
Project_Code_1d.m

By Joshua Norlin This project code calculates the characteristic exponents of the rotor system without reversed flow for values $B=1$ and $p=1$, determines the stability of the system, and plots the results. Stable1 and Stable2 contain a true/false analysis for each characteristic exponent's stability, and Nu contains the characteristic exponent pairs. For $\gamma=0:16$, values 1:17 correspond with $\mu=0$, values 18:34 correspond with $\mu=1$, and values 35:51 correspond with $\mu=2$.

```
t = 0:pi/8:2*pi;
conditions1 = [1 0];
conditions2 = [0 1];
Q = []; % Matrix of transition matrices; Values 1:34 are for mu=0,
35:69
% are for mu=1, 70:102 are for mu=2.
Lambda = []; % Eigenvalues of transition matrices
Nu = []; % Natural logarithms of transition matrices
Stable1 = []; % Values of stability for Nu, 1=stable, 0=unstable
Stable2 = []; % Values of stability for Nu, 1=stable, 0=unstable
% Values 1:17 are for mu=0, gamma=0:16, 18:34 are for mu=1, and 35:51
are
% for mu=2.
for mu = 0:2
    for gamma = 0:16
        [t,beta2] =
ode45(@(t,beta)rhs(t,beta,mu,gamma),t,conditions2);
        [t,beta1] =
ode45(@(t,beta)rhs(t,beta,mu,gamma),t,conditions1);
        Beta1 = beta1(:,1);
        Beta2 = beta2(:,1);
        dBeta1dt = beta1(:,2);
        dBeta2dt = beta2(:,2);
        period1a = Beta1(length(t));
        period1b = dBeta1dt(length(t));
        period2a = Beta2(length(t));
        period2b = dBeta2dt(length(t));
        q = [period1a period2a; period1b period2b];
        lambda = eig(q);
        nu = (1/(2*pi))*log(lambda);
        if real(nu(1)) <= 0
            stable1 = 1;
        elseif real(nu(1)) > 0
            stable1 = 0;
        end
        Stable1 = [Stable1 stable1];
        if real(nu(2)) <= 0
            stable2 = 1;
        elseif real(nu(2)) > 0
            stable2 = 0;
        end
        Stable2 = [Stable2 stable2];
        Q = [Q q];
        Lambda = [Lambda lambda];
    end
end
```

```

        Nu = [Nu nu];
    end
end
figure
hold on
mu = 0:2;
gamma = 0:16;
subplot(2,1,1)
for i = 1:length(gamma)
    plot3(mu(1),gamma(i),real(Nu(:,i)),'+')
    hold on
    plot3(mu(2),gamma(i),real(Nu(:,17+i)),'o')
    hold on
    plot3(mu(3),gamma(i),real(Nu(:,34+i)),'*')
    hold on
end
title('Real Nu')
xlabel('mu')
ylabel('gamma')
zlabel('real(Nu)')
subplot(2,1,2)
for i = 1:length(gamma)
    plot3(mu(1),gamma(i),imag(Nu(:,i)),'+')
    hold on
    plot3(mu(2),gamma(i),imag(Nu(:,17+i)),'o')
    hold on
    plot3(mu(3),gamma(i),imag(Nu(:,34+i)),'*')
    hold on
end
title('Imaginary Nu')
xlabel('mu')
ylabel('gamma')
zlabel('imag(Nu)')
Nu
Stable1
Stable2

function dbetadt = rhs(t,beta,mu,gamma)
B = 1;
p = 1;
K1 = gamma.*((B.^3).*mu.*cos(t)/6 + (B.^2).*(mu.^2).*sin(2.*t)/8);
K = K1;
C1 = gamma.*((B.^4)/8 + (B.^3).*mu.*sin(t)/6);
C = C1;
dbetadt_1 = beta(2);
dbetadt_2 = -(p.^2 + K).*beta(1) - C.*beta(2);
dbetadt = [dbetadt_1; dbetadt_2];
end

Nu =

