentric form

$$l_j(x) = \frac{\dfrac{\lambda_j}{x - x_j}}{\displaystyle\sum_{k=0}^{n} \frac{\lambda_k}{x - x_k}} \quad \text{, where} \quad \lambda_j = \frac{1}{\displaystyle\prod_{\substack{k=0 \\ k \neq j}}^{n} (x_j - x_k)}$$

Spectral Method

First order differential matrix D_1

$$D_1 : D_{ij}^{(1)} = l_j'(x_i) = \begin{cases} \dfrac{\lambda_j/\lambda_i}{x_i - x_j}, i \neq j \\ -\sum\limits_{\substack{k=0 \\ k \neq i}}^{n} l_k'(x_i), i = j \end{cases}$$

Spectral Method

Second order differential matrix D_2

$$D_2 : D_{ij}^{(2)} = l_j''(x_i) = \begin{cases} \dfrac{-2\lambda_j/\lambda_i}{x_i - x_j}\left(\displaystyle\sum_{\substack{k=0 \\ k \neq i}}^{n} \dfrac{\lambda_k/\lambda_i}{x_i - x_k} + \dfrac{1}{x_i - x_j} \right) = 2D_{ij}^{(1)}\left(D_{ii}^{(1)} - \dfrac{1}{x_i - x_j}\right), i \neq j \\ -\displaystyle\sum_{\substack{k=0 \\ k \neq i}}^{n} l_k''(x_i), i = j \end{cases}$$

Construct Points

- Uniform distance points

$$x_j = -1 + jh \ \ for\ j = 0,1,\ldots,n; \ \ h = \frac{2}{n}$$

- Chebyshev points

$$x_j = -cos(j\pi/n), \ \ j = 0,1,\ldots,n$$

Differential Matrices of Spectral Method

1. Given a set of nodes $\{x_i\}_1^n$
2. Construct λ_j
3. Construct D1, D2

Compare D1*D1 with D2

D1*D1 and D2 are close when n small

```
x = [1,2,3]
D1 = construct_D1(x)
D2 = construct_D2(x,D1)
```

```
D1:
 [[-1.5  2.  -0.5]
 [-0.5 -0.   0.5]
 [ 0.5 -2.   1.5]]
D2:
 [[ 1. -2.  1.]
 [ 1. -2.  1.]
 [ 1. -2.  1.]]
D1*D1:
 [[ 1. -2.  1.]
 [ 1. -2.  1.]
 [ 1. -2.  1.]]
```

```
x = np.linspace(0,100,10)
D1 = construct_D1(x)
D2 = construct_D2(x,D1)
D1D1=D1@D1
# print('D1:\n', D1)
# print('D2:\n', D2)
# print('D1*D1:\n', D1D1)
# print('D2-D1*D1:\n', D2-D1D1)
print('abs(D2 - D1*D1)<1e-9\n', abs(D2-D1D1)<1e-9)
```

```
(D2 - D1*D1)<1e-9
 [[ True  True  True  True  True  True  True  True  True  True]
 [ True  True  True  True  True  True  True  True  True  True]
 [ True  True  True  True  True  True  True  True  True  True]
 [ True  True  True  True  True  True  True  True  True  True]
 [ True  True  True  True  True  True  True  True  True  True]
 [ True  True  True  True  True  True  True  True  True  True]
 [ True  True  True  True  True  True  True  True  True  True]
 [ True  True  True  True  True  True  True  True  True  True]
 [ True  True  True  True  True  True  True  True  True  True]
 [ True  True  True  True  True  True  True  True  True  True]]
```

```
x = np.linspace(0,100, 50)
D1 = construct_D1(x)
D2 = construct_D2(x,D1)
D1D1=D1@D1
# print('D1:\n', D1)
# print('D2:\n', D2)
# print('D1*D1:\n', D1D1)
# print('D2-D1*D1:\n', D2-D1D1)
print('abs(D2 - D1*D1)<1e-9\n', abs(D2-D1D1)<1e-9)
```

```
abs(D2 - D1*D1)<1e-9
 [[False False False ..., False False False]
 [False False False ..., False False False]
 [False False False ..., False False False]
 ...,
 [False False False ..., False False False]
 [False False False ..., False False False]
 [False False False ..., False False False]]
```

Convert the Differential Eq to System of Eqs

$$au'' + buu' + cu = 0$$

$$U = \begin{bmatrix} u_0 \\ \vdots \\ u_n \end{bmatrix} \quad \text{where} \quad u_i \quad \text{is} \quad \text{an} \quad \text{approximation} \quad \text{of} \quad u(x_i)$$

$$D_1 \begin{bmatrix} u_0 \\ \vdots \\ u_n \end{bmatrix} \quad \text{approx.} \quad \text{of} \quad u' \qquad D_2 \begin{bmatrix} u_0 \\ \vdots \\ u_n \end{bmatrix} \quad \text{approx.} \quad \text{of} \quad u''$$

$$\Rightarrow aD_2 \begin{bmatrix} u_0 \\ \vdots \\ u_n \end{bmatrix} + b \begin{bmatrix} u_0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & u_n \end{bmatrix} D_1 \begin{bmatrix} u_0 \\ \vdots \\ u_n \end{bmatrix} + c \begin{bmatrix} u_0 \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

Convert the Differential Eq to System of Eqs

apply B.C. $u_0 = \alpha, \; u_n = \beta$

$$aD_2 \begin{bmatrix} u_0 \\ \vdots \\ u_n \end{bmatrix} + b \begin{bmatrix} u_0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & u_n \end{bmatrix} D_1 \begin{bmatrix} u_0 \\ \vdots \\ u_n \end{bmatrix} + c \begin{bmatrix} u_0 \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$y = \begin{bmatrix} u_1 \\ \vdots \\ u_{n-1} \end{bmatrix}$$

$$\Rightarrow aD_2 \begin{bmatrix} \alpha \\ y \\ \beta \end{bmatrix} + b\, diag[\alpha\, y\, \beta] D_1 \begin{bmatrix} \alpha \\ y \\ \beta \end{bmatrix} + c \begin{bmatrix} \alpha \\ y \\ \beta \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

Convert the Differential Eq to System of Eqs

$$aD_2 \begin{bmatrix} \alpha \\ y \\ \beta \end{bmatrix} + b\,diag[\alpha\, y\, \beta]D_1 \begin{bmatrix} \alpha \\ y \\ \beta \end{bmatrix} + c \begin{bmatrix} \alpha \\ y \\ \beta \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

let $B =$ middle of D_1, and $A =$ middle of D_2

$$F(y) = aAy + b\,diag[y]By + cy + constant$$

```
function x = F_value(D_2,D_1,y,a,b,c,alpha,beta)
length = size(y,1);
B = D_1(2:length+1,2:length+1);
A = D_2(2:length+1,2:length+1);
U = zeros(length);
for i = 1:length
    U(i,i)=y(i);
end
constant = a*D_2(2:length+1,1)*alpha+a*D_2(2:length+1,length+2)*beta+b*D_1(2:length+1,1)*alpha+b*D_1(2:length+1,length+2)*beta;
x = a*A*y + b*U*B*y + c*y + constant;
```

Solve the System of Equations

- Newton method

Goal: solve F(y) = 0

$$\partial F(y_{k+1} - y_k) = -F(y_k)$$

$$\Rightarrow y_{k+1} = y_k - (\partial F)^{-1} F(y_k)$$

```
function x=Jacobian(D_2,D_1,y,a,b,c)
length = size(y,1);
U = zeros(length);
C = zeros(length);
B = D_1(2:length+1,2:length+1);
A = D_2(2:length+1,2:length+1);
for i = 1:length
    U(i,i)=y(i);
    C(i,i)=B(i,:)*y;
end
x = a*A+b*(C+U*B)+c*eye(length);
```

