Polynomialization/Quadratic-Linearization of Tubular Reactor

Benjamin Peherstorfer, James Buenfil, Melissa Schmidt-Landin

July 16, 2018

Setup

The non-discretized system of differential equations is:

$$\frac{\partial y}{\partial \tau} = \frac{1}{Pe}\frac{\partial^2 y}{\partial s^2} - \frac{\partial y}{\partial s} - Df(y,\theta),$$
$$\frac{\partial \theta}{\partial \tau} = \frac{1}{Pe}\frac{\partial^2 \theta}{\partial s^2} - \frac{\partial \theta}{\partial s} - \beta(\theta - \theta_0) + \frac{BD}{\mu} f(y,\theta),$$

where $s \in (0,1)$, $\tau \geq 0$. $f(y,\theta) = ye^{\gamma(1-\frac{1}{\theta})}$. s and τ are the non dimensional length and time respectively. y and θ represent the non-dimensional concentration and temperature respectively. Pe is the Péclet number. γ, B, β, and θ_0 are known constants of the system. D is the Damköhler number which controls the dynamics of the system. Robin boundary condition is imposed at the left boundary ($s = 0$) while the Neumann condition is given at the right boundary ($s = 1$):

$$\left.\frac{\partial y}{\partial s}\right|_{s=0} = Pe(y|_{s=0} - \mu)$$
$$\left.\frac{\partial \theta}{\partial s}\right|_{s=0} = Pe(\theta|_{s=0} - 1)$$
$$\left.\frac{\partial y}{\partial s}\right|_{s=1} = 0$$
$$\left.\frac{\partial \theta}{\partial s}\right|_{s=1} = 0$$

We let $Pe = 5$, $\gamma = 25$, $B = .5$, $\beta = 2.5$, $\mu = 1$, and $\theta_0 = 1$.

The system is discretized as in [1] in the spatial domain with resolution $\Delta s = \frac{1}{N+1}$, where N is the number of interior grid points. We define for the original system a state vector containing the concentration and temperature evaluated at the interior grid points:

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{\theta} \end{bmatrix} \in \mathbb{R}^{2N\times 1}, \text{ with } \boldsymbol{y} = \begin{bmatrix} y_1(\tau) \\ \vdots \\ y_N(\tau) \end{bmatrix} \in \mathbb{R}^{N\times 1}, \text{ and } \boldsymbol{\theta} = \begin{bmatrix} \theta_1(\tau) \\ \vdots \\ \theta_N(\tau) \end{bmatrix}.$$

$y_i(\tau) = y(s_i, \tau)$, and $\theta_i(\tau) = \theta(s_i, \tau)$, with $s_i = i\Delta s$. A second-order centered difference method is applied in the interior of the domain. Second-order forward and backward difference schemes are used for the inflow and outflow boundary conditions respectively. The semi-discretized system is found to be

$$\dot{\boldsymbol{u}} = \boldsymbol{A}\boldsymbol{u} + \boldsymbol{b} + \boldsymbol{F}(\boldsymbol{u}; \mathrm{D}) \tag{1}$$

where

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A^D} - \boldsymbol{A^C} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A_1} \end{bmatrix} \in \mathbb{R}^{2N\times 2N}, \; \boldsymbol{A_1} = \boldsymbol{A^D} - \boldsymbol{A^C} - \beta\boldsymbol{I} \tag{2}$$

$$\boldsymbol{A^D} = \frac{1}{Pe(\Delta s)^2} \begin{bmatrix} A^D_{1,1} & A^D_{1,2} & & & & \\ 1 & -2 & 1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & 1 & -2 & 1 & \\ & & & A^D_{N,N-1} & A^D_{N,N} & \end{bmatrix} \in \mathbb{R}^{N\times N} \tag{3}$$

$$\boldsymbol{A^C} = \frac{1}{2\Delta s} \begin{bmatrix} A^C_{1,1} & A^C_{1,2} & & & & \\ -1 & 0 & 1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & -1 & 0 & 1 & \\ & & & A^C_{N,N-1} & A^C_{N,N} & \end{bmatrix} \in \mathbb{R}^{N\times N} \tag{4}$$

$$A^D_{1,1} = \frac{4}{3 + 2\Delta s Pe} - 2 \qquad A^C_{1,1} = \frac{-4}{3 + 2\Delta s Pe}$$

$$A^D_{1,2} = -\frac{1}{3 + 2\Delta s Pe} + 1 \qquad A^C_{1,2} = \frac{1}{3 + 2\Delta s Pe} + 1$$

$$A^D_{N,N-1} = \frac{2}{3} \qquad A^C_{N,N-1} = \frac{4}{3}$$

$$A^D_{N,N} = -\frac{2}{3} \qquad A^C_{N,N} = -\frac{4}{3}$$

Note that $\boldsymbol{A}$ and $\boldsymbol{b}$ are defined differently in [1], and unless the corrections are made, their plots aren't reproducible.

$$\boldsymbol{b} = \begin{bmatrix} \boldsymbol{b_y} \\ \boldsymbol{b_\theta} \end{bmatrix} \in \mathbb{R}^{2N\times 1}, \text{ where } \boldsymbol{b}_y = \begin{bmatrix} b_0\mu \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{N\times 1}, \boldsymbol{b}_\theta = \begin{bmatrix} b_0 + \beta\theta_0 \\ \beta\theta_0 \\ \vdots \\ \beta\theta_0 \end{bmatrix} \in \mathbb{R}^{N\times 1}$$

and

$$b_0 = \frac{2 + Pe\Delta s}{\Delta s(3 + 2\Delta s Pe)}.$$

$$\boldsymbol{F(u)} = \begin{bmatrix} -D\boldsymbol{f(u)} \\ \frac{BD}{\mu}\boldsymbol{f(u)} \end{bmatrix} \in \mathbb{R}^{2N\times 1}$$

with

$$\boldsymbol{f(u)} = \begin{bmatrix} f(y_1, \theta_1) \\ \vdots \\ f(y_N, \theta_N) \end{bmatrix} \in \mathbb{R}^{N\times 1} \tag{5}$$

and

$$f(y_i, \theta_i) = y_i e^{\gamma - \frac{\gamma}{\theta_i}}.$$

$A^D_{1,1}$, $A^D_{1,2}$, $A^D_{N,N}$, $A^D_{N,N}$, $A^C_{1,1}$, $A^C_{1,2}$, $A^C_{N,N-1}$, $A^C_{N,N}$, and b_0 arise due to one-sided finite-difference approximations at the two boundaries.

Polynomialization

We follow the steps described on the bottom of page 1312 of [2], the part referring to composition of functions.

Steps 1 and 2

We let

$$g_1(x) = e^x, g_2(x) = \gamma(1 - x), g_3(x) = \frac{1}{x},$$

and

$$h_{3i} = g_3(\theta_i), h_{2i} = g_2(h_{3i}), h_{1i} = g_1(h_{2i}),$$

for $i = 1, 2, ...N$. From these definitions $f(y_i, \theta_i) = y_i e^{\gamma(1-\frac{1}{\theta_i})}$ can be written as $y_i h_{1i}$.

Step 3

The new differential equations are(from the chain rule):

$$\dot{h_{3i}} = -\frac{1}{\theta_i^2}\dot{\theta}_i = -h_{3i}^2\dot{\theta}_i \tag{6}$$

$$\dot{h_{2i}} = -\gamma\dot{h_{3i}} = \gamma h_{3i}^2\dot{\theta}_i \tag{7}$$

$$\dot{h_{1i}} = h_{1i}\dot{h_{2i}} = \gamma h_{1i}h_{3i}^2\dot{\theta}_i \tag{8}$$

for $i = 1, 2, ...N$.

Quadraticization

From equation (1), the θ differential equations are(a sub i on a matrix or column denotes the ith row):

$$\dot{\theta}_i = (\mathbf{A}^D - \mathbf{A}^C - \beta\mathbf{I})_i\boldsymbol{\theta} + (\boldsymbol{b}_\theta)_i + \frac{BD}{\mu}y_i h_{1i}. \tag{9}$$

Plugging (9) into equations (6), (7), and (8) we see that these differential equations are not quadratic. To make them quadratic we follow the procedure in [2](pages 1313-1314) and introduce $3N$ new variables:

$$p_i = h_{3i}^2$$
$$q_i = h_{1i}h_{3i}^2 = h_{1i}p_i$$
$$r_i = h_{1i}^2h_{3i}^2 = h_{1i}q_i$$

or

$$0 = p_i - h_{3i}^2 \tag{10}$$

$$0 = q_i - h_{1i}p_i \tag{11}$$

$$0 = r_i - h_{1i}q_i \tag{12}$$

for $i = 1, ...N$. Now the new differential equations become:

$$\dot{h}_{3i} = -(p_i(\mathbf{A}^D - \mathbf{A}^C - \beta\mathbf{I})_i\boldsymbol{\theta} + (\mathbf{b}_\theta)_i p_i + \frac{BD}{\mu}y_i q_i) \tag{13}$$

$$\dot{h}_{2i} = \gamma(p_i(\mathbf{A}^D - \mathbf{A}^C - \beta\mathbf{I})_i\boldsymbol{\theta} + (\mathbf{b}_\theta)_i p_i + \frac{BD}{\mu}y_i q_i) \tag{14}$$

$$\dot{h}_{1i} = \gamma(q_i(\mathbf{A}^D - \mathbf{A}^C - \beta\mathbf{I})_i\boldsymbol{\theta} + (\mathbf{b}_\theta)_i q_i + \frac{BD}{\mu}y_i r_i) \tag{15}$$

Transformed System

In matrix form, the complete set of differential equations become:

$$\boldsymbol{E}\dot{\boldsymbol{x}} = \boldsymbol{G}\boldsymbol{x} + \boldsymbol{H}\boldsymbol{x}^{\boldsymbol{2}} + \bar{\boldsymbol{b}} \tag{16}$$

where $\boldsymbol{x} = (y_1, ..., y_N, \theta_1, ..., \theta_N, h_{31}, ..., h_{3N}, h_{21}, ...h_{2N}, h_{11}, ..., h_{1N}, p_1, ...p_N, q_1, ..., q_N, r_1, ..., r_N)^T \in \mathbb{R}^{8N}$.

$\boldsymbol{x}^2$ is defined as:

$$\boldsymbol{x}^2 = \begin{bmatrix} \boldsymbol{x}^{(1)} \\ \vdots \\ \boldsymbol{x}^{(N)} \end{bmatrix},$$

where

$$\boldsymbol{x}^{(i)} = x_i \begin{bmatrix} x_1 \\ \vdots \\ x_i \end{bmatrix}.$$

x_i is the ith element of $\boldsymbol{x}$, and $S = 8N(8N+1)/2$.

If x had n elements, then the $x_i x_j$ element of x^2 would be at the $(i-1)(i)/2+j$ position if $i \geq j$. Using equations (1), (13), (14), (15), (10), (11), and (12); $\boldsymbol{E} \in \mathbb{R}^{8N\times 8N}$, $\boldsymbol{G} \in \mathbb{R}^{8N\times 8N}$, and $\boldsymbol{H} \in \mathbb{R}^{8N\times S}$ can be constructed in blocks of N rows at a time.

$$\boldsymbol{E} = \begin{bmatrix} \boldsymbol{I} & \\ & \boldsymbol{0} \end{bmatrix},$$

where $\boldsymbol{I} \in \mathbb{R}^{5N\times 5N}$ and the rest of the matrix is zeros. $\boldsymbol{E}$ is referred to as the mass matrix.

The form of G and H is not obvious from matrices, but can be defined in Matlab more easily:

```
G = zeros(8*N, 8*N);
G(1:2*N,1:2*N) = A;
for i = 1:N
    G(2*N+i,5*N+i) = -b_theta(i);
    G(3*N+i,5*N+i) = gamma*b_theta(i);
    G(4*N+i,6*N+i) = gamma*b_theta(i);
end
G(5*N +1 : 8*N, 5*N + 1 :8*N) = eye(3*N, 3*N);
```

```
for i = 1:N
    H(i,(4*N+i-1)*(4*N+i)/2+i) = -D;
    H(N + i,(4*N+i-1)*(4*N+i)/2+i) = B*D/mu;
    H(2*N + i,(6*N+i-1)*(6*N+i)/2+i) = -B*D/mu;
    H(3*N + i,(6*N+i-1)*(6*N+i)/2+i) = gamma*B*D/mu;
    H(4*N + i,(7*N+i-1)*(7*N+i)/2+i) = gamma*B*D/mu;
    H(5*N + i,(2*N+i-1)*(2*N+i)/2+2*N+i) = -1;
    H(6*N + i,(5*N+i-1)*(5*N+i)/2+4*N+i) = -1;
    H(7*N + i,(6*N+i-1)*(6*N+i)/2+4*N+i) = -1;
    for j = 1:N
        H(2*N + i, (5*N+i-1)*(5*N+i)/2+N+j) = -A_1(i,j);
        H(3*N + i, (5*N+i-1)*(5*N+i)/2+N+j) = gamma*A_1(i,j);
        H(4*N + i, (6*N+i-1)*(6*N+i)/2+N+j) = gamma*A_1(i,j);
    end
end
```

$\bar{\boldsymbol{b}} \in \mathbb{R}^{8N\times 1}$ is defined as

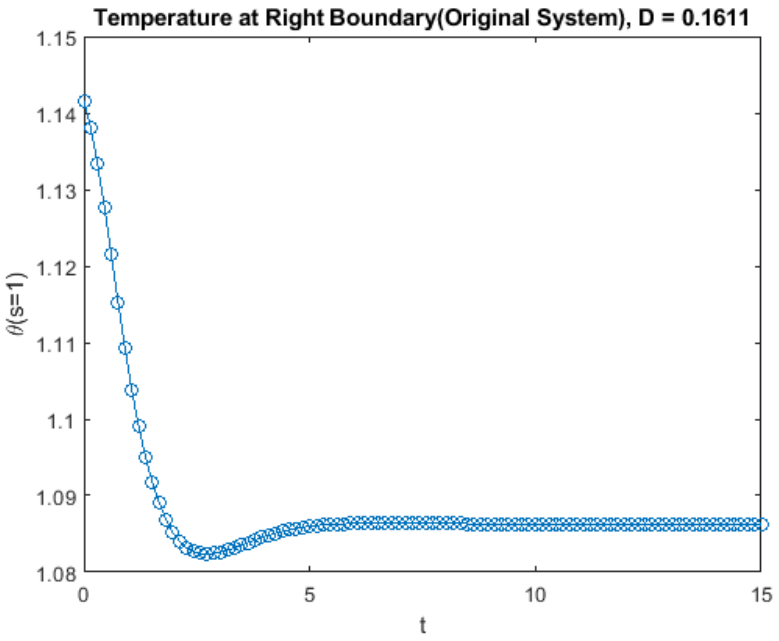
$$\bar{\boldsymbol{b}} = \begin{bmatrix} \boldsymbol{b} \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{17}$$

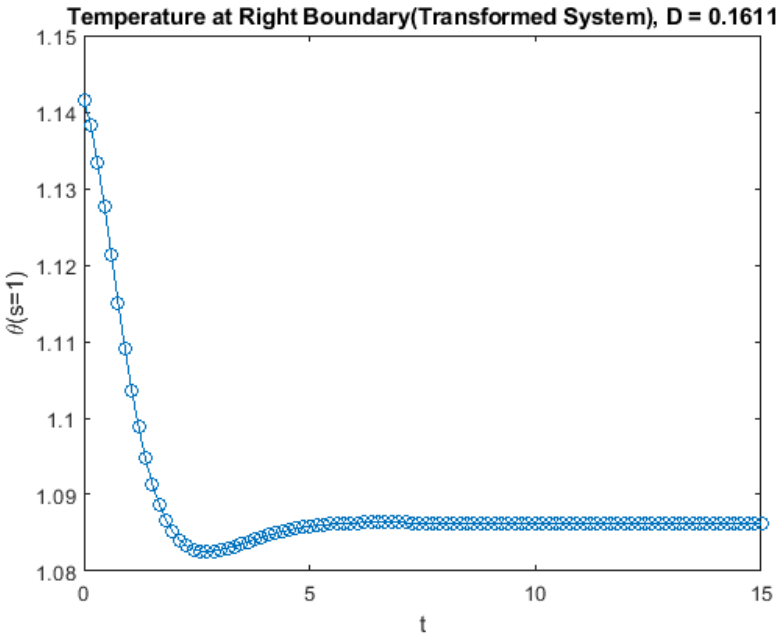
Discussion/Comparison

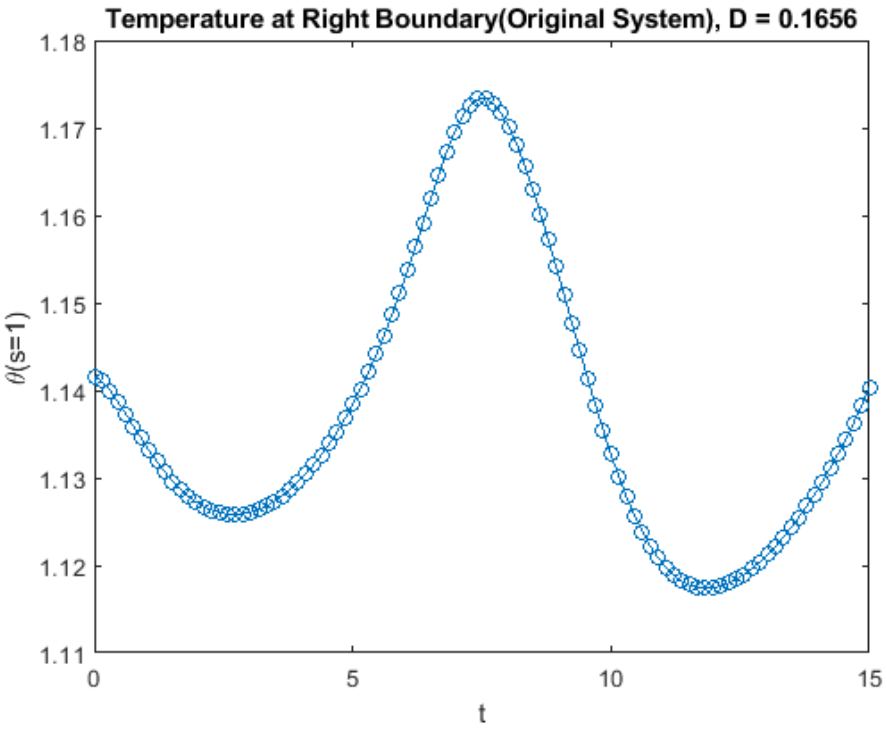
We used ode15s to timestep these quadraticized differential algebraic equations. When compared to timestepping the original system(equation (3.4) in [1]), the plots should be identical, as this is an exact transformation. It should be noted that the system went from size 2N to size 8N, with 3N new differential equations and 3N new algebraic equations. The simplification of the form of the equations(exponential terms to only quadratic terms) came at the price of more equations as well as algebraic ones. This simpler form of the new system could make it more amenable to certain types of model order reduction.

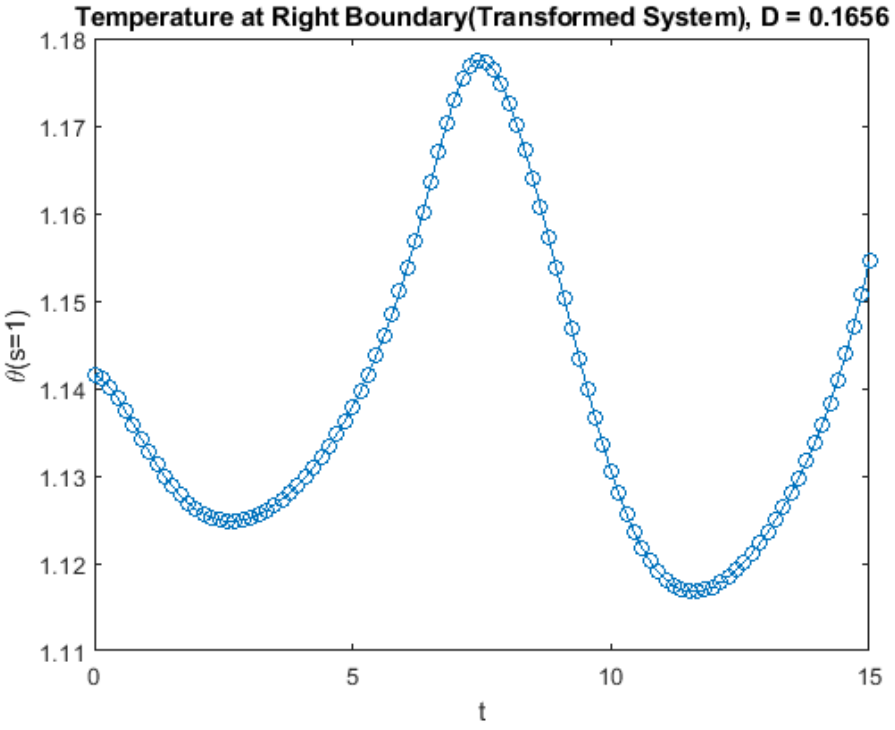
Limit Cycle Oscillation(LCO) Amplitude

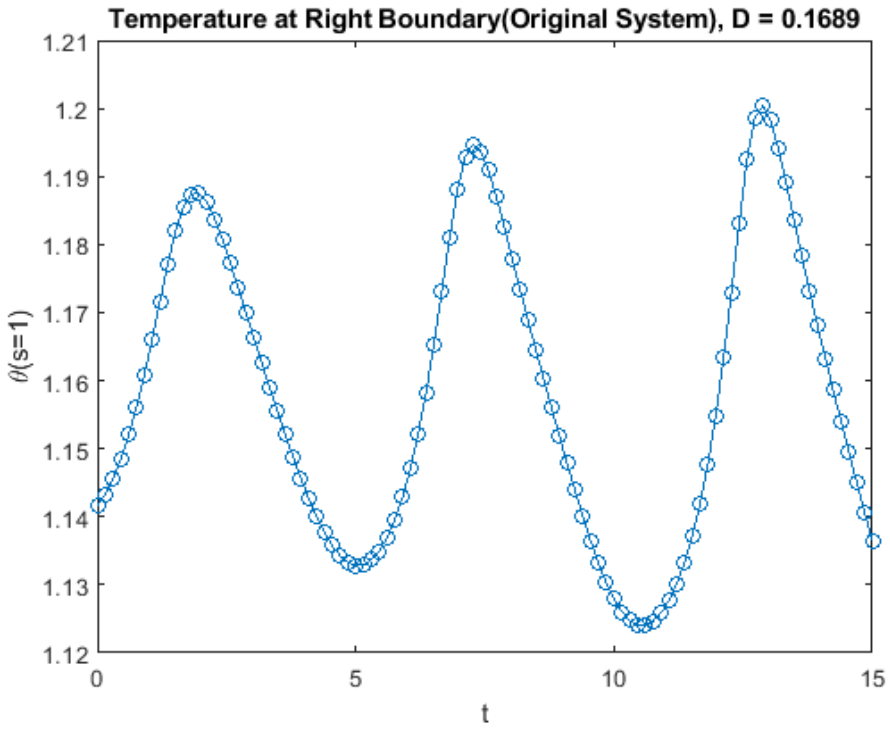
Below are plots of the temperature at the right boundary for various Damköhler numbers, as well as a comparison of the maximum temperature at the right boundary from t = 8 to 15 for ten Damköhler numbers chosen linearly from $D \in [.16, .17]$ between the original and transformed system. There are small discrepancies between the plots, and these could be attributed to either our ode solver or an error in implementation.

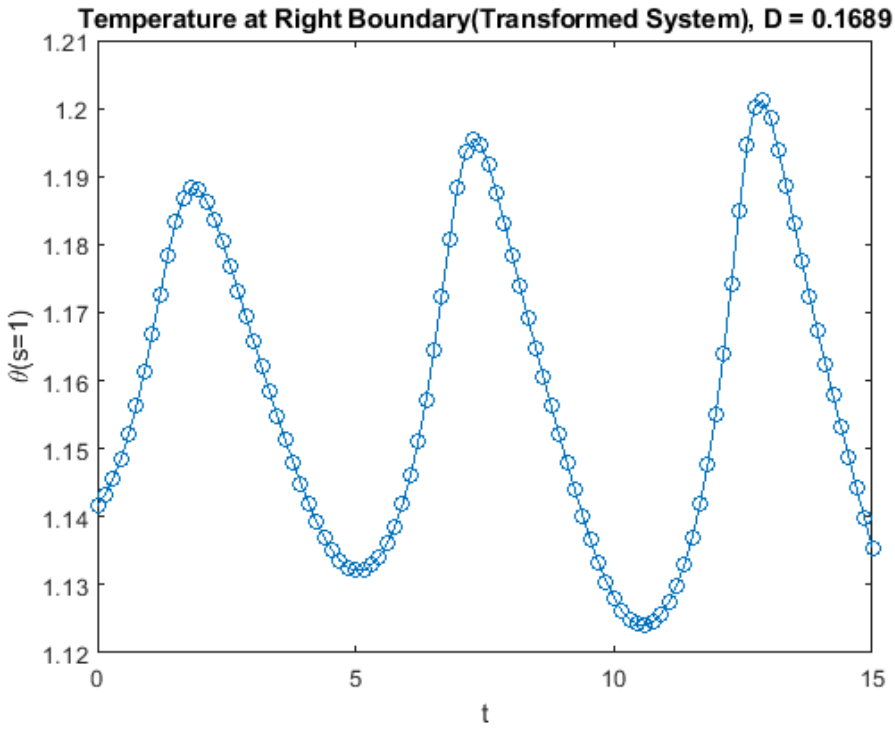

Temperature at Right Boundary(Original System), D = 0.1611
1.15
1.14
1.13
1.12
1.11
1.1
1.09
1.08
θ(s=1)
0
5
10
15
t

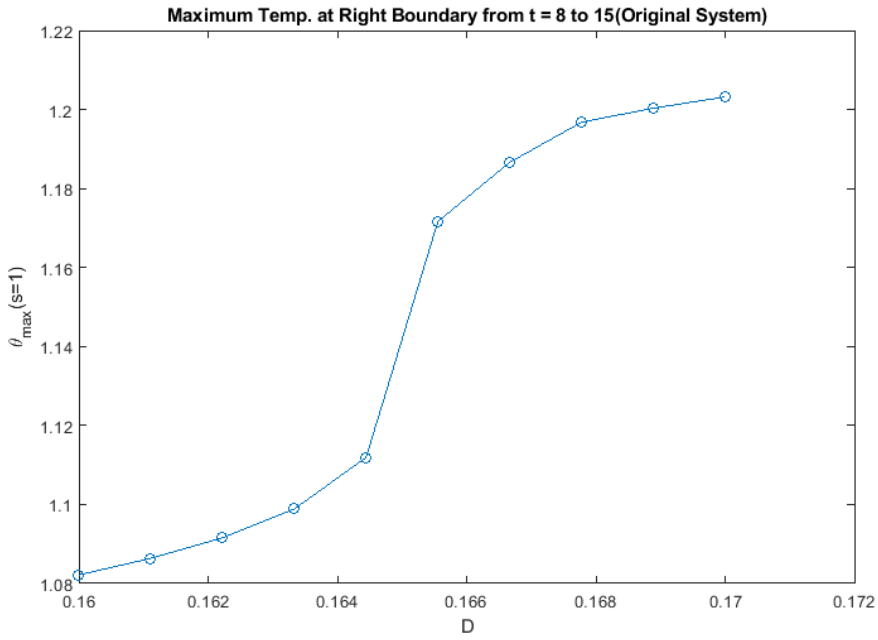

Temperature at Right Boundary(Transformed System), D = 0.1611
1.15
1.14
1.13
1.12
1.11
1.1
1.09
1.08
θ(s=1)
0
5
10
15
t

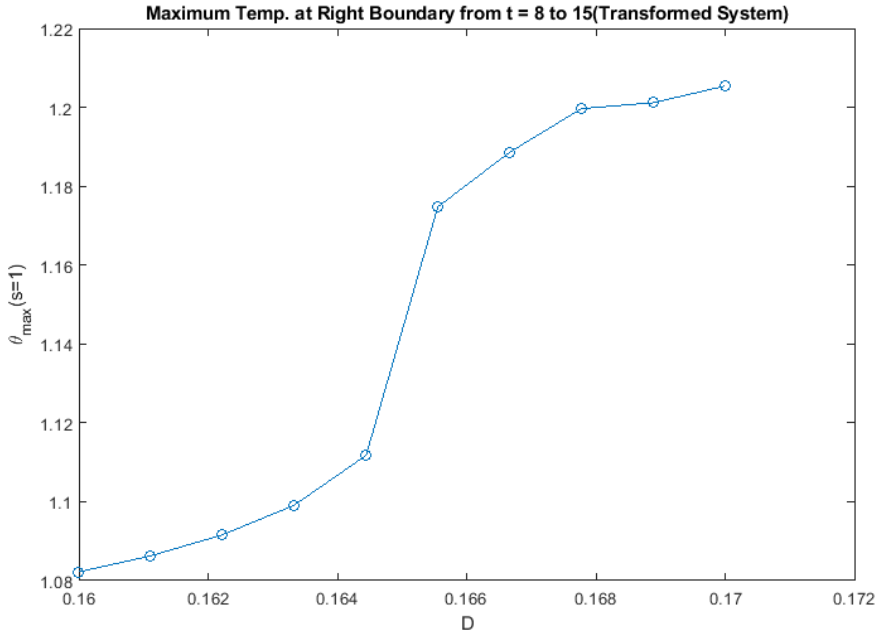

Temperature at Right Boundary(Original System), D = 0.1656
1.18
1.17
1.16
1.15
1.14
1.13
1.12
1.11
θ(s=1)
0
5
10
15
t


Temperature at Right Boundary(Transformed System), D = 0.1656
1.18
1.17
1.16
1.15
1.14
1.13
1.12
1.11
θ(s=1)
0
5
10
15
t


Temperature at Right Boundary(Original System), D = 0.1689
1.21
1.2
1.19
1.18
1.17
1.16
1.15
1.14
1.13
1.12
θ(s=1)
0
5
10
15
t

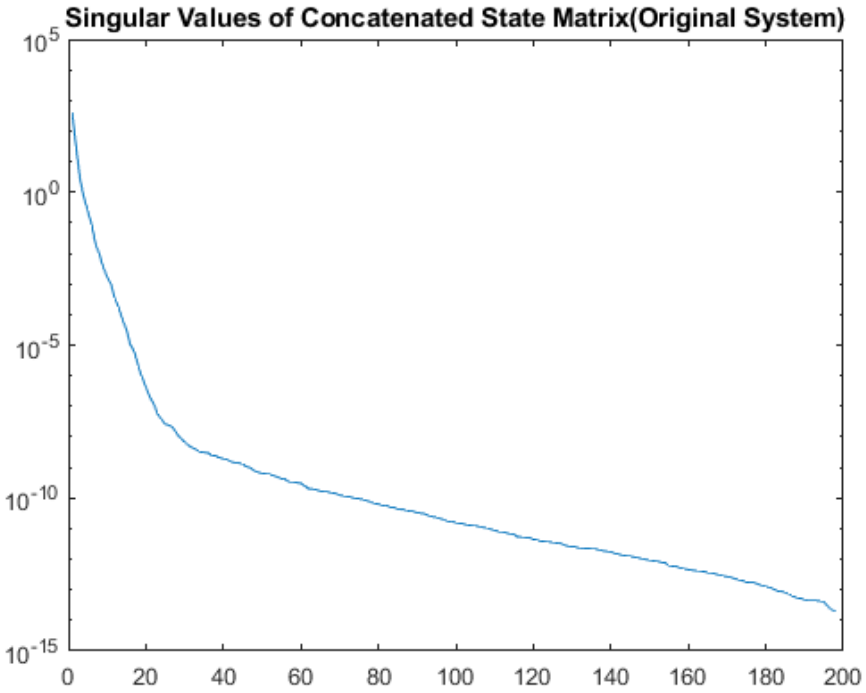

Temperature at Right Boundary(Transformed System), D = 0.1689
1.21
1.2
1.19
1.18
1.17
1.16
1.15
1.14
1.13
1.12
θ(s=1)
0
5
10
15
t

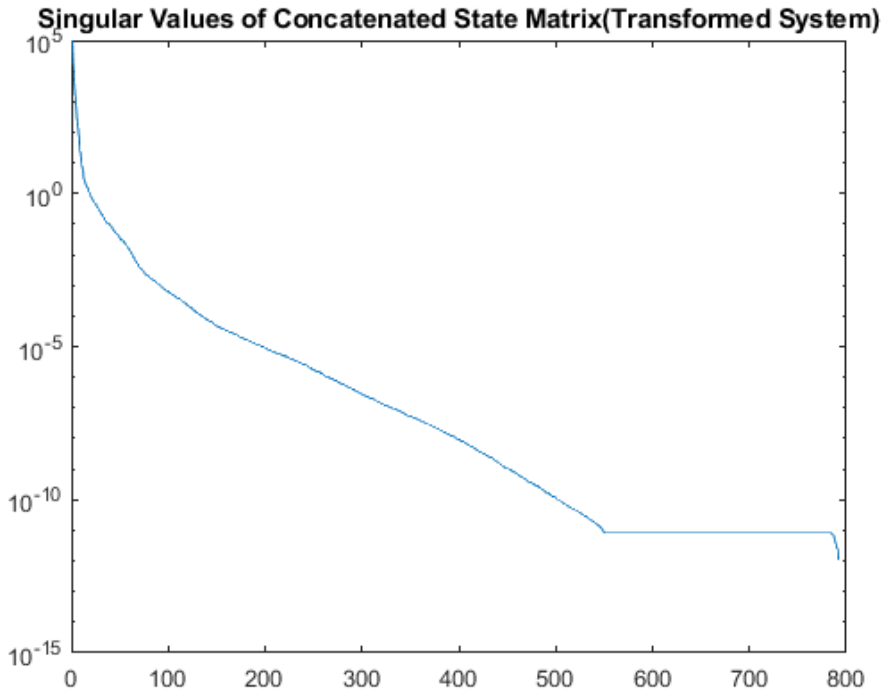

Maximum Temp. at Right Boundary from t = 8 to 15(Original System)
1.22
1.2
1.18
1.16
1.14
1.12
1.1
1.08
$\theta_{max}(s=1)$
0.16
0.162
0.164
0.166
0.168
0.17
0.172
D

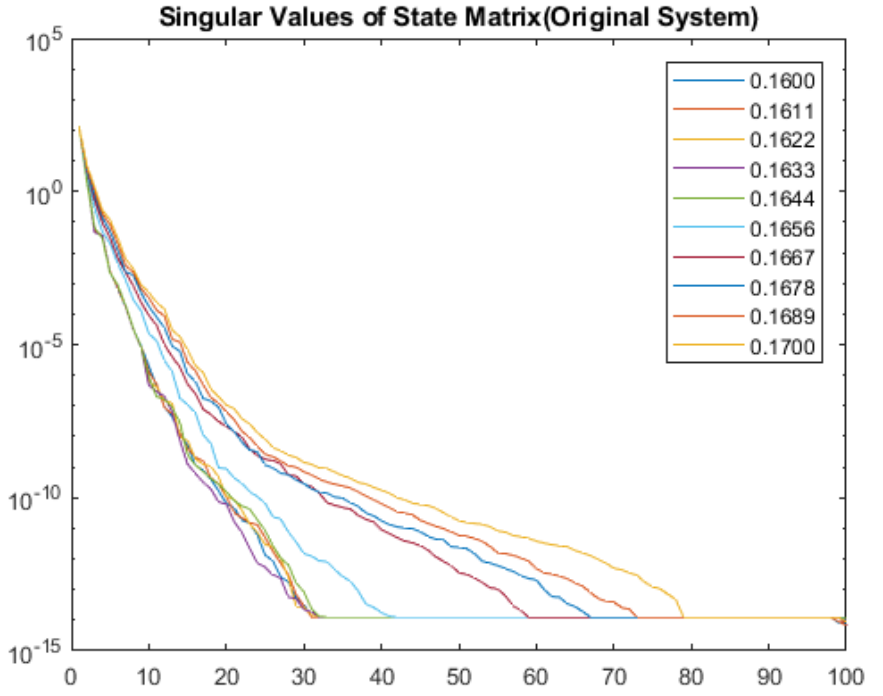

Maximum Temp. at Right Boundary from t = 8 to 15(Transformed System)
1.22
1.2
1.18
1.16
1.14
1.12
1.1
1.08
$\theta_{max}(s=1)$
0.16
0.162
0.164
0.166
0.168
0.17
0.172
D

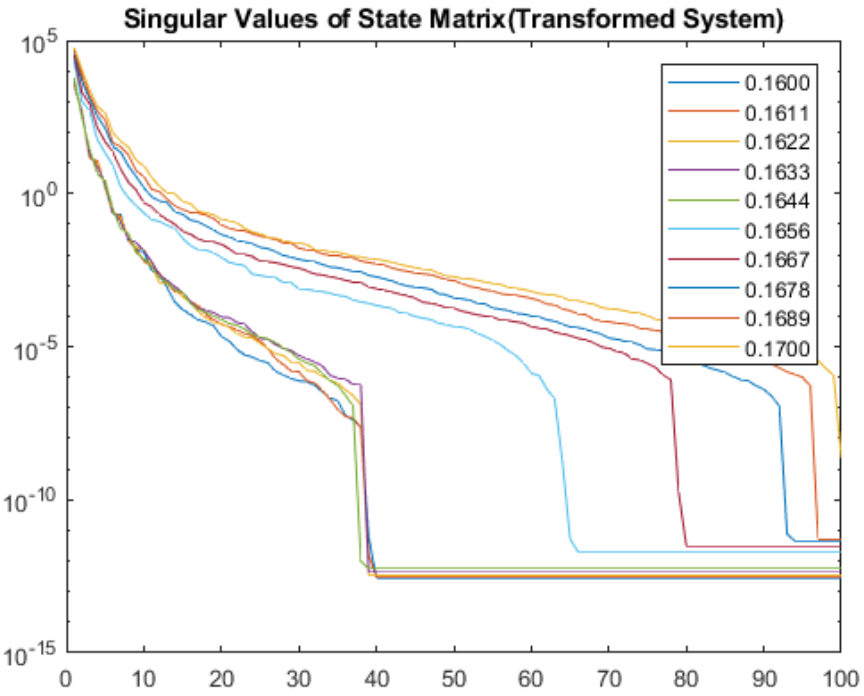
Singular Values

Below are plots of the largest singular values of the state matrices(matrices where each column corresponds to a timestep of the system) made with a 100 timesteps(equally spaced) between 0 and 15. The concatenated matrices were formed by adjoining 10 state matrices each with a different Damköhler number, spaced linearly from .16 to .17. Then we found the SVD of this larger matrix. This was done for both the original and transformed systems. We also plotted the singular values of the individual state matrices over the same range of .16 to .17 but on the same plots, for the original and transformed systems.



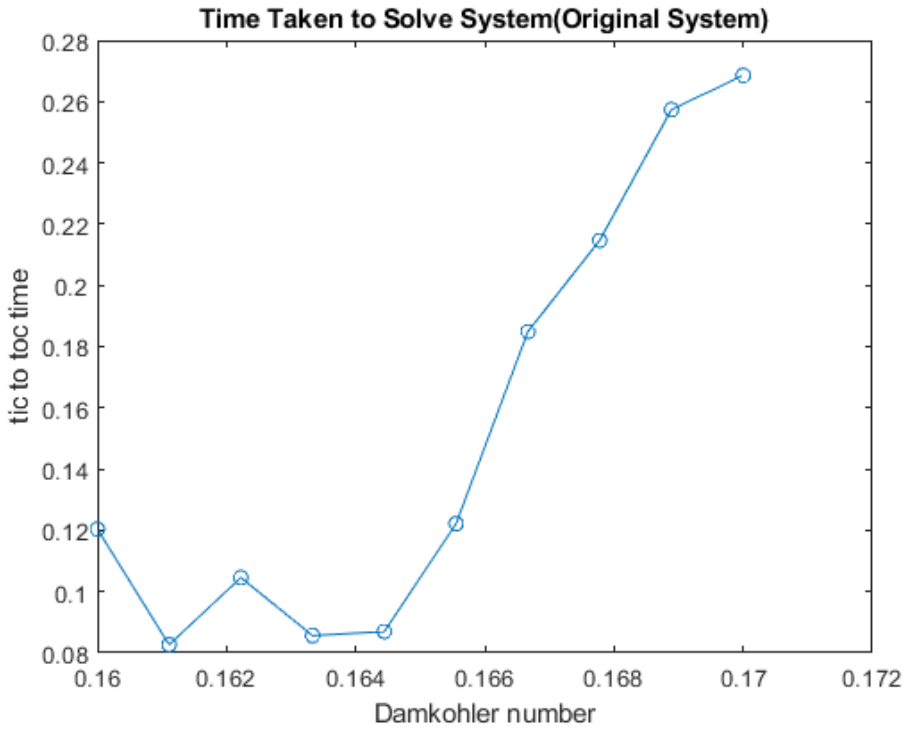

Singular Values of Concatenated State Matrix(Transformed System)
10^5
10^0
10^-5
10^-10
10^-15
0
100
200
300
400
500
600
700
800

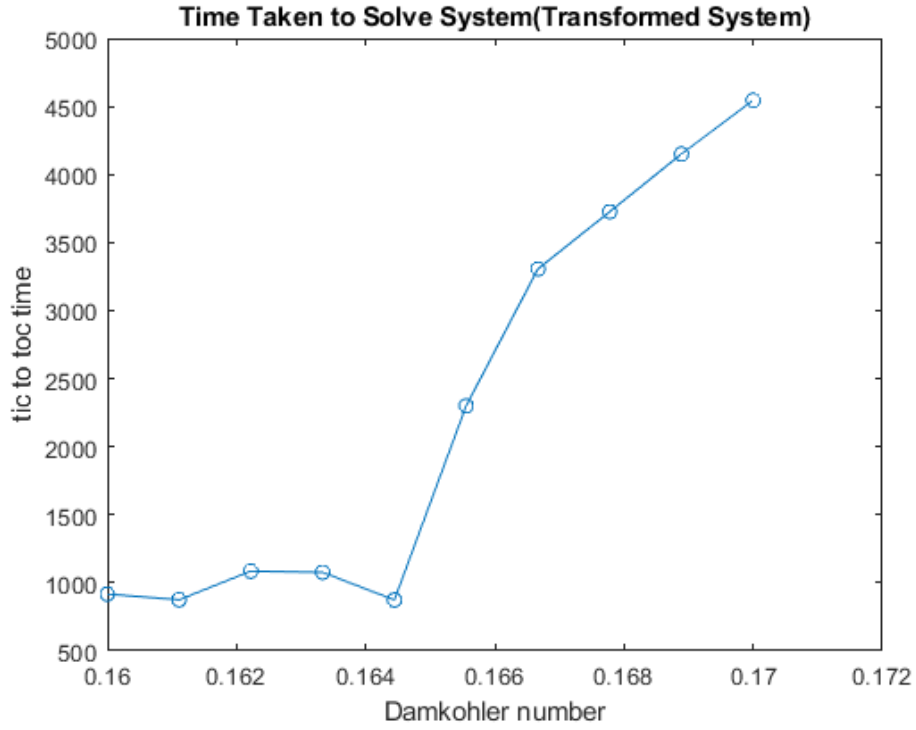

Singular Values of State Matrix(Original System)
10^5
10^0
10^-5
10^-10
10^-15
0
10
20
30
40
50
60
70
80
90
100
0.1600
0.1611
0.1622
0.1633
0.1644
0.1656
0.1667
0.1678
0.1689
0.1700



Computation Time

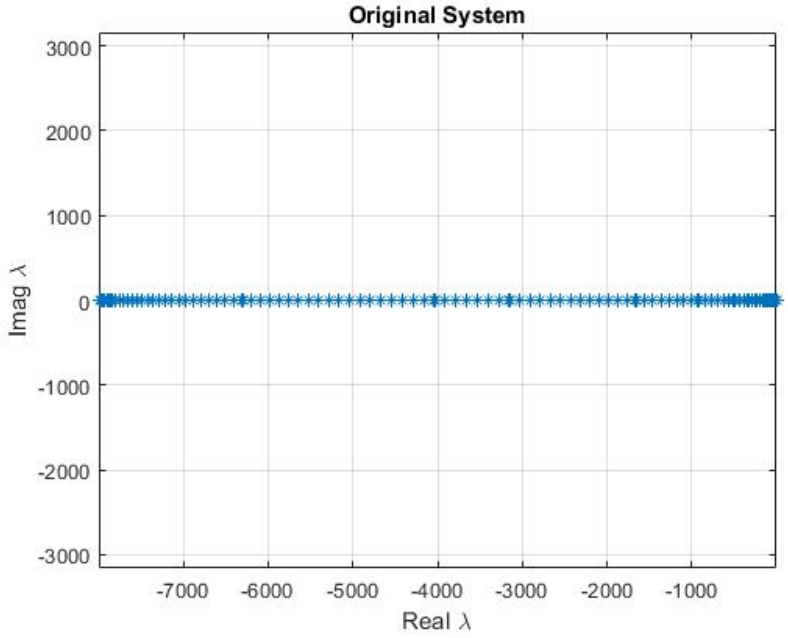
Below are plots of the time it took standard(no tuning) ode15s to solve the original and transformed system for various Damkölher numbers.

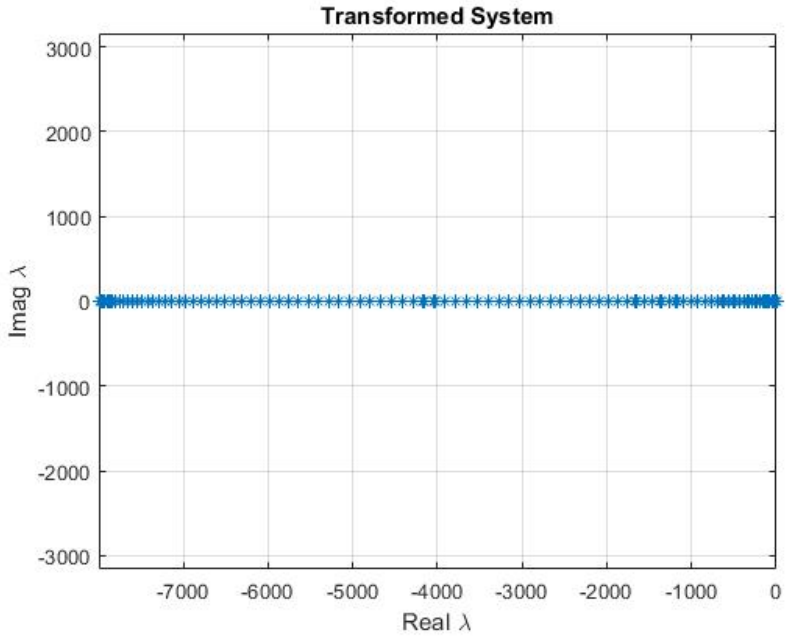

Time Taken to Solve System(Original System)
tic to toc time
Damkohler number
0.28
0.26
0.24
0.22
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.16
0.162
0.164
0.166
0.168
0.17
0.172


Time Taken to Solve System(Transformed System)
tic to toc time
Damkohler number
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0.16
0.162
0.164
0.166
0.168
0.17
0.172

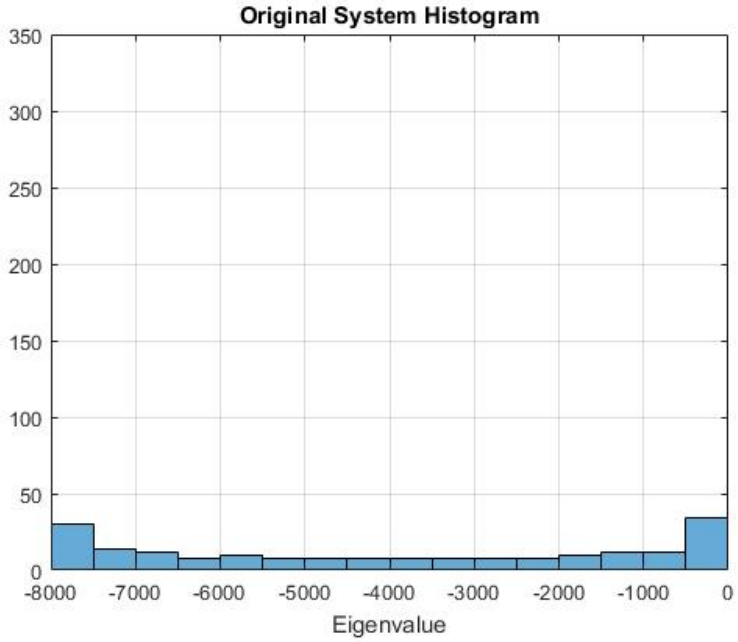
Stability Plots

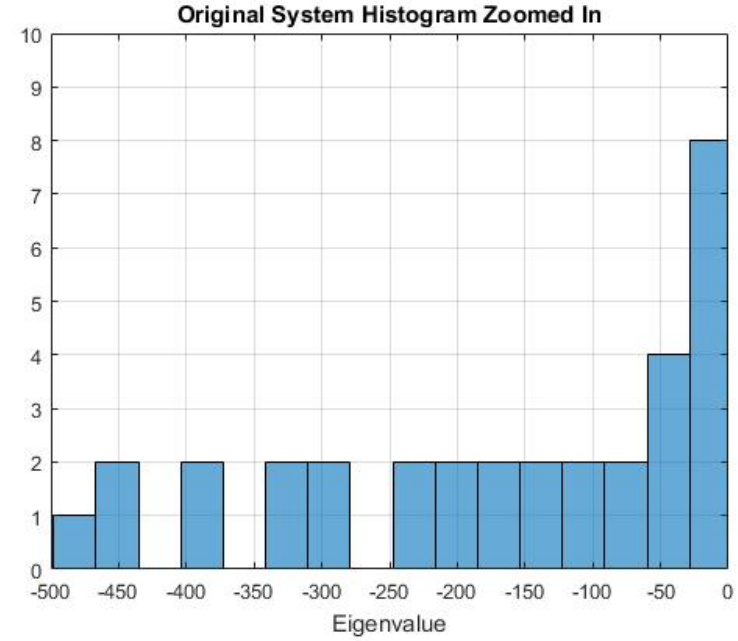
The stability of the eigen-values from the A matrix (from the original system) and the G matrix(from the transformed system) is analyzed. Below are the Imaginary vs. Real plots for the these two matrices. Because all the eigen-values are negative both systems are stable.

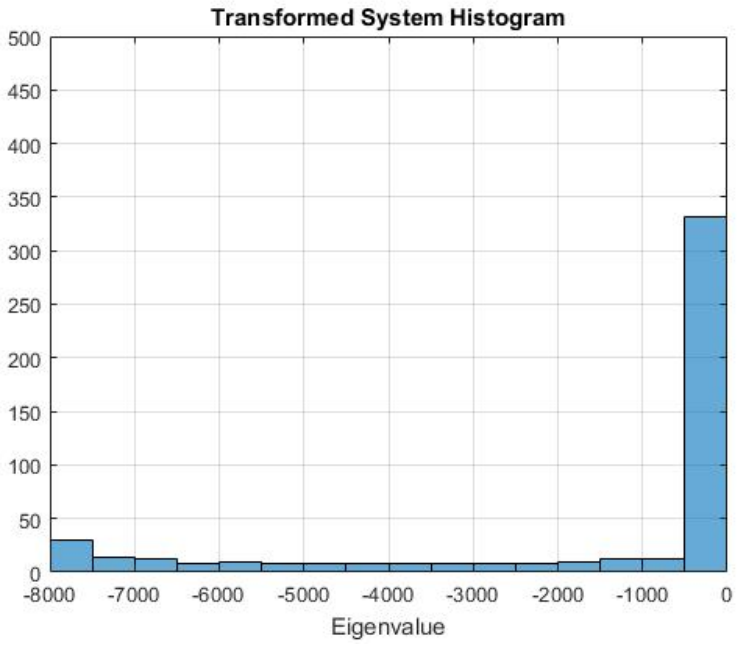


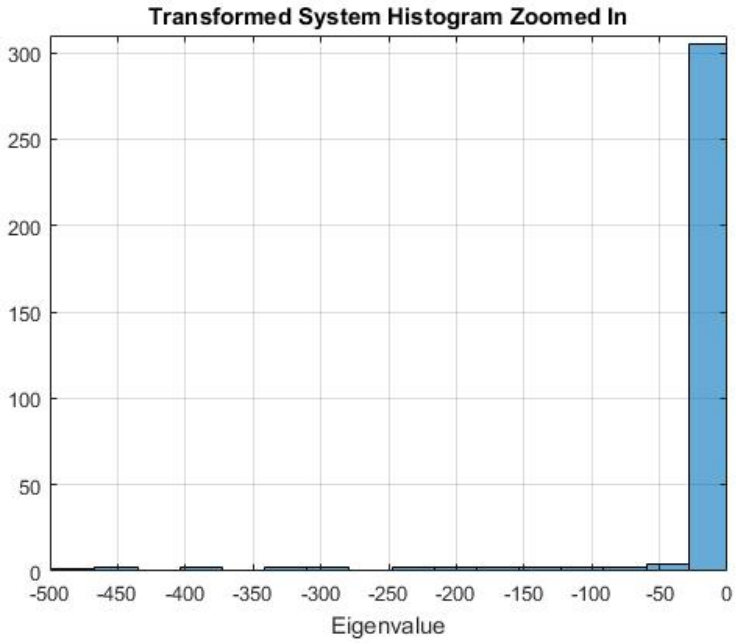


To determine how stable each system is histograms of the eigen-values are made. Below are the histograms of the original and transformed systems. The plots show that the transformed system has many more eigen-values close to zero than the original system. This means the the transformed system is less stable than the original system.

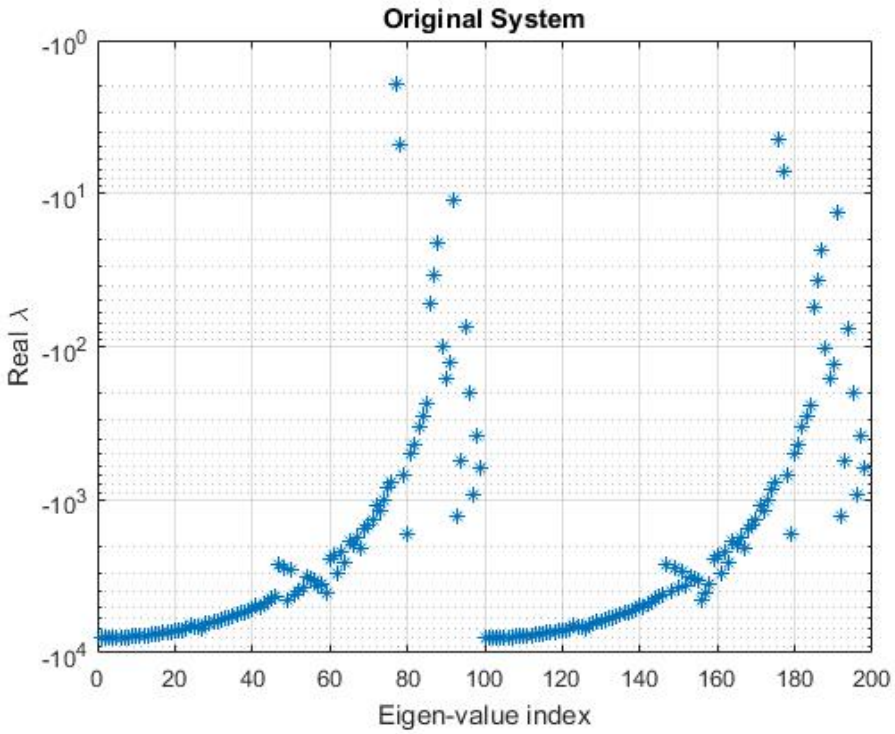


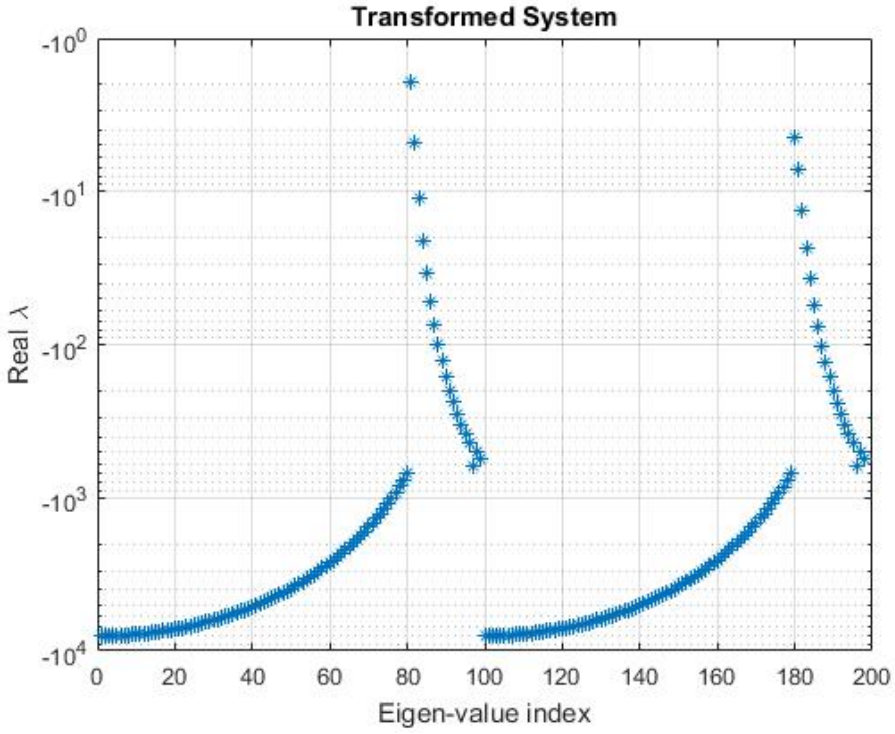






Log plots of the eigen-values for the original and transformed system are shown on the following page. Note that the zero-value eigen-values for the transformed system are not shown in the plot. This is because of the nature of the log plot function in matlab.


Original System
Real λ
Eigen-value index
-10^0
-10^1
-10^2
-10^3
-10^4
0
20
40
60
80
100
120
140
160
180
200


Transformed System
Real λ
Eigen-value index
-10^0
-10^1
-10^2
-10^3
-10^4
0
20
40
60
80
100
120
140
160
180
200

References:

[1] Zhou, Yuxiang Beckett. Model Reduction for Nonlinear Dynamical Systems with Parametric Uncertainties. 2012.

[2] Gu, Chenjie. QLMOR: A Projection-Based Nonlinear Model Order Reduction Approach Using Quadratic-Linear Representation of Nonlinear Systems. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 30, no. 9, 2011, pp. 13071320., doi:10.1109/tcad.2011.2142184.