```

Columns 1 through 4

$-0.0000 + 0.0000i$ $-0.0625 + 0.0019i$ $-0.1250 + 0.0078i$ $-0.1875 + 0.0177i$
 $-0.0000 - 0.0000i$ $-0.0625 - 0.0019i$ $-0.1250 - 0.0078i$ $-0.1875 - 0.0177i$

Columns 5 through 8

$-0.2500 + 0.0318i$ $-0.3125 + 0.0501i$ $-0.3750 + 0.0730i$ $-0.4375 + 0.1008i$
 $-0.2500 - 0.0318i$ $-0.3125 - 0.0501i$ $-0.3750 - 0.0730i$ $-0.4375 - 0.1008i$

Columns 9 through 12

$-0.5000 + 0.1340i$ $-0.5625 + 0.1732i$ $-0.6250 + 0.2194i$ $-0.6875 + 0.2738i$
 $-0.5000 - 0.1340i$ $-0.5625 - 0.1732i$ $-0.6250 - 0.2194i$ $-0.6875 - 0.2738i$

Columns 13 through 16

$-0.7500 + 0.3385i$ $-0.8125 + 0.4170i$ $-0.8751 + 0.4841i$ $-0.9375 + 0.3480i$
 $-0.7500 - 0.3385i$ $-0.8125 - 0.4170i$ $-0.8751 - 0.4841i$ $-0.9375 - 0.3480i$

Columns 17 through 20

$-0.9979 + 0.0000i$ $-0.0000 + 0.0000i$ $-0.0313 + 0.0000i$ $-0.0628 + 0.0000i$
 $-1.0021 + 0.0000i$ $-0.0000 - 0.0000i$ $-0.0937 + 0.0000i$ $-0.1872 + 0.0000i$

Columns 21 through 24

$-0.0948 + 0.0000i$ $-0.1278 + 0.0000i$ $-0.1621 + 0.0000i$ $-0.1986 + 0.0000i$
 $-0.2802 + 0.0000i$ $-0.3723 + 0.0000i$ $-0.4630 + 0.0000i$ $-0.5516 + 0.0000i$

Columns 25 through 28

$-0.2385 + 0.0000i$ $-0.2840 + 0.0000i$ $-0.3391 + 0.0000i$ $-0.4133 + 0.0000i$
 $-0.6366 + 0.0000i$ $-0.7160 + 0.0000i$ $-0.7865 + 0.0000i$ $-0.8367 + 0.0000i$

Columns 29 through 32

$-0.5424 + 0.0000i$ $-0.7502 + 0.2428i$ $-0.5425 + 0.5000i$ $-0.4295 + 0.5000i$
 $-0.8325 + 0.0000i$ $-0.7502 - 0.2428i$ $-1.0669 + 0.5000i$ $-1.3594 + 0.5000i$

Columns 33 through 36

$-0.3673 + 0.5000i$ $-0.3236 + 0.5000i$ $-0.0000 + 0.0000i$ $0.0622 + 0.0000i$
 $-1.4185 + 0.5000i$ $-1.5913 + 0.5000i$ $-0.0000 - 0.0000i$ $-0.1872 + 0.0000i$

Columns 37 through 40

$0.1230 + 0.0000i$ $0.1809 + 0.0000i$ $0.2354 + 0.0000i$ $0.2863 + 0.0000i$
 $-0.3729 + 0.0000i$ $-0.5556 + 0.0000i$ $-0.7316 + 0.0000i$ $-0.9026 + 0.0000i$

Columns 41 through 44

$0.3339 + 0.0000i$ $0.3785 + 0.0000i$ $0.4203 + 0.0000i$ $0.4597 + 0.0000i$
 $-1.0809 + 0.0000i$ $-1.2604 + 0.0000i$ $-1.3534 + 0.0000i$ $-1.6079 + 0.0000i$

Columns 45 through 48

$0.4970 + 0.0000i$ $0.5323 + 0.0000i$ $0.5655 + 0.0000i$ $0.5967 + 0.0000i$
 $-1.6484 + 0.0000i$ $-1.4189 + 0.5000i$ $-1.4140 + 0.5000i$ $-1.7306 + 0.5000i$

Columns 49 through 51

$0.6256 + 0.0000i$ $0.6519 + 0.0000i$ $0.6751 + 0.0000i$
 $-1.6315 + 0.5000i$ $-1.4180 + 0.5000i$ $-1.5367 + 0.0000i$

Stable1 =

Columns 1 through 13

1 1 1 1 1 1 1 1 1 1 1
1 1

Columns 14 through 26

1 1 1 1 1 1 1 1 1 1 1
1 1

Columns 27 through 39

1 1 1 1 1 1 1 1 1 0 0
0 0

Columns 40 through 51

0 0 0 0 0 0 0 0 0 0 0 0
 0

Stable2 =

Columns 1 through 13

1 1 1 1 1 1 1 1 1 1 1 1
 1 1

Columns 14 through 26

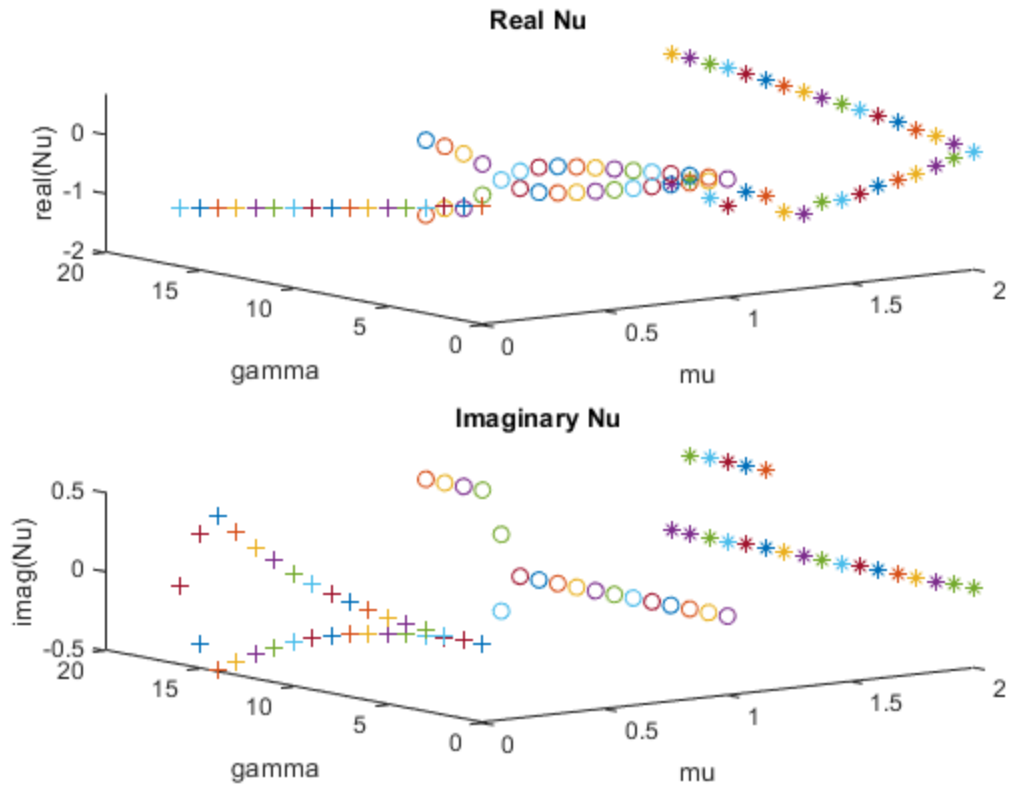
1 1 1 1 1 1 1 1 1 1 1 1
 1 1

Columns 27 through 39

1 1 1 1 1 1 1 1 1 1 1 1
 1 1

Columns 40 through 51

1 1 1 1 1 1 1 1 1 1 1 1
 1



Published with MATLAB® R2019a