$$F(y) = aAy + b\,diag[y]By + cy + constant$$

$$C = diag[By_k]$$

$$\partial F = aA + b(C + diag[y_k]B) + cI$$

3. Results & Analysis

Burger's Equation with $b = 0$

General Solution

$$u'' + u = 0, \; -1 \leq x \leq 1, \; u(-1) = \alpha, \;\; u(1) = \beta$$

$$u(x) = C_1 cosx + C_2 sinx$$

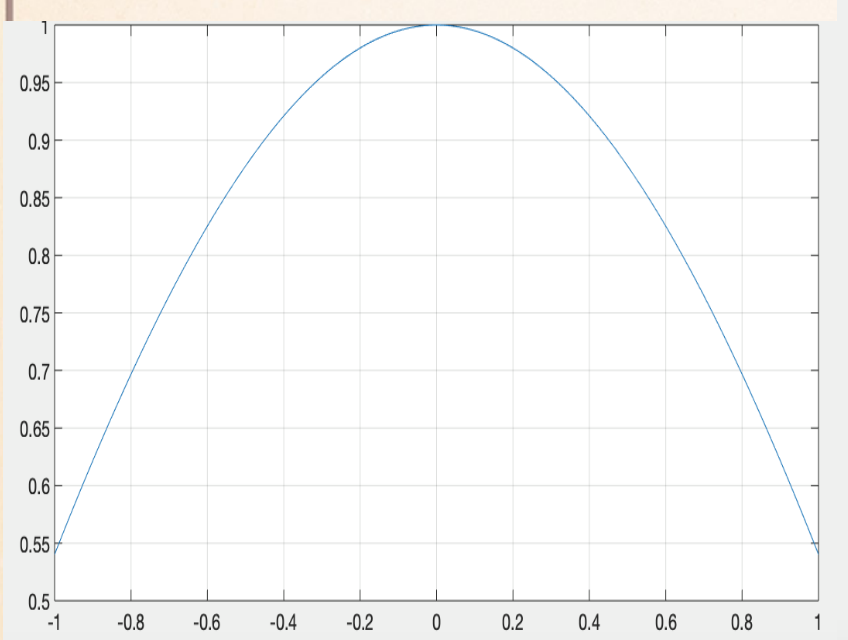
$$C_1 cos(-1) + C_2 sin(-1) = \alpha$$

$$C_1 cos(1) + C_2 sin(1) = \beta$$

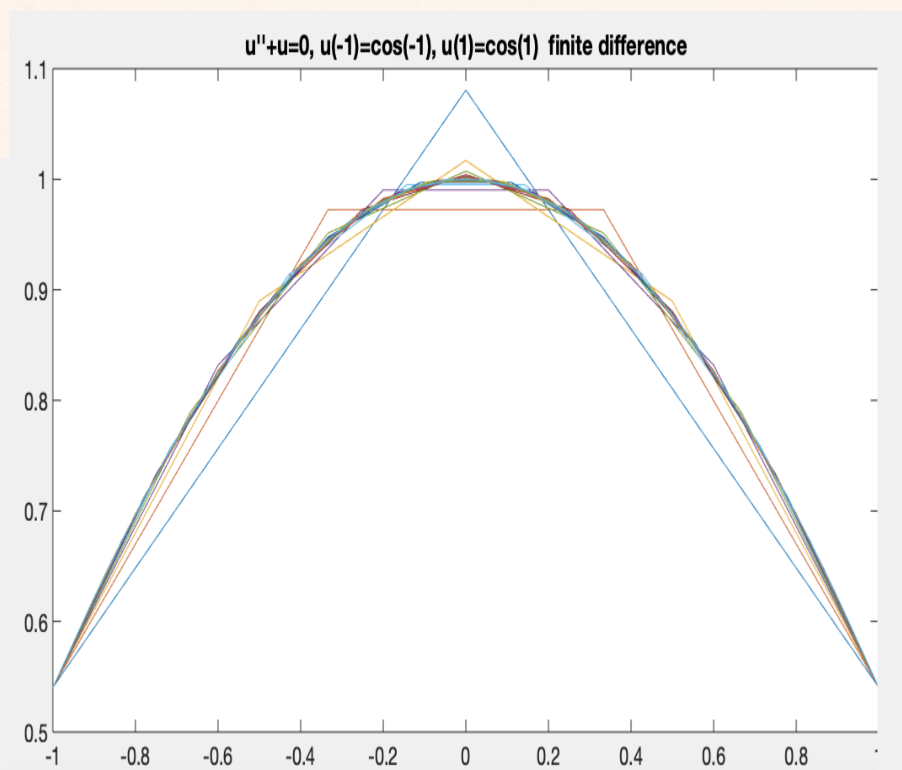
Graph of Solution

$$u'' + u = 0, \ -1 \le x \le 1, \ u(-1) = \alpha, \ \ u(1) = \beta$$

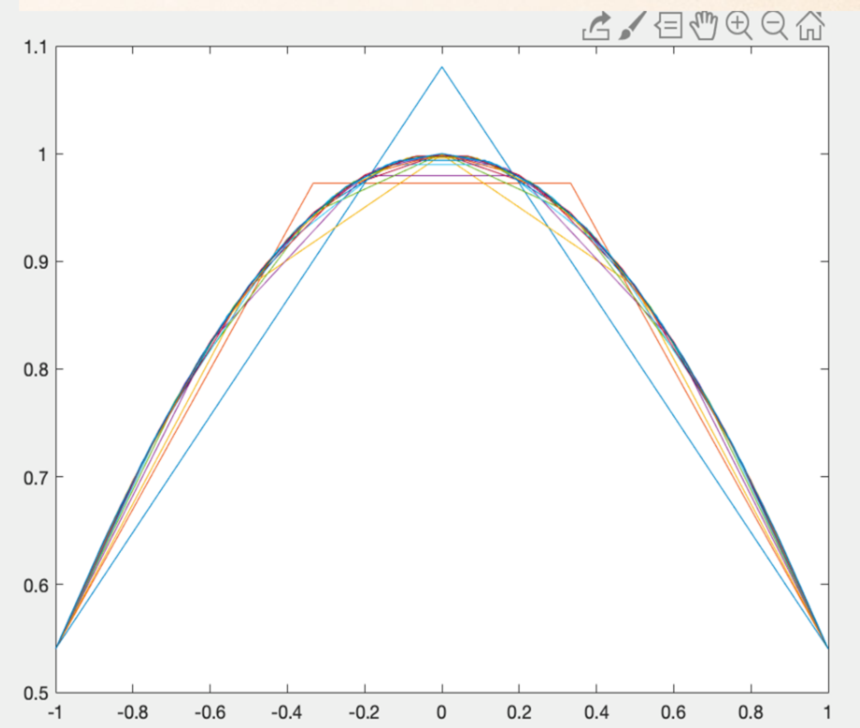
$$\alpha = cos(-1), \ \ \beta = cos(1)$$



Chebfun



Finite Difference

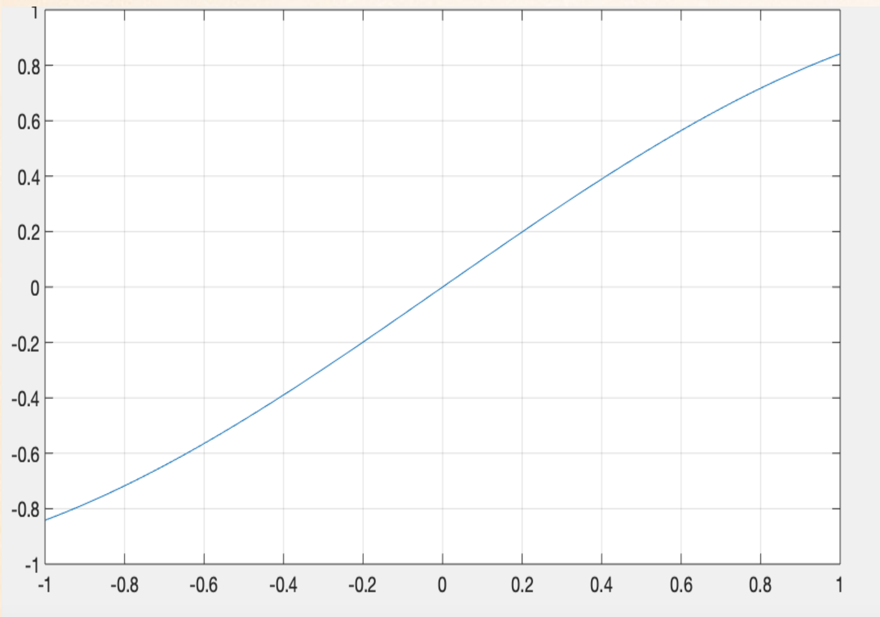


Spectral Method
with uniform nodes

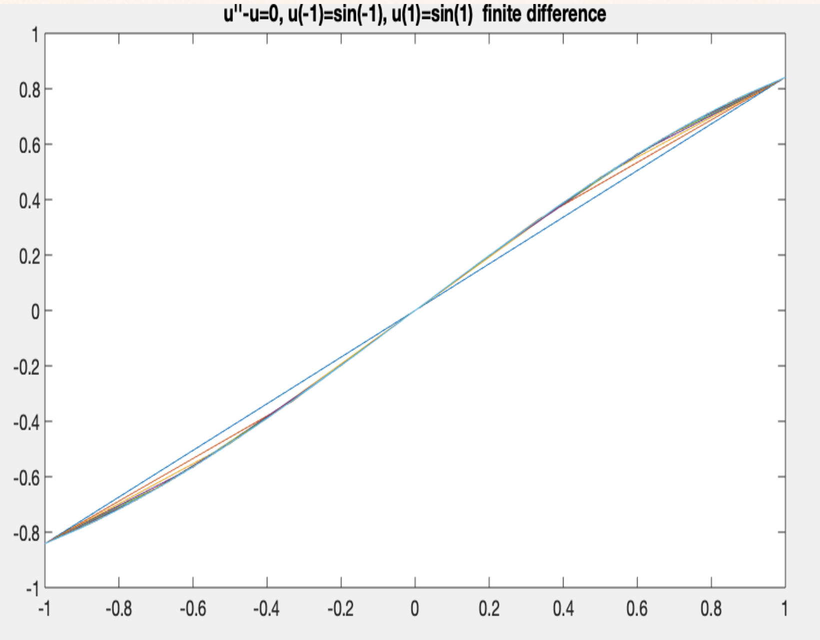
Graph of Solution

$$u'' + u = 0,\ -1 \le x \le 1,\ u(-1) = \alpha,\ \ u(1) = \beta$$

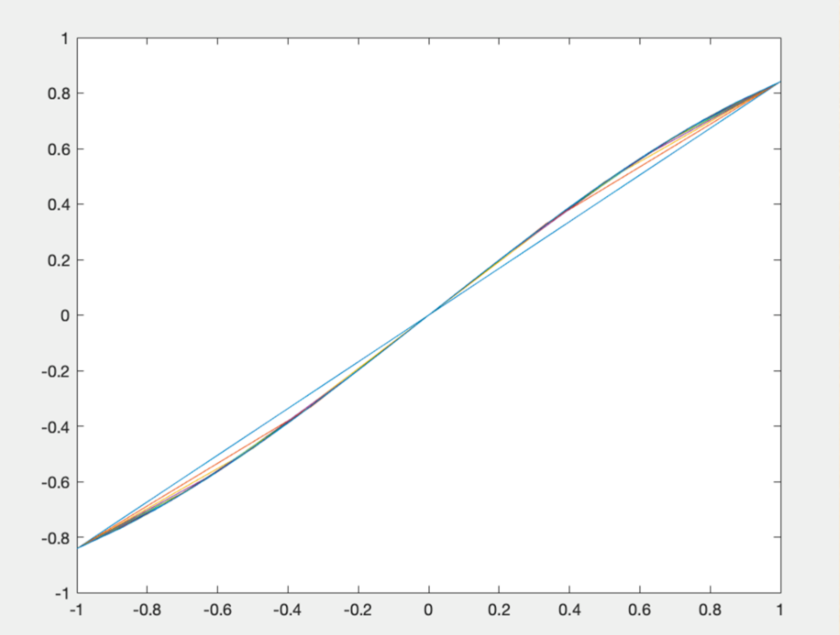
$$\alpha = sin(-1),\ \ \beta = sin(1)$$



Chebfun



Finite Difference

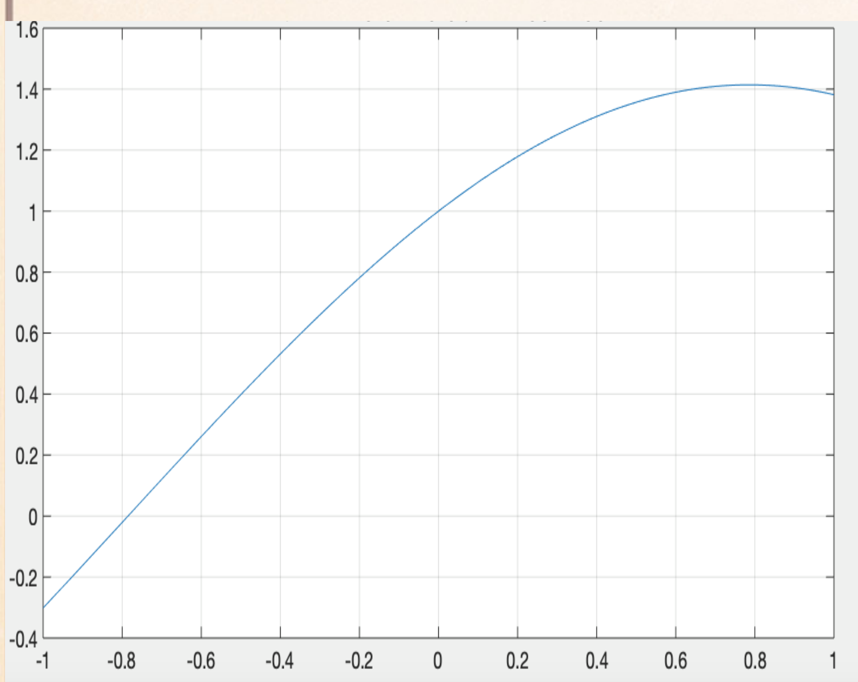


Spectral Method
with uniform nodes

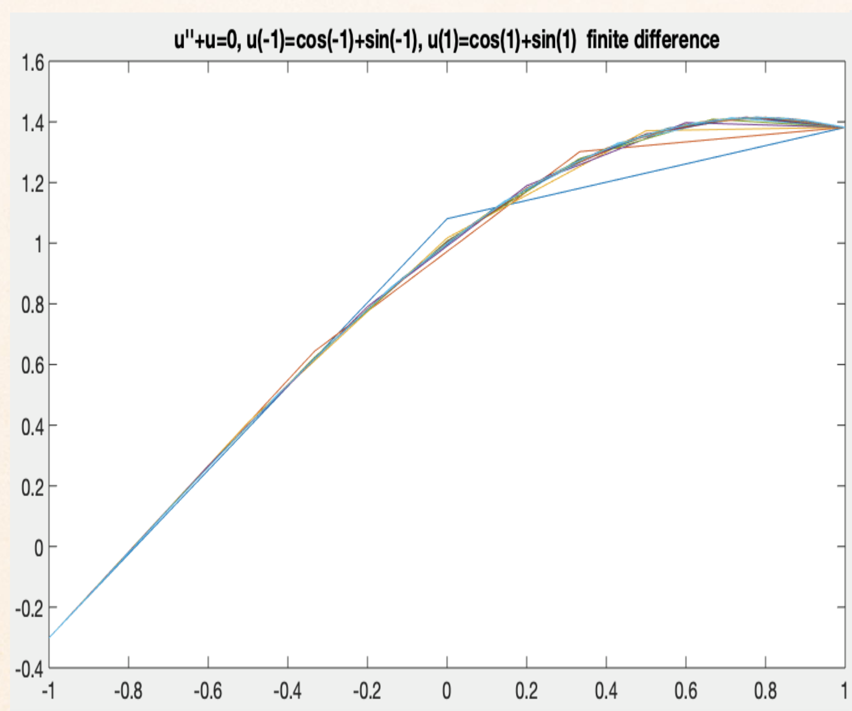
Graph of Solution

$$u'' + u = 0, \ -1 \leq x \leq 1, \ u(-1) = \alpha, \ \ u(1) = \beta$$

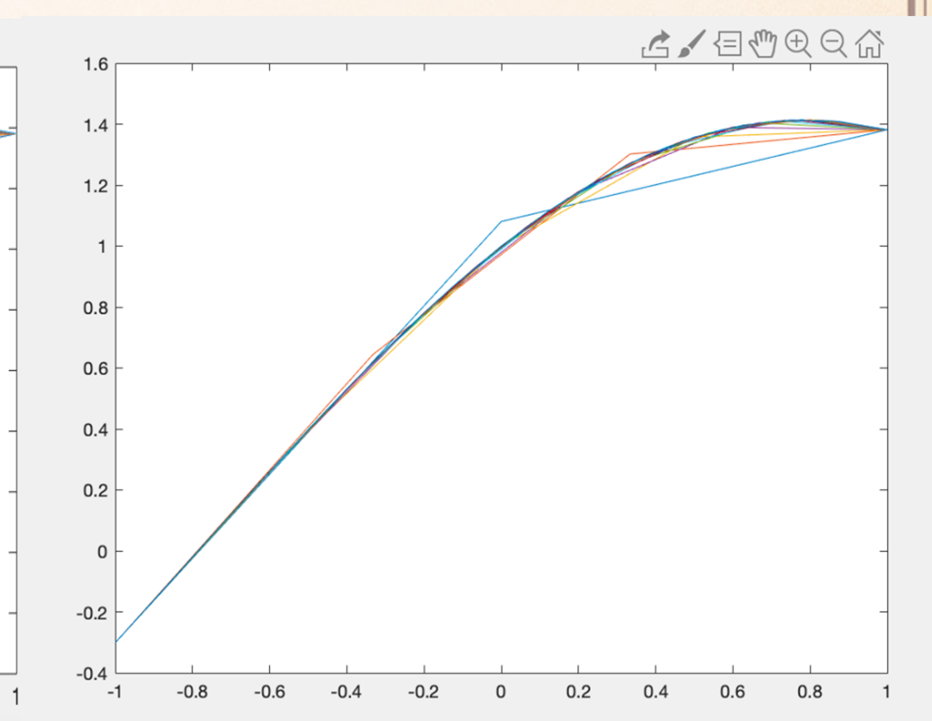
$$\alpha = cos(-1) + sin(-1), \ \ \beta = cos(1) + sin(1)$$



Chebfun



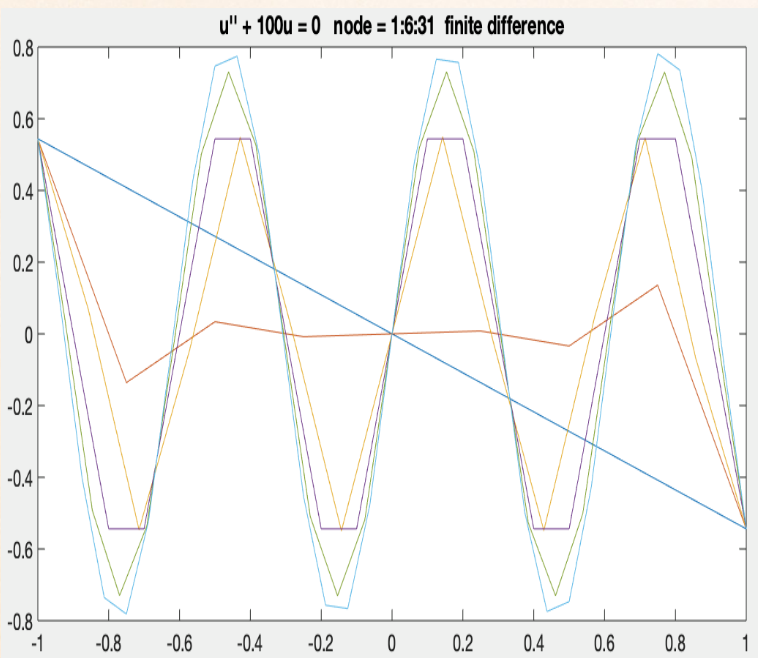
Finite Difference



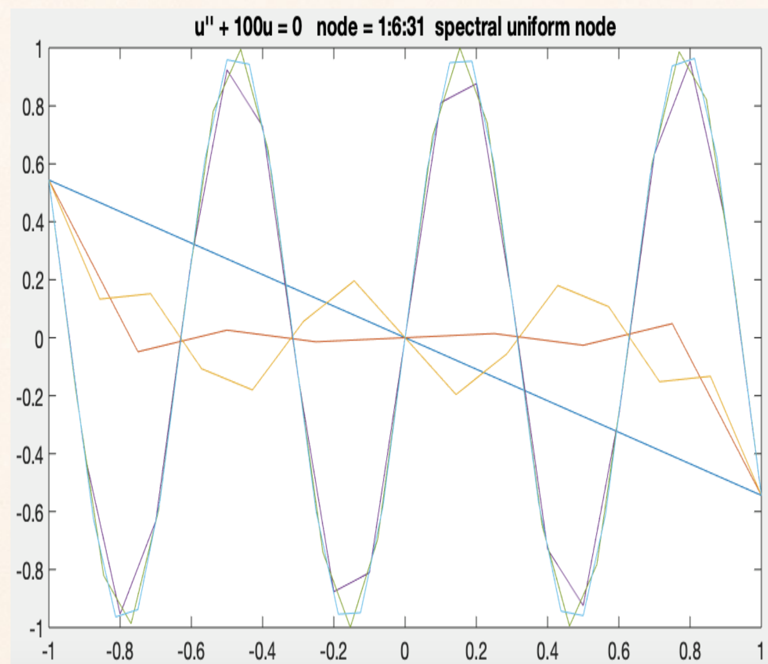
Spectral Method
with uniform nodes

Graph of Solution

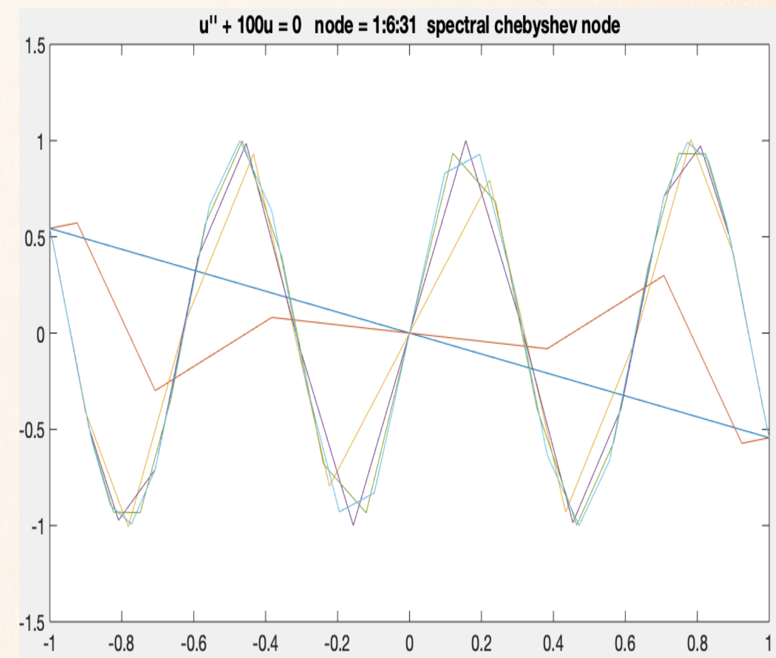
$$u'' + 100u = 0, \ -1 \le x \le 1, \ u(-1) = sin(-10), \ \ u(1) = sin(10)$$



Finite Difference



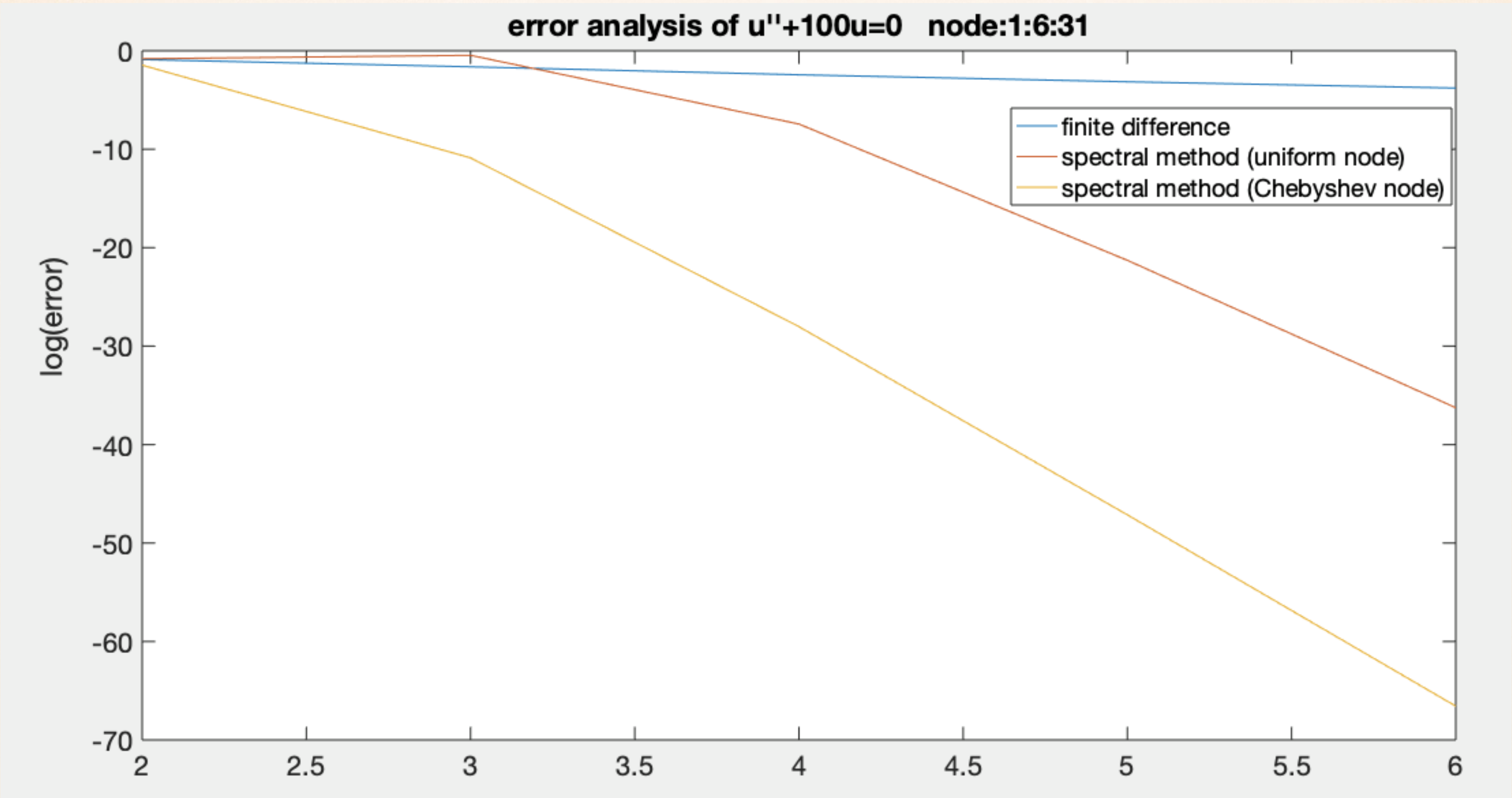
Spectral Method with uniform nodes



Spectral Method with Chebyshev nodes

- Spectral method with uniform nodes cannot be used with large numbers of nodes (should be less than 40).

Error Analysis



General Solution

$$u'' - u = 0, \; -1 \le x \le 1, \; u(-1) = \alpha, \;\; u(1) = \beta$$

$$u(x) = C_1 e^{x} + C_2 e^{-x}$$

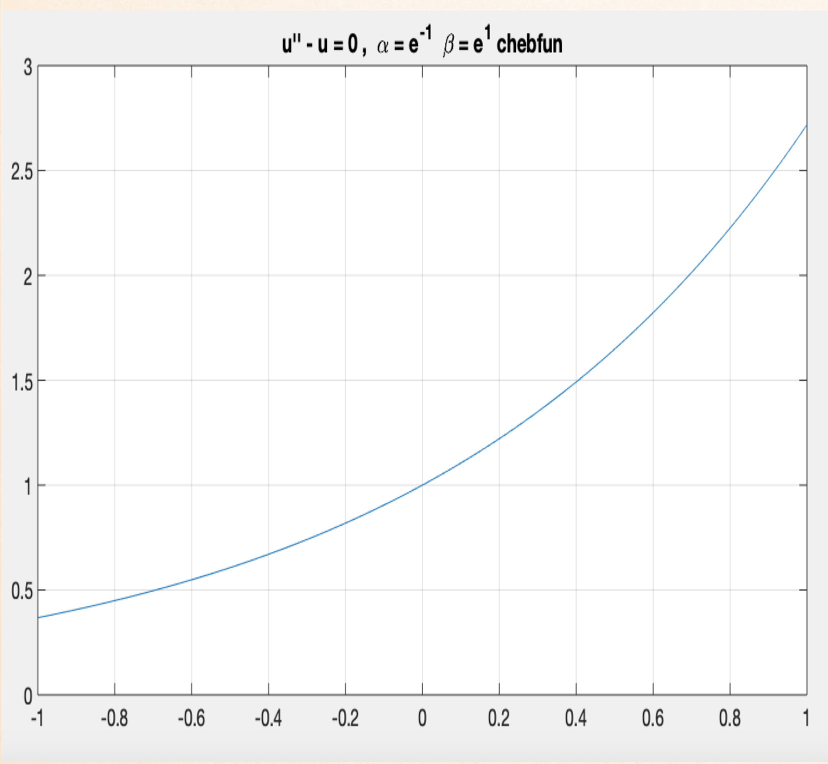
$$C_1 e^{-1} + C_2 e = \alpha$$

$$C_1 e + C_2 e^{-1} = \beta$$

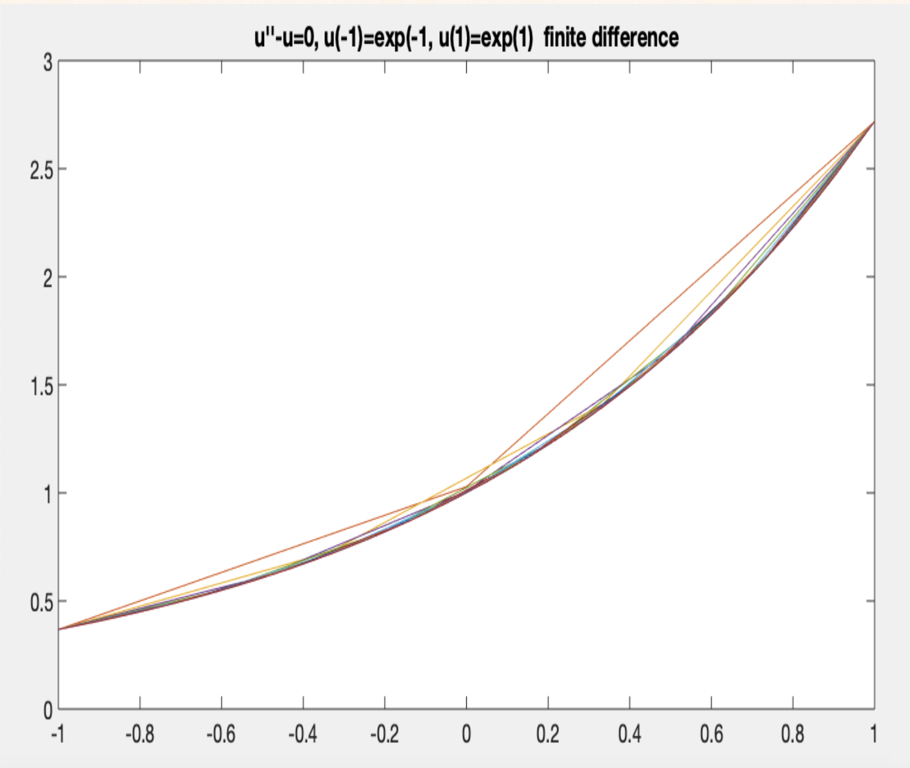
Graph of Solution

$$u'' - u = 0, \; -1 \le x \le 1, \; u(-1) = \alpha, \;\; u(1) = \beta$$

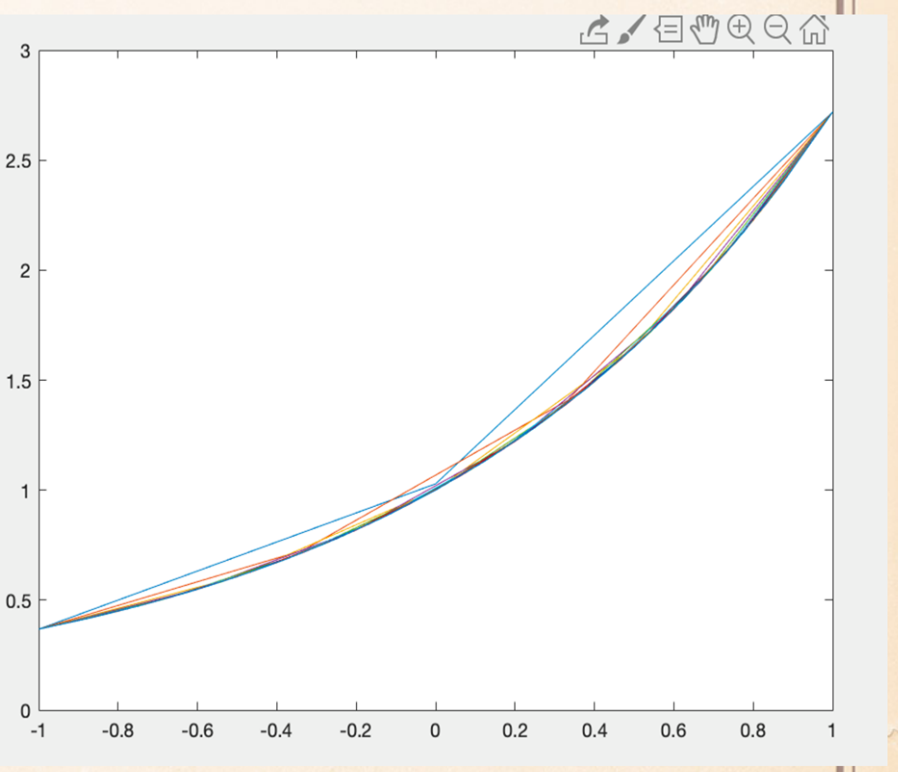
$$\alpha = e^{-1}, \;\; \beta = e$$



Chebfun



Finite Difference



Spectral Method
with uniform nodes

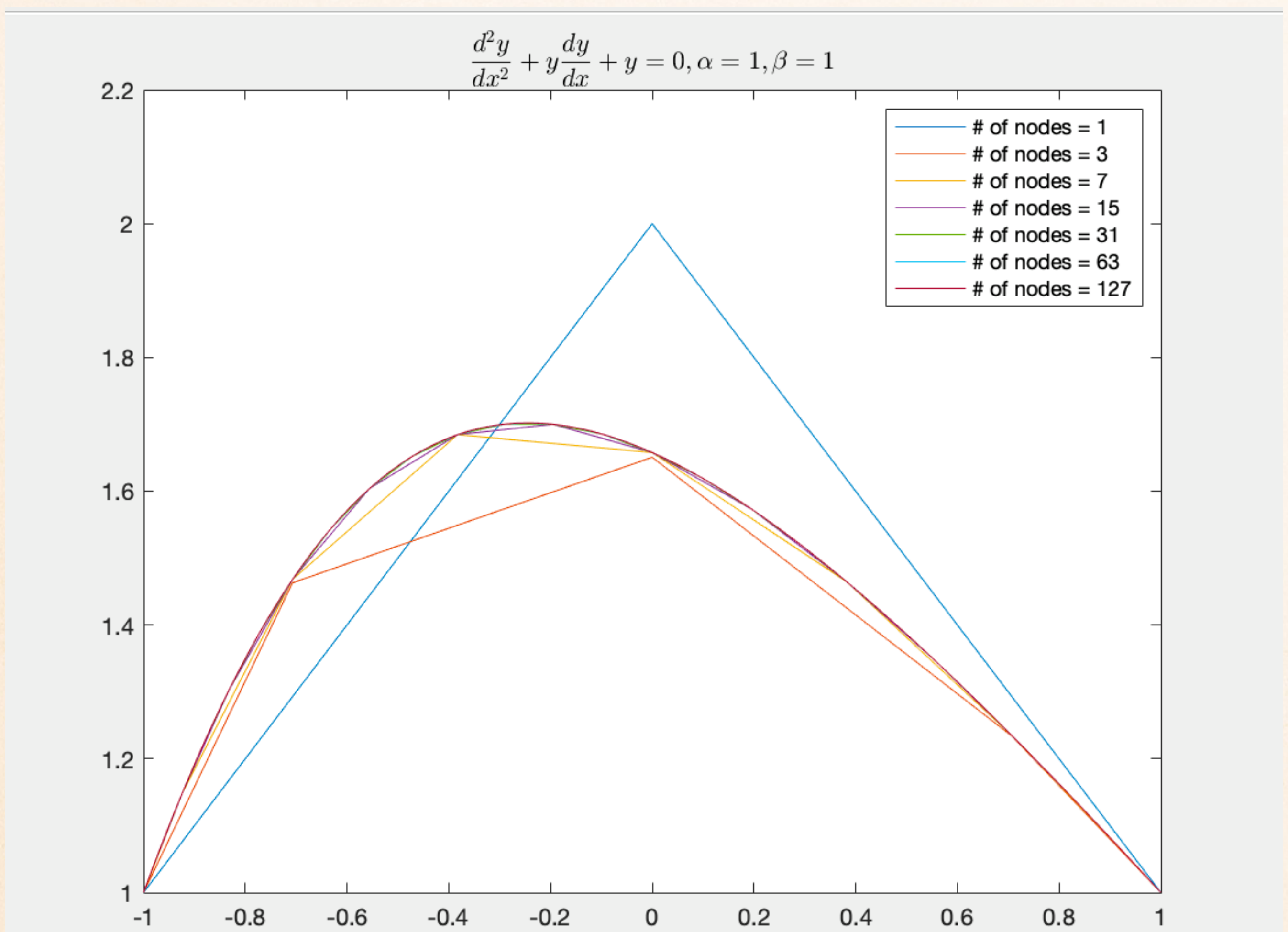
Burger's Equation with $b \neq 0$

Graph of Solution

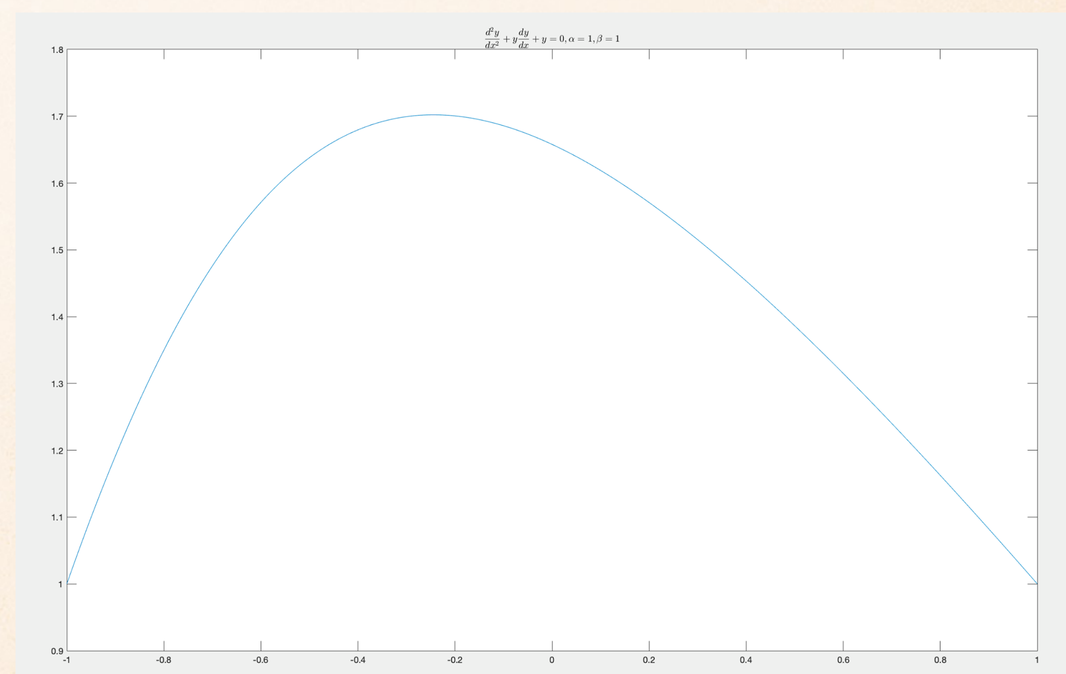
$$u'' + uu' + u = 0$$

- Spectral method with Chebyshev nodes

$\alpha = 1, \ \beta = 1$



Chebfun



Equations With Interior Layers

$$\epsilon u'' + uu' - u = 0, \epsilon \ll 1$$

General Solution of a Reduced Eq.

$$uu' - u = 0$$

$$u(u' - 1) = 0$$

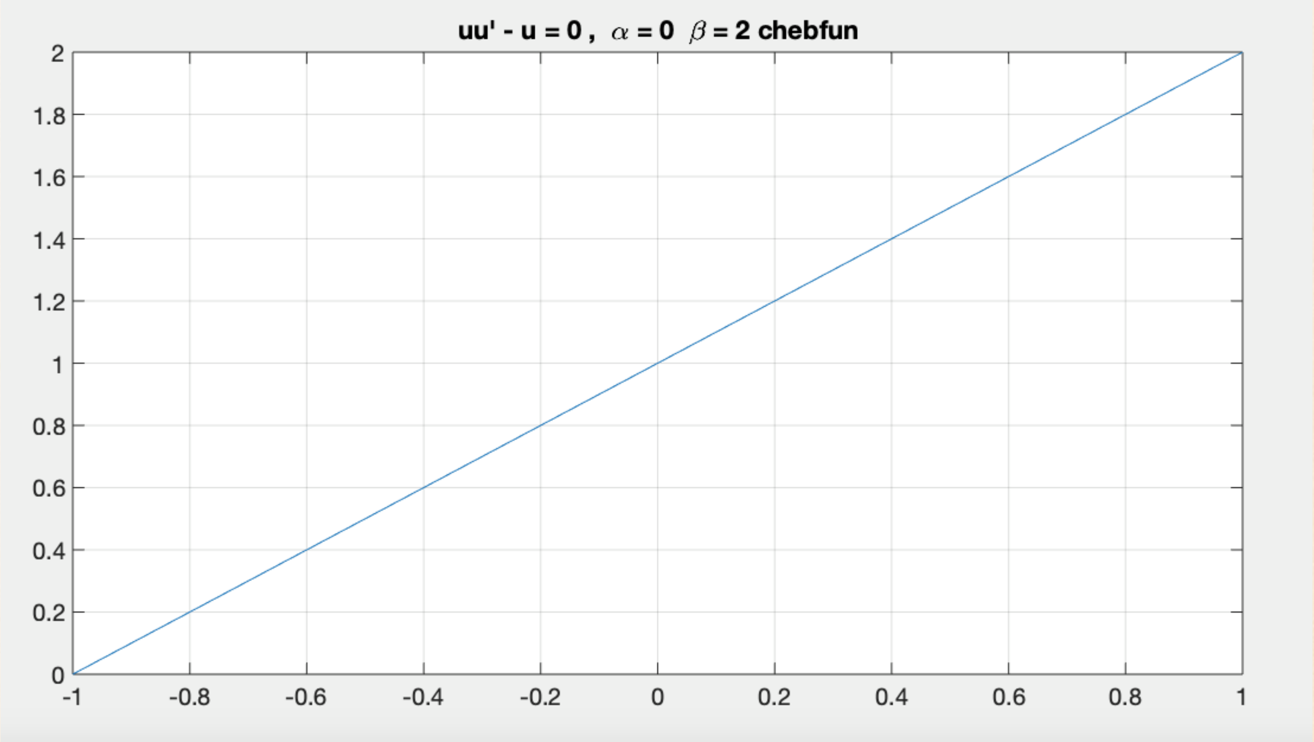
$$\Rightarrow u(x) = 0 \ or \ x + C$$

if $\quad u(-1) = \alpha$

$$\Rightarrow u(x) = x + \alpha + 1$$

if $\quad u(1) = \beta$

$$\Rightarrow u(x) = x + \beta - 1$$



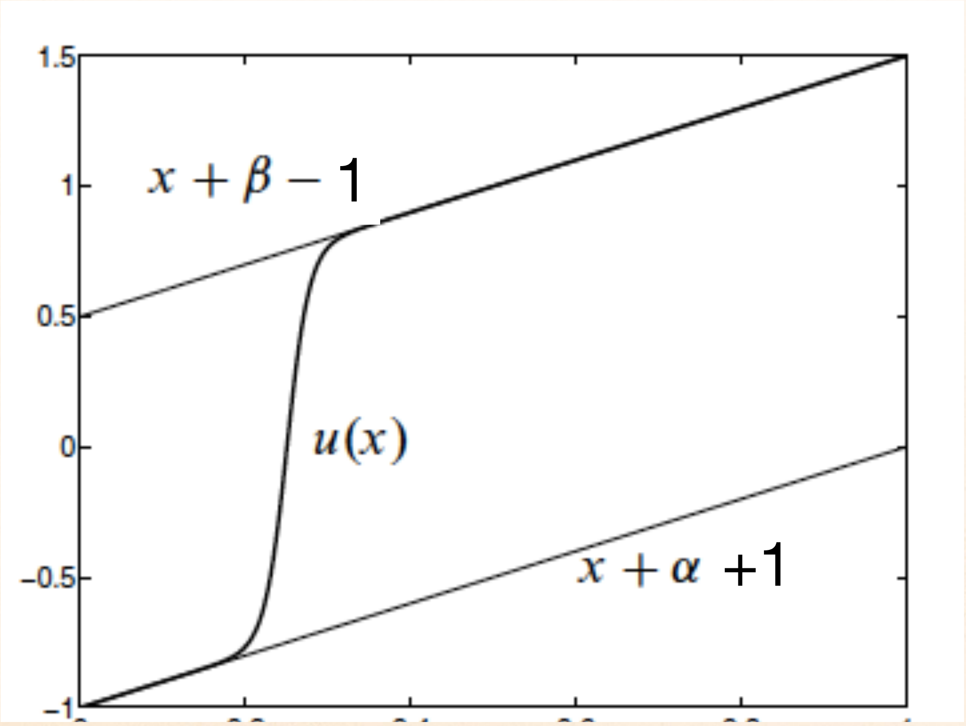
Solution Analysis

$$\epsilon u'' + uu' - u = 0, \epsilon \ll 1$$

When u'' is small, $\epsilon u''$ is negligible.

So the solution should be close to

that of the reduced equation $uu' - u = 0$



- u(x) can be approximated:

$$\omega_0 = \frac{1}{2}(-2 + \beta - \alpha)$$

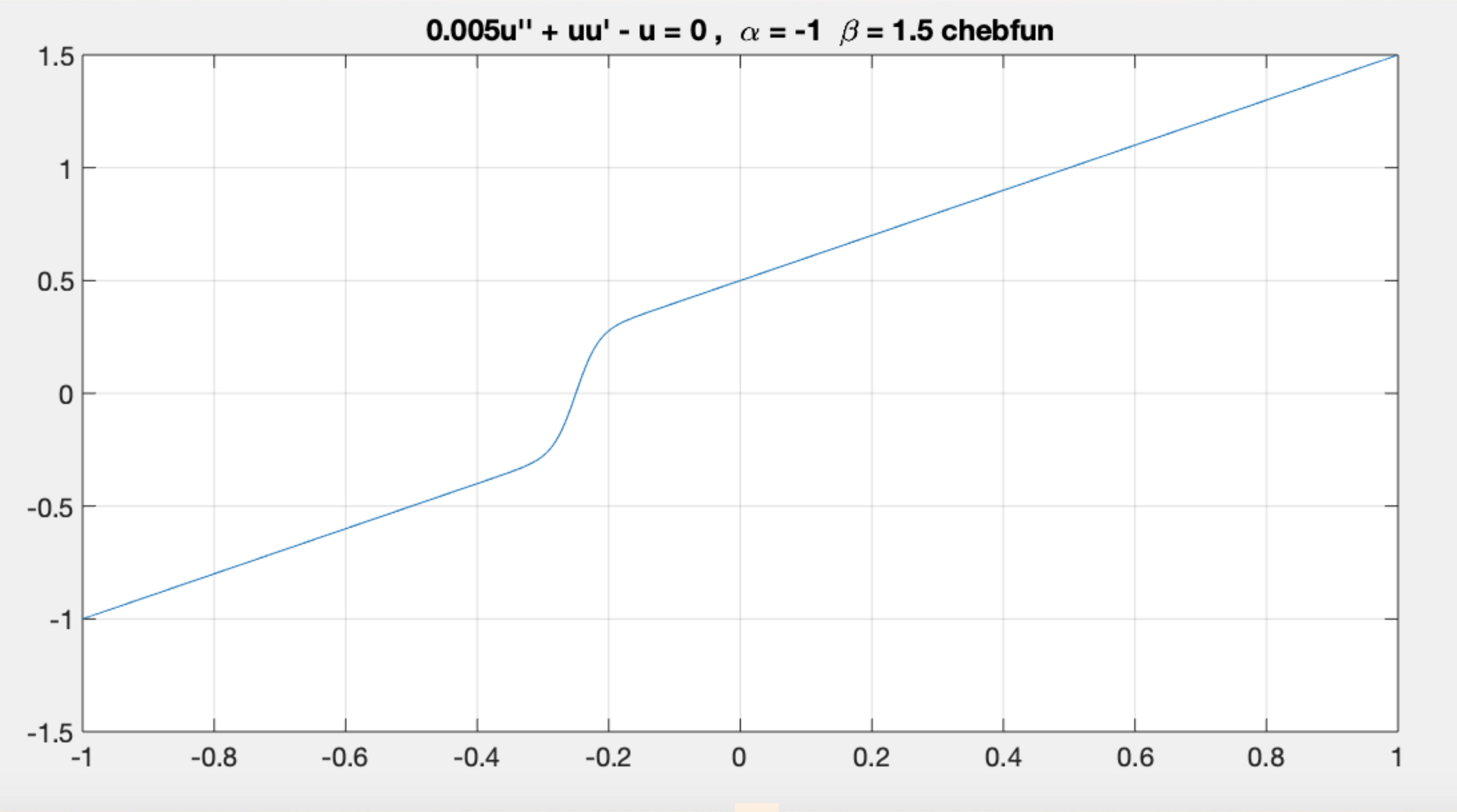
$$\bar{x} = -\frac{1}{2}(\alpha + \beta)$$

$$u(x) \approx x - \bar{x} + \omega_0 tanh(\omega_0 (x - \bar{x})/2\epsilon)$$

Graph of Solution

$$\epsilon u'' + uu' - u = 0, \epsilon \ll 1$$

- Chebfun $\epsilon = 0.005,\ \alpha = -1,\ \beta = 1.5$



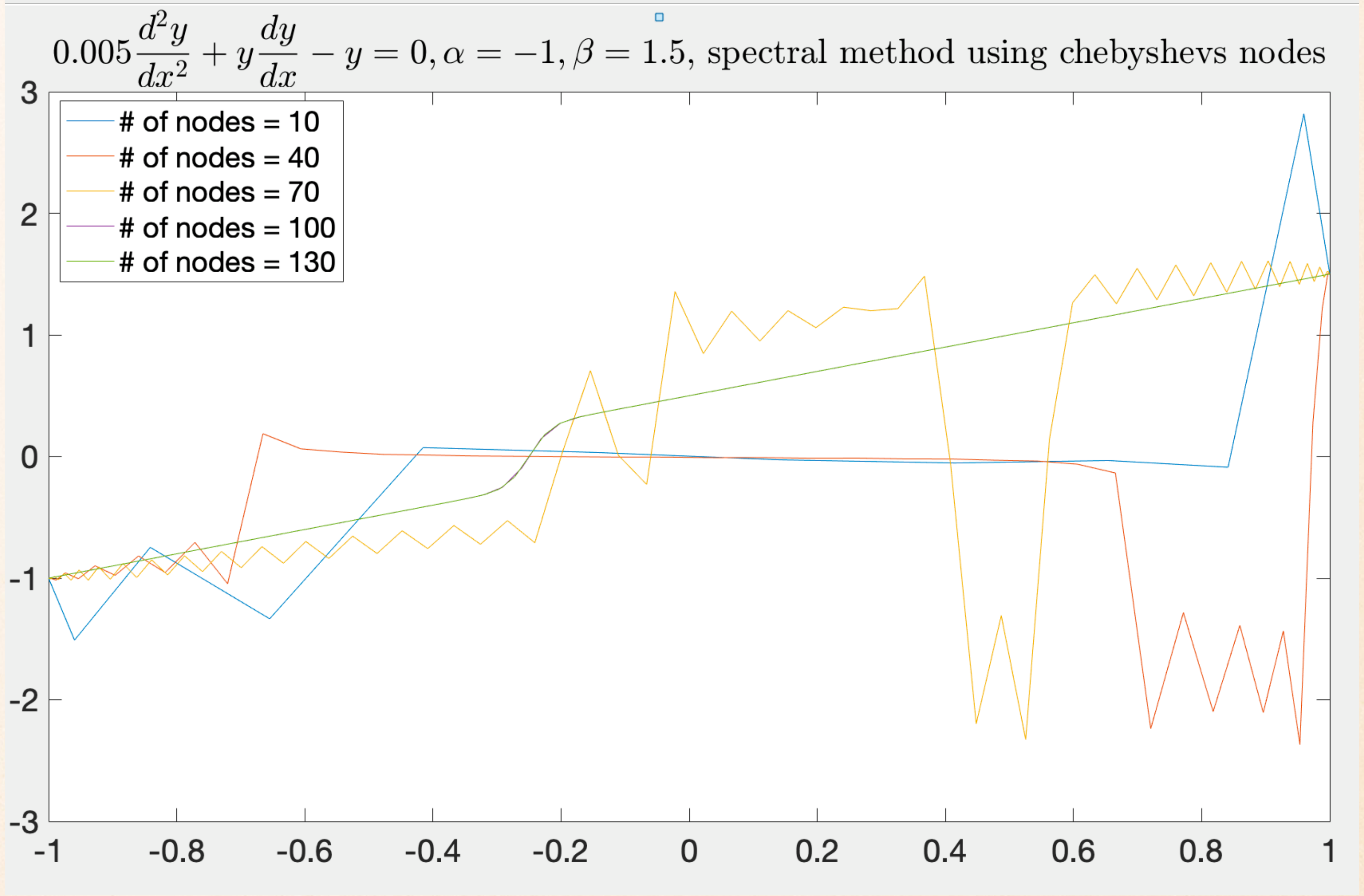
Graph of Solution

$$\epsilon u'' + uu' - u = 0, \epsilon \ll 1$$

- Spectral method with Chebyshev nodes

$\epsilon = 0.005$

$\alpha = -1, \ \beta = 1.5$

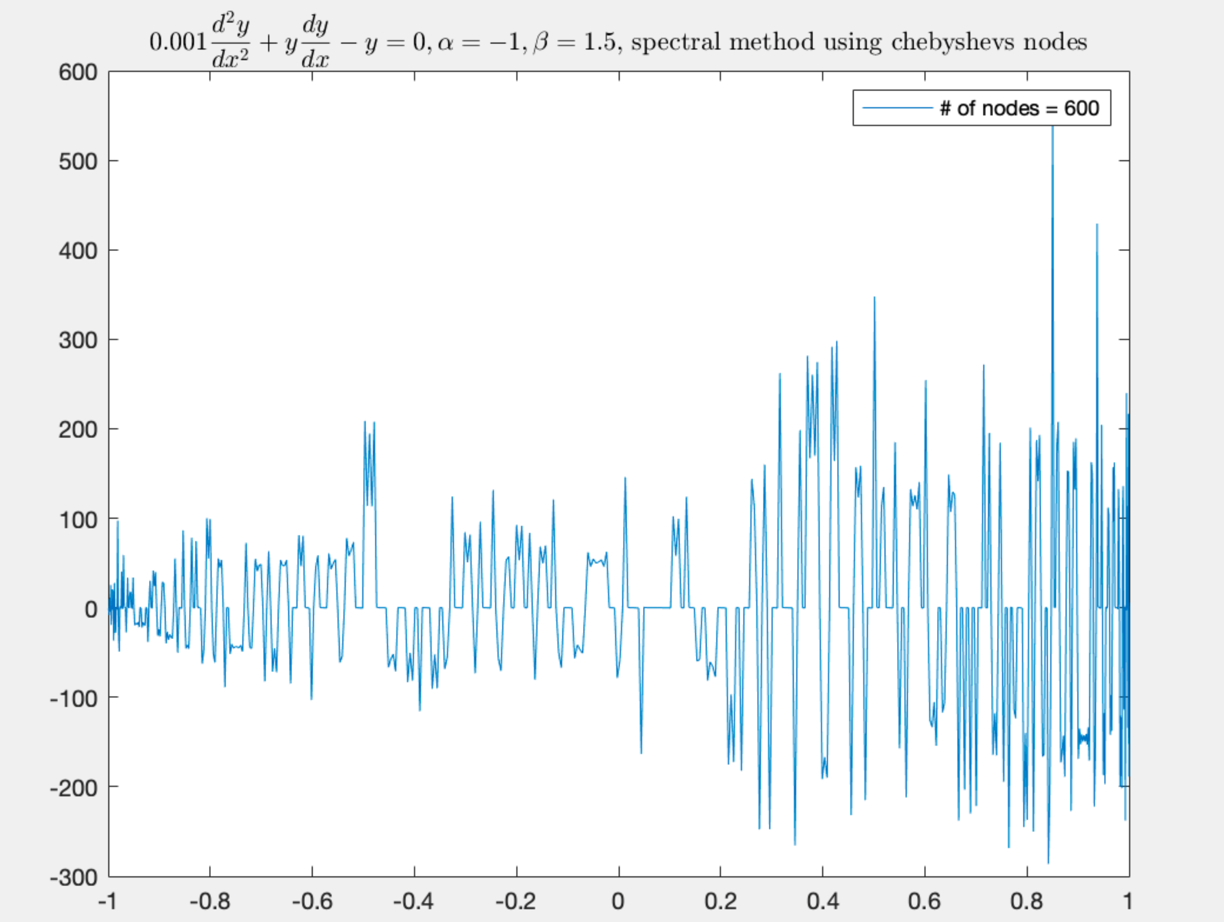


Graph of Solution

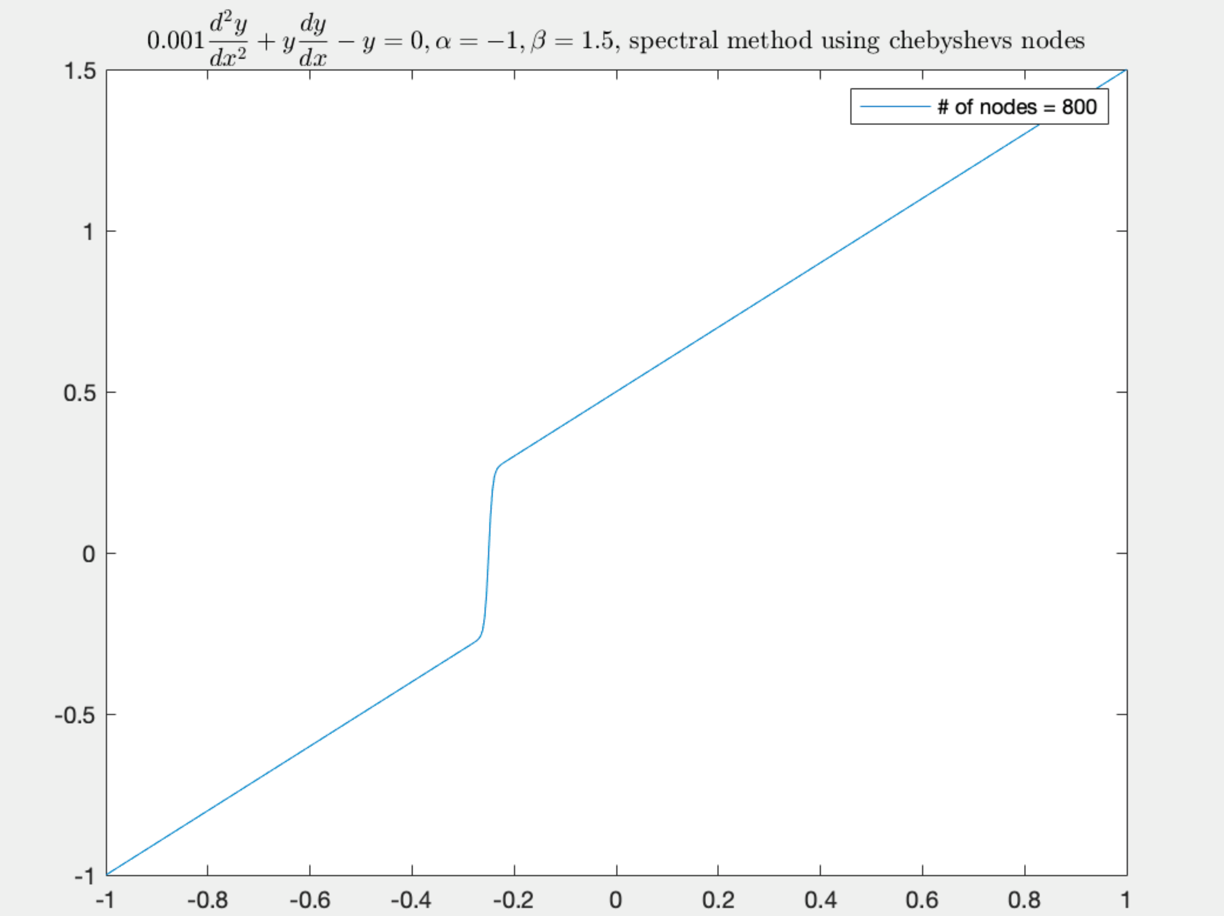
$$\epsilon u'' + uu' - u = 0, \epsilon \ll 1$$

- Spectral method with Chebyshev nodes

$\epsilon = 0.001, \ \alpha = -1, \ \beta = 1.5$



600 nodes



800 nodes

Better Than Chebfun

$$\epsilon u'' + uu' - u = 0, \epsilon \ll 1$$

- Chebfun $\epsilon = 0.001,\ \alpha = -1,\ \beta = 1.5$ FAIL!

Editor – /Users/chenyuming/Documents/MATLAB/cheb.m

+6 | cheb.m | main_without_layer_uniform.m | main_without_layer_chebyshev.m | +

```
1 -   L = chebop(-1,1);
2 -   L.op = @(x,y)  0.001*diff(y,2) + 1*y*diff(y) - y;
3 -   L.lbc = -1; L.rbc = 1.5;
4 -   y = L\0;
5 -   plot(y);
6
```

Command Window

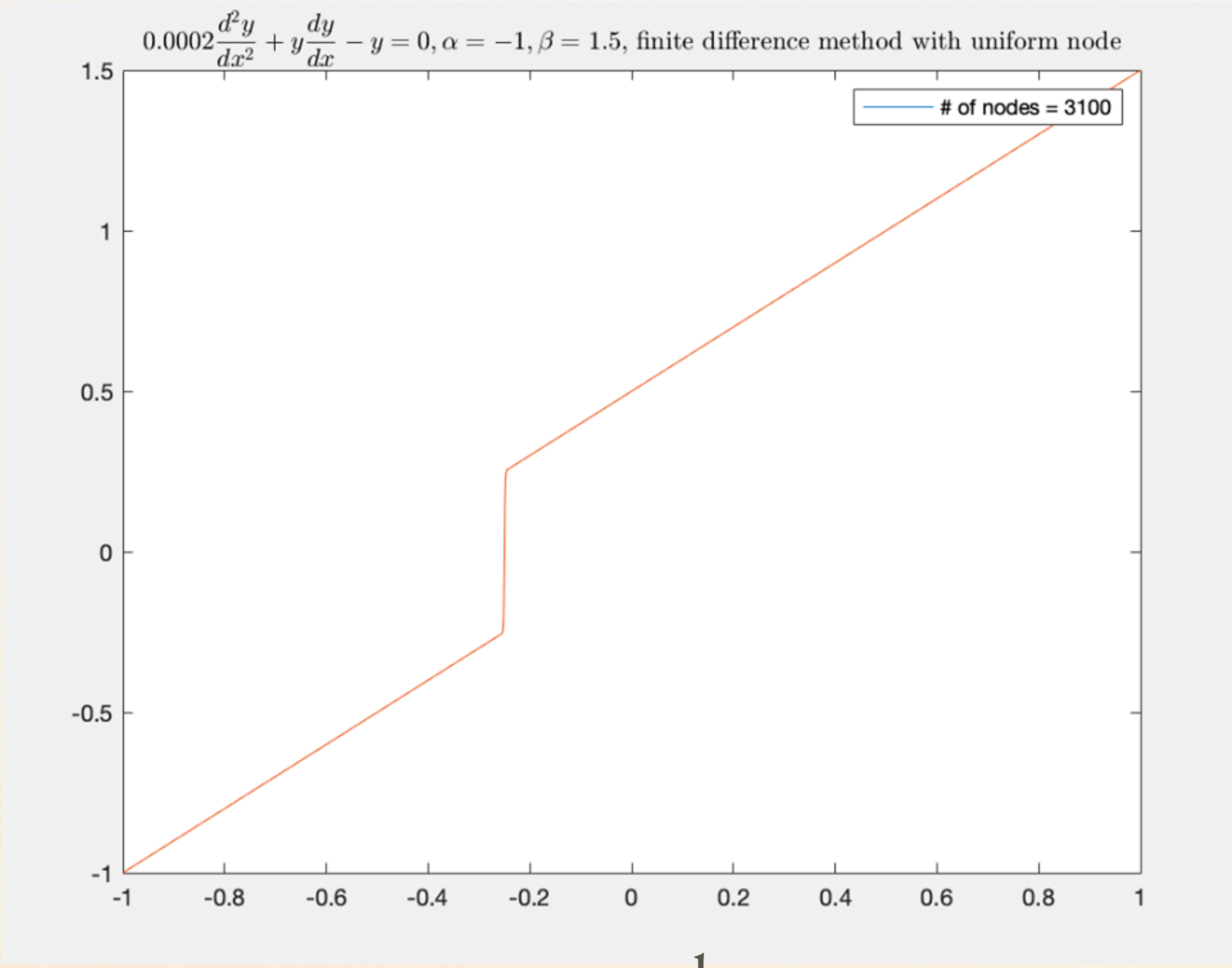
```
>> cheb
Warning: Linear system solution may not have converged.
> In linop/linsolve (line 165)
  In chebop/dampingErrorBased (line 129)
  In chebop/solvebvpNonlinear (line 129)
  In chebop/solvebvp (line 241)
  In \ (line 56)
  In cheb (line 4)
Undefined function or variable 'cFactor'.

Error in chebop/dampingErrorBased (line 201)
dampingInfo.cFactor =       cFactor;

Error in chebop/solvebvpNonlinear (line 129)
        [u, dampingInfo] = dampingErrorBased(N, u, rhs, delta, ...

Error in chebop/solvebvp (line 241)
    [u, info] = solvebvpNonlinear(N, rhs, L, u0, residual, pref, displayInfo);

Error in \ (line 56)
    [varargout{1:nargout}] = solvebvp(N, rhs, varargin{:});

Error in cheb (line 4)
y = L\0;

fx >>
```

Graph of Solution

$\epsilon u'' + uu' - u = 0, \epsilon \ll 1$

- Finite difference, the number of nodes can be large.

$\epsilon = 0.0002, \ \ \alpha = -1, \ \ \beta = 1.5$

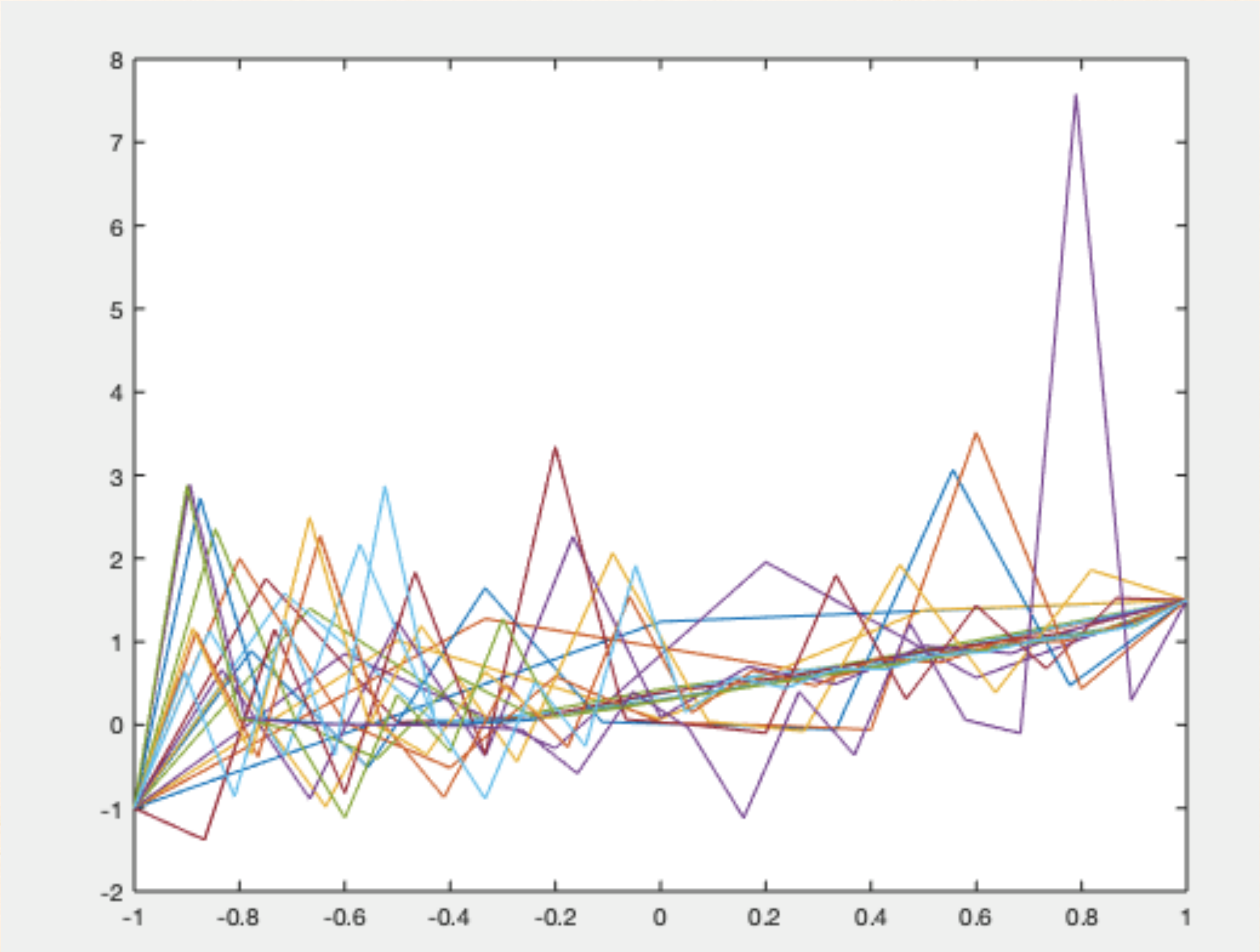


3100 nodes

Discussion

$$\epsilon u'' + uu' - u = 0, \epsilon \ll 1$$

- Finite difference doesn't work if the initial value is randomly chosen from [0, 1].

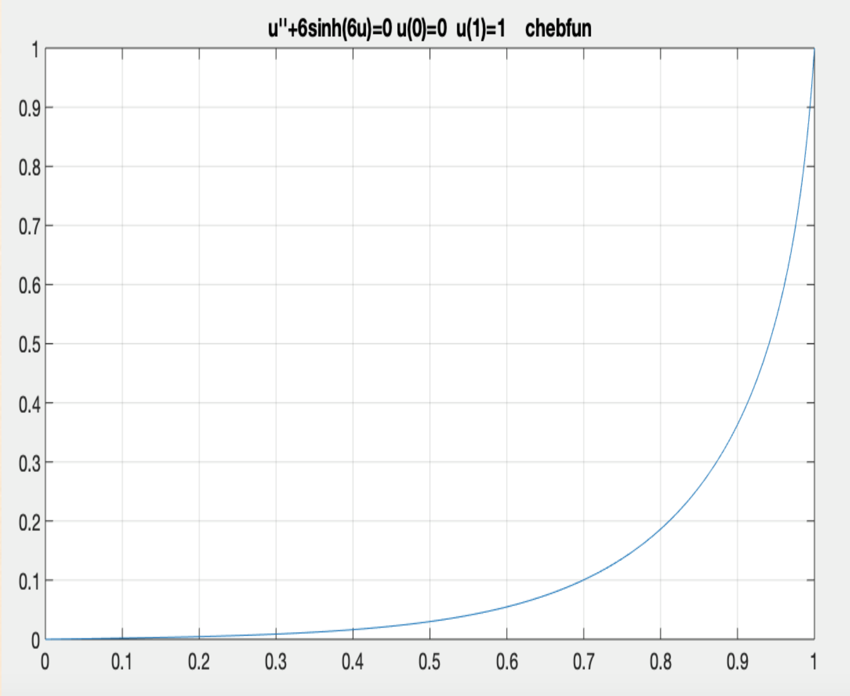


Troesch Equation

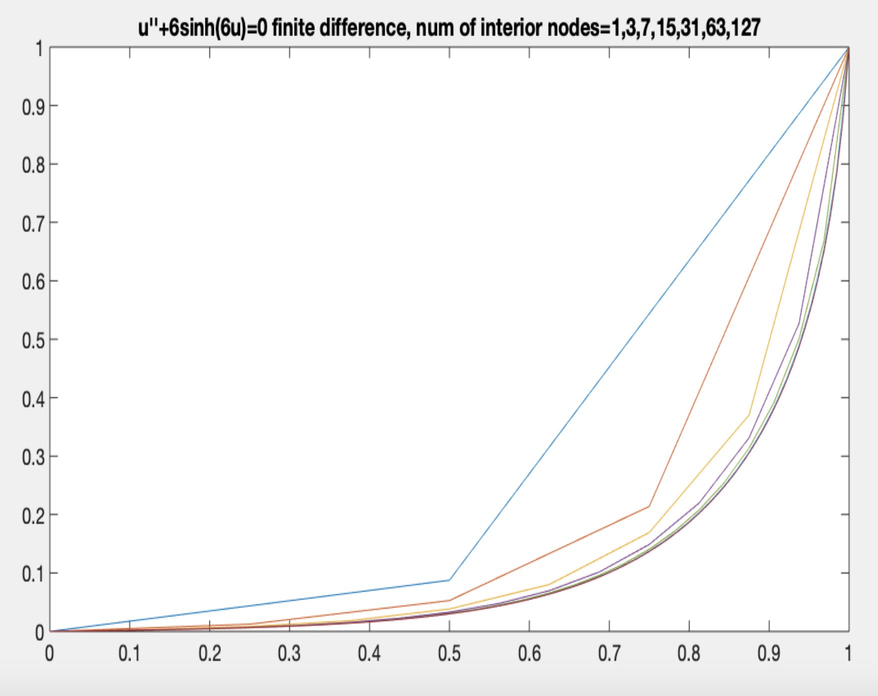
Graph of Solution

$$u'' - 6sinh(6u) = 0,\ \ 0 \le x \le 1$$

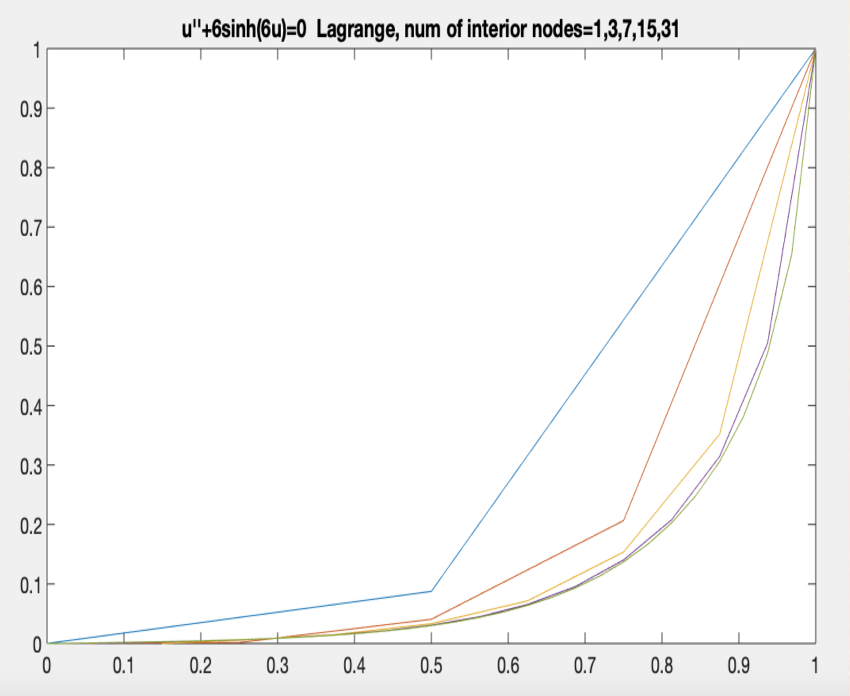
$$u(0) = 0,\ u(1) = 1$$



Chebfun



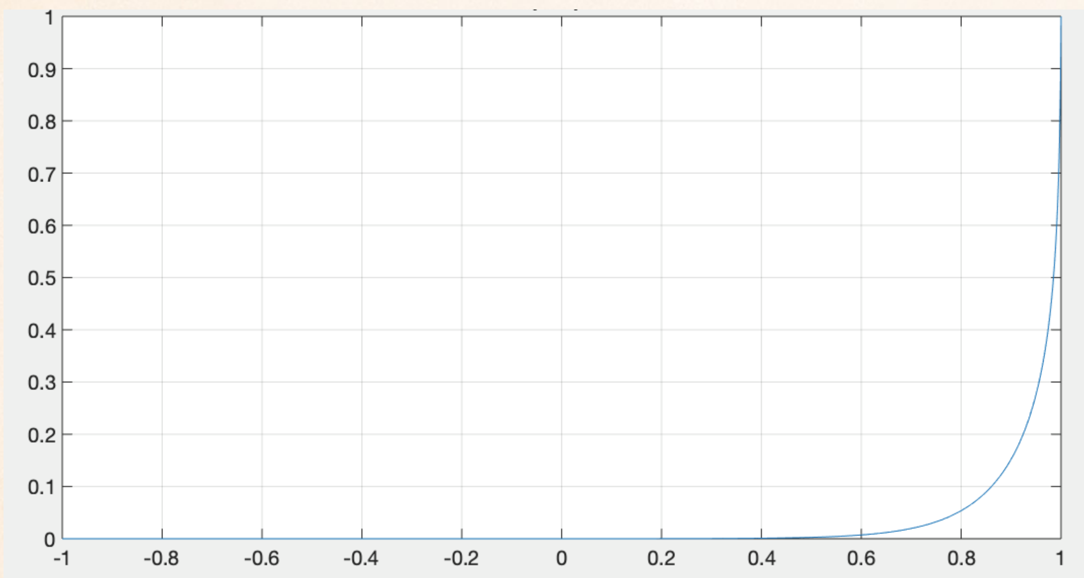
Finite Difference



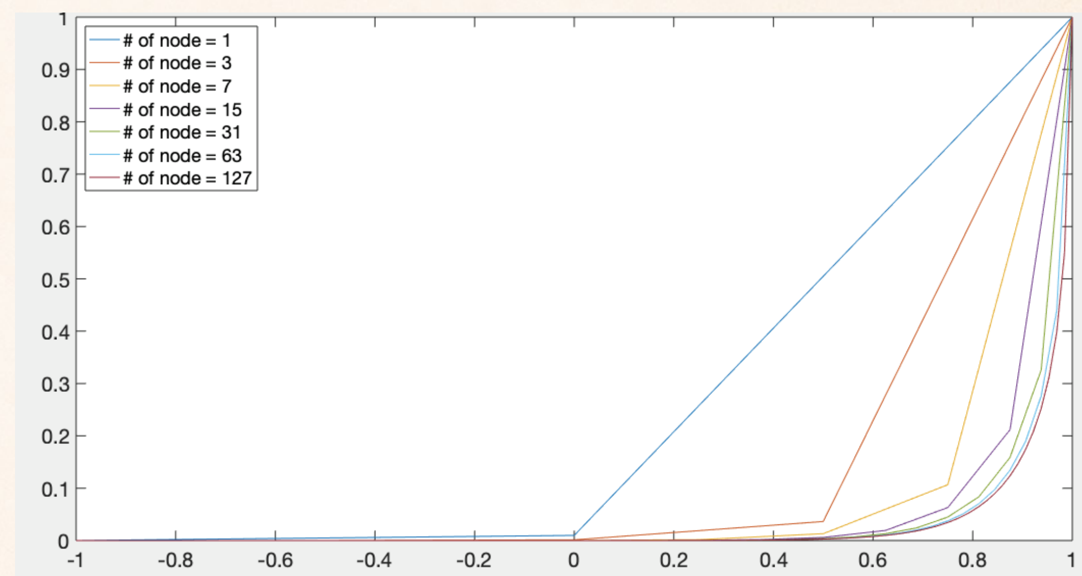
Spectral Method

Graph of Solution

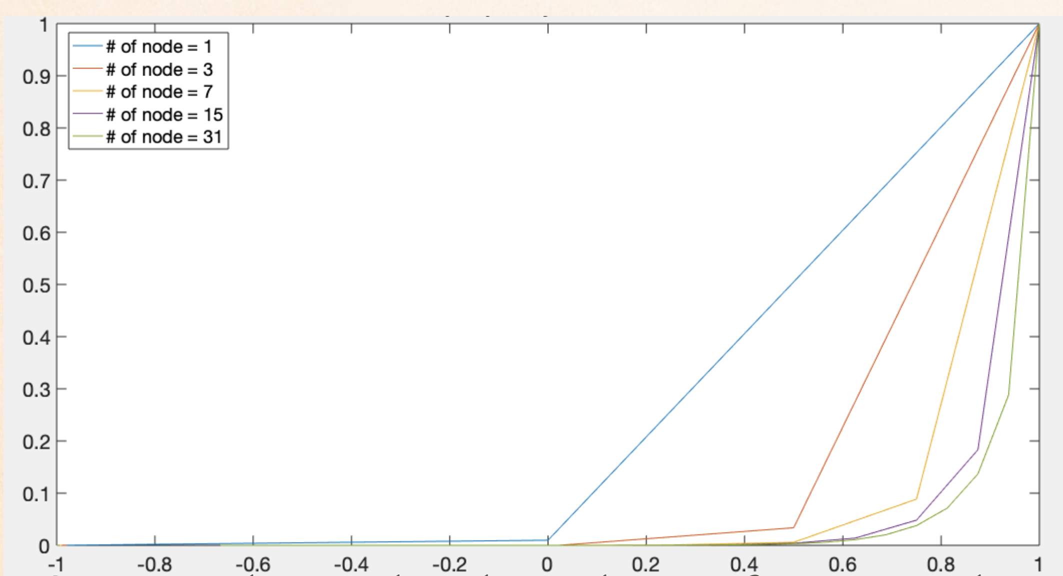
$$u'' - 10sinh(10u) = 0, \;\; 0 \le x \le 1 \qquad u(0) = 0, \; u(1) = 1$$



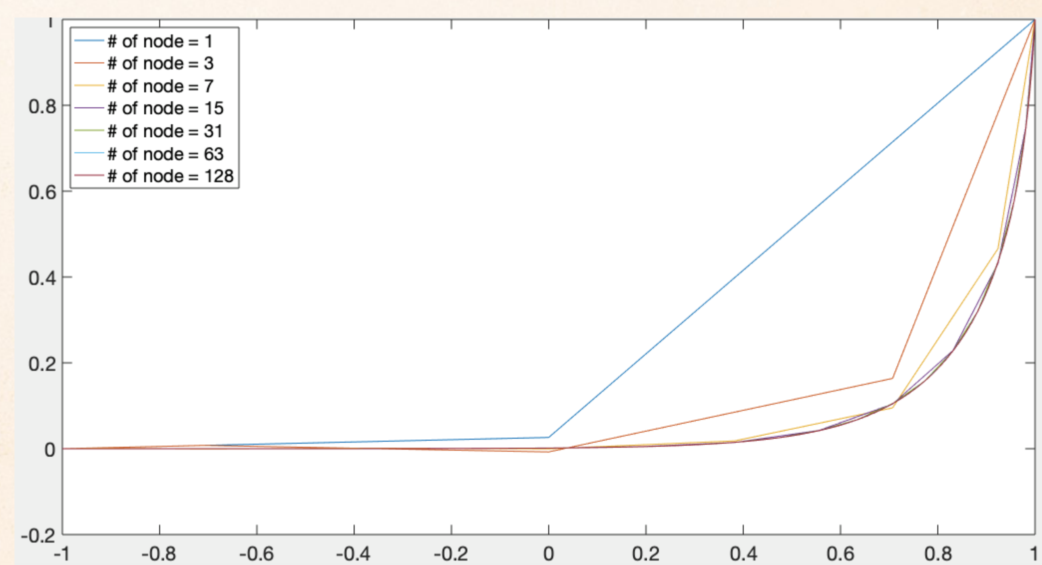
Chebfun



Finite Difference



Spectral Method with uniform nodes



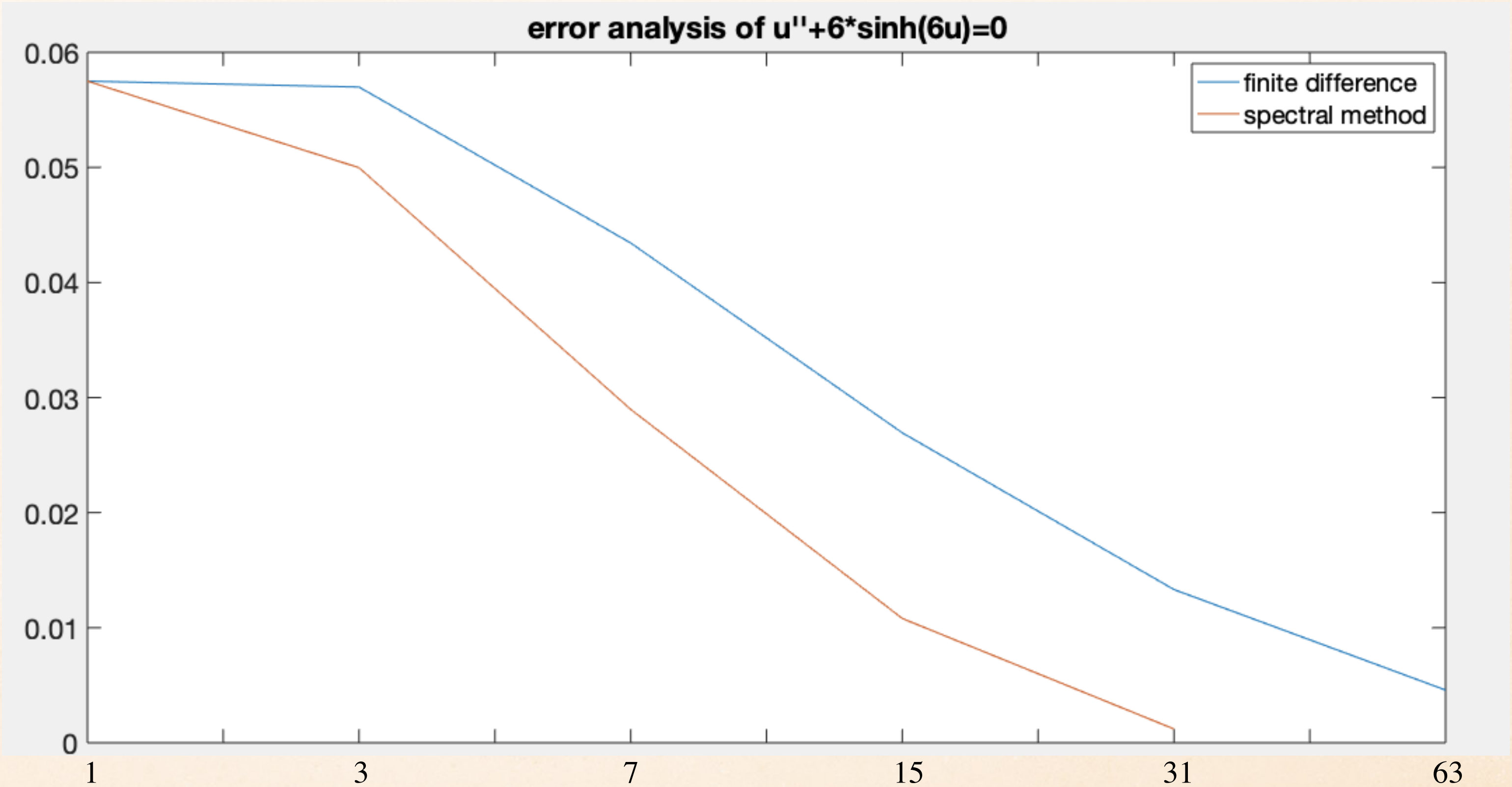
Spectral Method with Chebyshev nodes

Error Analysis

$$u'' - 6sinh(6u) = 0, \;\; 0 \leq x \leq 1$$

$$u(0) = 0, \; u(1) = 1$$

take finite difference with 128 intervals as an approximation of real solution



Discussion

- Spectral method with uniform nodes will cause ill-conditioned matrices when the number of nodes (n) is too large. But with Chebyshev nodes, n can be much larger. For finite difference, the condition of differential matrix is even better (tri-diagonal).

```
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND =  3.375589e-26.
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND =  3.379787e-26.
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND =  3.380329e-26.
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND =  3.378893e-26.
[Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND =  3.375284e-26.>>
```

- The condition of differential matrix:

  finite difference > Chebyshev nodes > uniform nodes

- Spectral method converges faster than finite difference. In particular, the convergence rate:

  Chebyshev nodes > uniform nodes > finite difference

Future Work

- Improve the condition of Jacobian matrix, such as using SVD to rule out small eigenvalues.
- Use Chebyshev polynomial to construct differential matrices.

Q&A