Computational Methods

Francisco Castillo
Homework 5

April 12, 2018

Problem 1

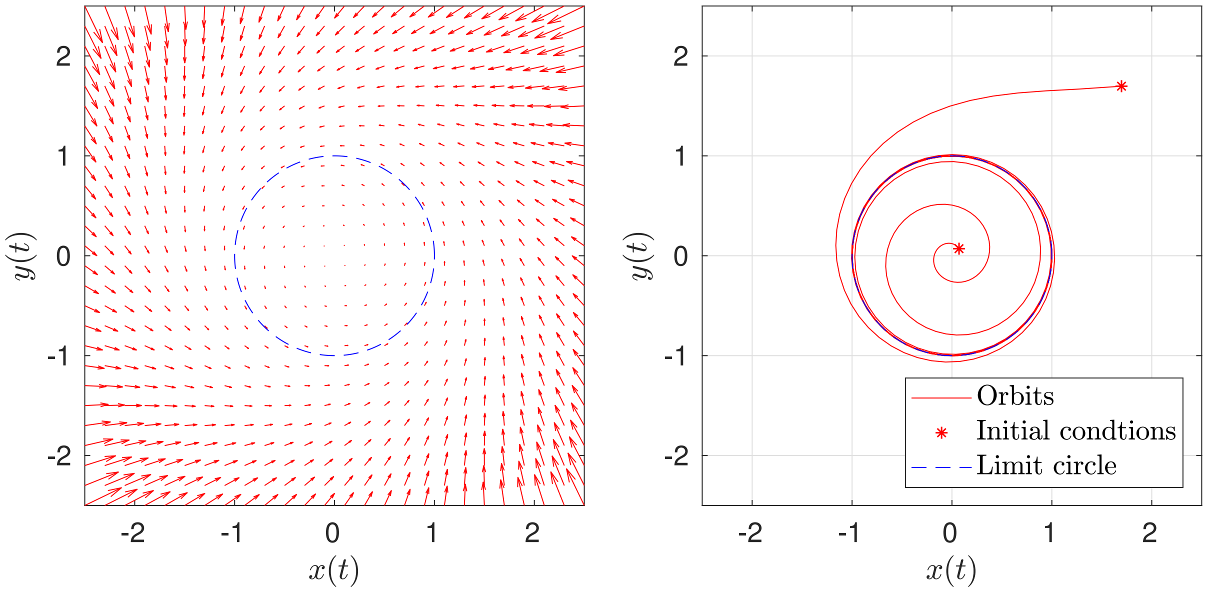
Compute te solution to the systems for the given initial conditions using `rk4.m`. Plot your results in the phase plane.

---

1. **Using initial data conditions with $x(0)^2 + y(0)^2$ both smaller and larger than 1, solve**

$$x'(t) = -4y + x(1 - x^2 - y^2),$$
$$y'(t) = 4x + y(1 - x^2 - y^2),$$

over the interval $0 \leq t \leq 10$. Given the system of ODEs, we see that when at the unit circle the parenthesis of the right hand side vanishes. Moreover, if we express the ODEs in polar coordinates, we see that, at the unit circle, we obtain $dr = 0$ and $d\theta = 4$. This implies that the orbits, once they reach the unit circle, the stay indefinitely there since $dr = 0$ and moving counterclockwise since $d\theta = 4$. Inside the unit circle the vector field is divergent, it points away from the origin. This implies that the orbits are gonna approach the unit circle if we place the initial condition inside it. We have the inverse situation outside the unit circle. Thus, the unit circle constitutes an $\omega - \textit{limit}$ set of this system of ODEs. In the next figure we can see the behaviour described and the vector field.



(a) Vector field

(b) Orbits

Figure 1: Solutions of the system of ODEs.

Matlab code for this problem

```
%% Problem 1a
legendfontsize=14;
axisfontsize=16;

F = @(t,V) [-4*V(2)+V(1).*(1-V(1).^2-V(2).^2);...
                4*V(1)+V(2).*(1-V(1).^2-V(2).^2)];
theta=pi/4;
R=[0.1; 2.4];
tspan=[0 10];

N=300;
a=2.5;
for i=1:length(R)
    V0 = [R(i)*cos(theta); R(i)*sin(theta)];
    [t,V] = rk4(F,tspan,V0,N);
    figure(1)
    l(1)=plot(V(:,1),V(:,2),'r');
    hold on
    l(2)=plot(V0(1),V0(2),'r*');
    axis([-a a -a a])
    grid on
end
% Plot orbits
```

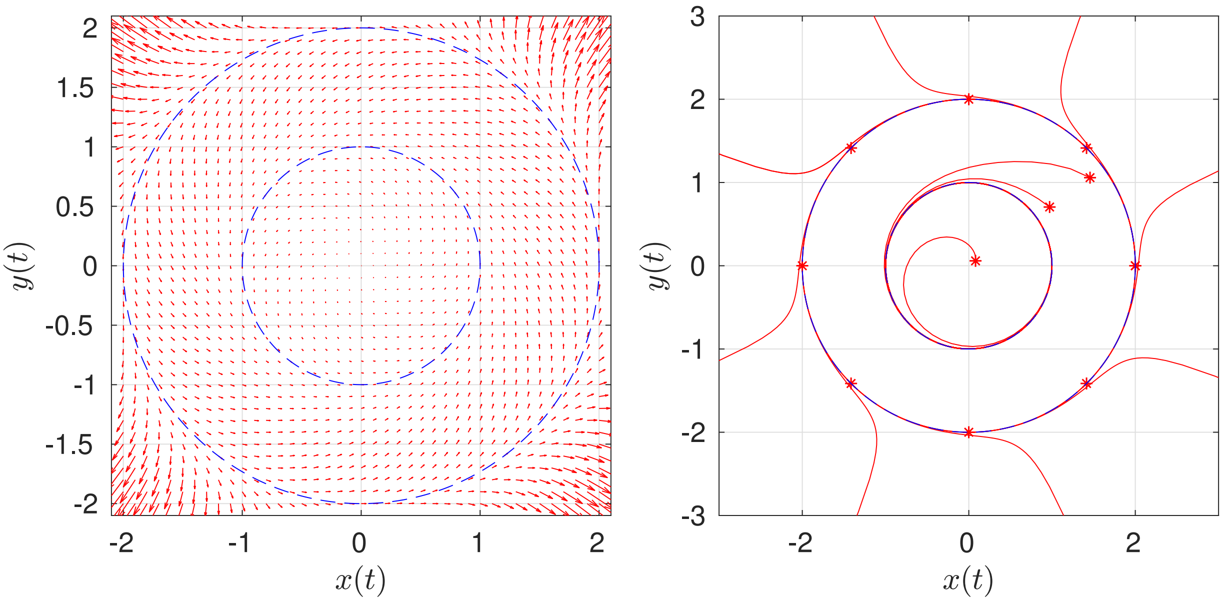
```
figure(1)
hold on
l(3)=plot(cos(0:0.001:2*pi),sin(0:0.001:2*pi),'--b');
xlabel('$x(t)$','interpreter','latex')
ylabel('$y(t)$','interpreter','latex')
pbaspect([1 1 1])
set(gca,'fontsize',14)
legend([l(1) l(2) l(3)],{' Orbits',' Initial condtions',' Limit circle'},...
 'Interpreter','latex','fontsize',legendfontsize,'location','SouthEast')
txt='Latex/FIGURES/P1_1';
saveas(gcf,txt,'epsc')

% Plot vector field
step=0.2;
[x,y] = meshgrid(-a:step:a,-a:step:a);
u=-4*y+x.*(1-x.^2-y.^2);
v=4*x+y.*(1-x.^2-y.^2);
figure(2)
q=quiver(x,y,u,v,'r');
hold on
plot(cos(0:0.001:2*pi),sin(0:0.001:2*pi),'--b');
xlabel('$x(t)$','interpreter','latex')
ylabel('$y(t)$','interpreter','latex')
pbaspect([1 1 1])
axis([-a a -a a])
set(gca,'fontsize',14)
set(q,'AutoScale','on', 'AutoScaleFactor', 1.5)
txt='Latex/FIGURES/P1_1field';
saveas(gcf,txt,'epsc')
```

2. **Using initial data conditions with $x(0)^2 + y(0)^2$ both inside and outside of the circles of radius 1 and 2, solve**

$$\begin{aligned} x'(t) &= -4y + x(1 - x^2 - y^2)(4 - x^2 - y^2), \\ y'(t) &= 4x + y(1 - x^2 - y^2)(4 - x^2 - y^2), \end{aligned}$$

over the interval $0 \leq t \leq 10$.

Given the system of ODEs, we see that when at the unit circle or at the circle of radius 2, the second term of the right hand side cancels in both ODEs. This give us the same condition as before, once the orbits reach those circles, they stay there indefinitely. However, the addition of the extra factor makes the vector field divergent outside the outer circle, and changes the behaviour of the system. If the initial condition is outside the outter circl, the solutions are unstable. We can see the vector field and the orbits of the solutions in the next figure. Note that the initial conditions outisde the circle of radius 2 are placed just very close to the limit.



(a) Vector field

(b) