Homework Submission

CMDA 3605 Mathematical Modeling: Methods and Tools I

Virginia Tech Honor Code Pledge:

"I have neither given nor received unauthorized assistance on this assignment."

Solution to Exercise 1

(a) $\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} x(t)$

The matrix is upper triangular, so the eigenvalues are just the values on the diagonal. Therefore, $\lambda_1 = 0$, $\lambda_2 = -1$.

Compute the corresponding eigenvectors:

$\lambda_1 = 0$:

$$det(A - 0I) = \begin{bmatrix} 0-0 & 1 \\ 0 & -1-0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{aligned} 0x + 1y &= 0 \\ 0x - 1y &= 0 \\ y &= 0x \end{aligned}$$

Therefore, $v_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

$\lambda = -1$

$$det(A - (-1)I) = \begin{bmatrix} 0-(-1) & 1 \\ 0 & -1-(-1) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{aligned} x + y &= 0 \\ y &= -x \end{aligned}$$

Therefore, $v_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

Now calculate the weights using x_0:

$$\begin{bmatrix}1\\1\end{bmatrix} = \begin{bmatrix}1\\0\end{bmatrix} w_1 + \begin{bmatrix}1\\-1\end{bmatrix} w_2$$

$$\begin{aligned} w_1 + w_2 &= 1 \\ 0w_1 - w_2 &= 1 \end{aligned}$$

From the second equation, it can be determined that w_2 equals -1. As a result, substituting w_2 back into the first equation:

$$\begin{aligned} w_1 - 1 &= 1 \\ w_1 &= 2 \end{aligned}$$

Since the $\alpha(A) = 0$ (spectral abscissa), the dynamical system is stable and the solution formula will look like this as t approaches infinity:

$$x(t) = e^{0t} 2 \begin{bmatrix}1\\0\end{bmatrix}$$

So the non-zero steady state is as follows:

$$\hat{x} = 1 \times \begin{bmatrix}1\\0\end{bmatrix} \times 2 = \begin{bmatrix}2\\0\end{bmatrix}$$

(b) $\dot{x}(t) = \begin{bmatrix}-2 & 0 & 3\\0 & -0.5 & -3\\0 & 0 & -1\end{bmatrix} x(t) + \begin{bmatrix}3\\1\\1\end{bmatrix}$, x(0) $= \begin{bmatrix}1\\0\\-1\end{bmatrix}$

The matrix is upper triangular, so the eigenvalues are just the values on the diagonal. Therefore, $\lambda_1 = -0.5$, $\lambda_2 = -1$, $\lambda_3 = -2$.

Since the system is a non-homogeneous linear continuous time system, $\hat{x} = -A^{-1}b$:
if $\hat{x}$ is a steady state then:

$$\begin{aligned} A\hat{x} + b &= 0 \\ A\hat{x} &= -b \end{aligned}$$

$$\begin{bmatrix}-2 & 0 & 3\\0 & -0.5 & -3\\0 & 0 & -1\end{bmatrix} \begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix} = \begin{bmatrix}-3\\-1\\-1\end{bmatrix}$$

$$\begin{aligned} -2x_1 + 0x_2 + 3x_3 &= -3 \\ 0x_1 - 0.5x_2 - 3x_3 &= -1 \\ 0x_1 + 0x_2 - x_3 &= -1 \end{aligned}$$

From the system of equations, $x_3 = 1$, calculating x_1, x_2. First use the second equation:

$$\begin{aligned}
-0.5x_2 - 3(1) &= -1 \\
-0.5x_2 &= 2 \\
x_2 &= -4
\end{aligned}$$

Lastly, find x_1 using equation 1:

$$\begin{aligned}
-2x_1 + 3(1) &= -3 \\
-2x_1 &= -6 \\
x_1 &= 3
\end{aligned}$$

Since the $\alpha(A)$ = -0.5 (spectral abscissa), the dynamical system will converge to steady state $\hat{x} = -A^{-1}b$:

$$\hat{x} = \begin{bmatrix} 3 \\ -4 \\ 1 \end{bmatrix}$$

Solution to Exercise 2

(a) The power method converges to (λ_3, v_3) because that is the eigenvalue with the largest magnitude. The rate of convergence is:

$$|\frac{7}{-8}| \approx 0.875$$

(b)

$$\mu_1 = \frac{1}{\lambda_1 - \mu} = \frac{1}{7-5} = \frac{1}{2}$$

$$\mu_2 = \frac{1}{\lambda_2 - \mu} = \frac{1}{-2-5} = \frac{1}{-7}$$

$$\mu_3 = \frac{1}{\lambda_3 - \mu} = \frac{1}{-8-5} = \frac{1}{-13}$$

$$\mu_4 = \frac{1}{\lambda_4 - \mu} = \frac{1}{-4-5} = \frac{1}{-9}$$

$$\mu_5 = \frac{1}{\lambda_5 - \mu} = \frac{1}{1-5} = \frac{1}{-4}$$

$$\mu_6 = \frac{1}{\lambda_6 - \mu} = \frac{1}{-6-5} = \frac{1}{-11}$$

The iteration will converge to λ_1 since it is the dominant eigenvalue of the new matrix $B = (A - \mu I)^{-1}$. The speed of convergence is given by:

$$|\frac{\mu_5}{\mu_1}| = |\frac{-1/4}{1/2}| = \frac{1}{2}$$

(c) Repeat the same process:

$$\mu_1 = \frac{1}{\lambda_1 - \mu} = \frac{1}{7 - (-5)} = \frac{1}{12}$$

$$\mu_2 = \frac{1}{\lambda_2 - \mu} = \frac{1}{-2 - (-5)} = \frac{1}{3}$$

$$\mu_3 = \frac{1}{\lambda_3 - \mu} = \frac{1}{-8 - (-5)} = \frac{1}{-3}$$

$$\mu_4 = \frac{1}{\lambda_4 - \mu} = \frac{1}{-4 - (-5)} = \frac{1}{1} = 1$$

$$\mu_5 = \frac{1}{\lambda_5 - \mu} = \frac{1}{1 - (-5)} = \frac{1}{6}$$

$$\mu_6 = \frac{1}{\lambda_6 - \mu} = \frac{1}{-6 - (-5)} = \frac{1}{-1} = -1$$

Since there is no unique dominant eigenvalue of $B = (A - \mu I)^{-1}$, i.e, λ_6 and λ_4 are equally close to the shift mu, the system will not converge. We could try changing the shift to ensure convergence. Depending on the type of system, the convergence could oscillate, since $\mu_6 = -1$

and the calculated convergence rate for this situation would be:

$$|\frac{\mu_6}{\mu_4}| = |\frac{-1}{1}| = 1$$

Solution to Exercise 3

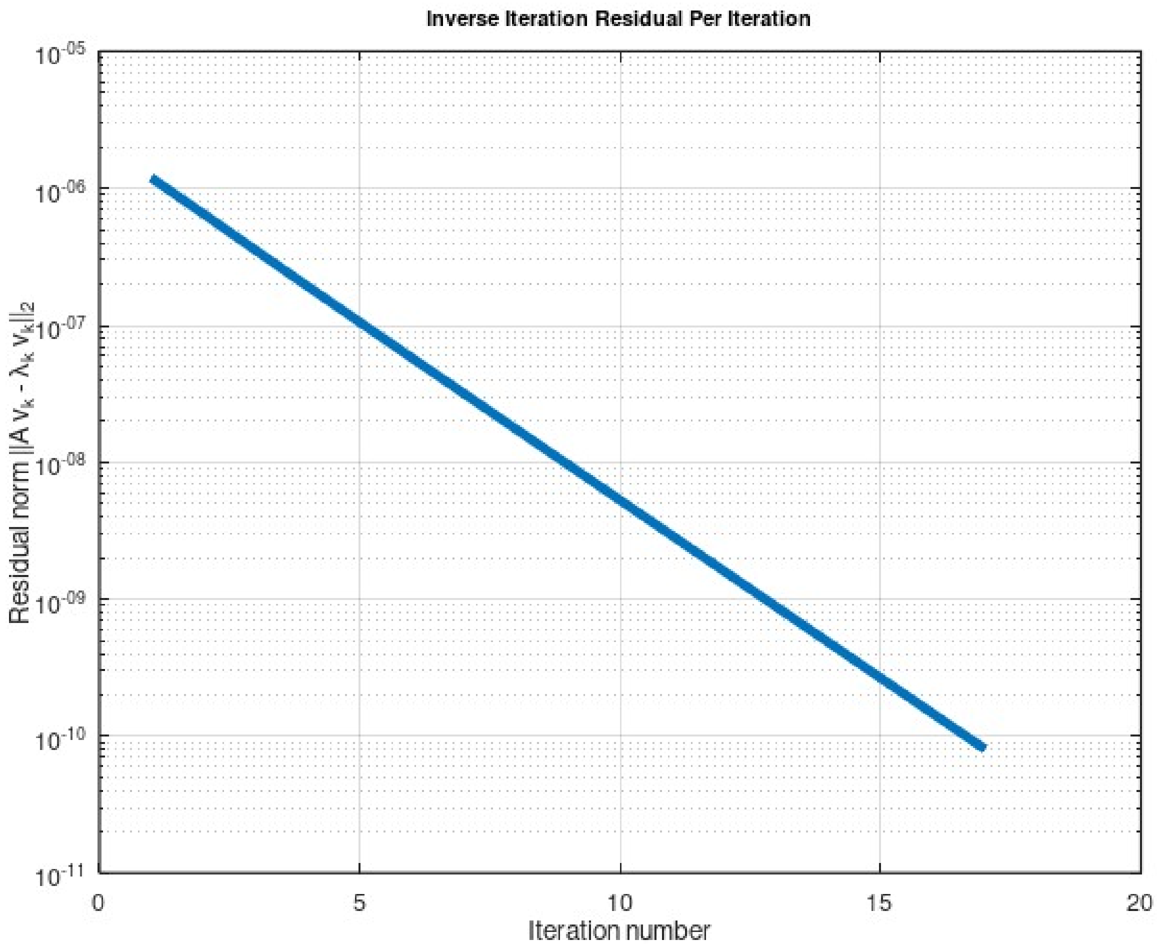
(a)

```
function [lambda, v, res] = inverseiteration(A, mu, x0, maxiter, tol)
%INVERSE ITERATION Computes dominant eigenvalue closest to shift mue
%
% INPUTS:
%   A       - matrix of dimensions n x n
%   x0      - starting vector of dimension n (x 1)
%   maxiter - maximum number of iteration steps
%   tol     - convergence tolerance for the residual.
%   mu      - potentially complex shift
%
% OUTPUTS:
%   lambda - eigenvalue closest to mu
%   v      - corresponding eigenvector
%   res    - array of computed residuals

% Normalize the starting vector and save the result in v.
v =  x0 / norm(x0, 2);

% Residual memory.
res = zeros(1, maxiter);

% Iteration.
for k = 1:maxiter
    % Apply the matrix.
    v = (A - mu*eye(size(A))) \ v;

    % Normalize the iteration vector.
    v = v / norm(v, 2);

    % Compute the eigenvalue estimate.
    lambda = v' * A * v;

    % Convergence criterion.
    res(k) = norm(A * v - lambda * v, 2);
    if res(k) < tol
        break;
    end
end

% Prepare outputs.
res = res(1:k);
```

Solution to Exercise 3 cont.

(b)

```
eigenvalue clostest to mu shift 0:
-1.0626e-05
>> |
```



Solution to Exercise 4

(a)

For a nonlinear discrete-time system $x(k+1) = f(x(k))$, a steady state occurs when

$$\hat{x} = x(k+1) = x(k) = f(\hat{x}).$$

For part (a) we have:

$$f(x(k)) = a\,x(k)e^{-rx(k)}.$$

Now solve for $\hat{x} = f(\hat{x})$:

$$\hat{x} = a\hat{x}e^{-r\hat{x}}.$$

Immediately, a real steady state is

$$\hat{x}_1 = 0.$$

If $\hat{x} \neq 0$, we can divide both sides by $\hat{x}$:

$$1 = ae^{-r\hat{x}}.$$

$$\frac{1}{a} = e^{-r\hat{x}}.$$

Now take the natural log of each side to get:

$$\ln\left(\frac{1}{a}\right) = -r\hat{x}.$$

$$\hat{x} = \frac{\ln(1/a)}{-r}.$$

Because $\ln(1/a) = -\ln(a)$ the result is positive:

$$\hat{x_2} = \frac{\ln(a)}{r}.$$

Solution to Exercise 4 cont.

(b)

For part (b), we will solve again for $\hat{x} = f(\hat{x})$ on:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \dfrac{ax_1x_2}{1+x_1} \\ \dfrac{bx_1x_2}{1+x_2} \end{bmatrix}$$

Again we can notice that one real solution to $\hat{x} = f(\hat{x})$ is:

$$\hat{x_1} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

If $\hat{x}_1$ and $\hat{x}_2 \neq 0$:
First, simplify each equation:

First equation:

$$\left(\frac{1}{x_1}\right) x_1 = \frac{ax_1x_2}{1+x_1}\left(\frac{1}{x_1}\right)$$

$$(1+x_1)\cdot 1 = \frac{ax_2}{1+x_1}(1+x_1)$$

$$1 + x_1 = ax_2$$

$$x_1 = ax_2 - 1$$

Second equation:

$$\left(\frac{1}{x_2}\right) x_2 = \frac{bx_1x_2}{1+x_2}\left(\frac{1}{x_2}\right)$$

$$(1+x_2)\cdot 1 = \frac{bx_1}{1+x_2}(1+x_2)$$

$$1 + x_2 = bx_1$$

$$x_2 = bx_1 - 1$$

Now we can back substitute x_2 into the equation for x_1 to solve for x_1:

$$x_1 = a(bx_1 - 1) - 1$$

$$x_1 = abx_1 - a - 1$$

$$x_1 - abx_1 = -a - 1$$

$$x_1(1 - ab) = -a - 1$$

$$x_1 = \frac{-a - 1}{1 - ab}$$

Now, plug this into the equation for x_2:

$$x_2 = b\left(\frac{-a - 1}{1 - ab}\right) - 1$$

Therefore, the steady state is:

$$\hat{x_2} = \begin{bmatrix} \frac{-a - 1}{1 - ab} \\ b\left(\frac{-a - 1}{1 - ab}\right) - 1 \end{bmatrix}$$

Solution to Exercise 5

(a)

```
function [xr, res] = newtonmethod(g, dg, x0, maxiter, tol)
%Newtons Method Computes roots of scalar non linear equation.
%
% INPUTS:
%   g       - function handle
%   dg      - first derrivative function handle
%   x0      - starting point
%   maxiter - maximum number of iteration steps
%   tol     - tolerance for convergence criterion
%
% OUTPUTS:
%   xr     - root such that g(xr) approx = 0
%   res    - array of computed residuals

% Not normalizing for scalar root finding.
xr =  x0;

% Residual memory.
res = zeros(1, maxiter);

% Iteration.
for k = 1:maxiter
    gx = g(xr);
    dgx = dg(xr);

    % saving residual
    res(k) = abs(gx);


    if res(k) < tol
       res = res(1:k);
       return;
    end
    % making sure not divide by zero (no zero derrivative)
    dgx = dg(xr);
        if dgx == 0            % safeguard
            error('Newton fails: g''(x) = 0 at x = %g.', xr);
        end

    %newton update/iteration
    xr = xr - gx / dgx;

end

% If here, hit maxiter without meeting tol
    res = res(1:maxiter);
    warning('Max iterations reached: |g(x)| = %g at x = %g.', res(end), xr);
% Prepare outputs.
res = res(1:k);
```

Solution to Exercise 5 cont.

(b)

What are two different roots of the function $g(x) = e^x - x - 2$?

Using the Newton's Method function starting as $x_0 = 3$ and $x_0 = -3$ we get:

```
two different roots of the function
1.1462
-1.8414
```

Now we check which root converges the implementation faster by looking at the array approximation residuals, res:

```
how many steps it took:
For X0 = 3:
 Columns 1 through 7:

   15.0855    4.9023    1.3738    0.2654    0.0188    0.0001    0.0000

 Column 8:

         0
For x0 = -3:
   1.0498   0.0455   0.0002   0.0000         0
```

This shows that root -1.8414 converges the implementation faster at the fourth step, unlike root 1.1462 which converges on the seventh step.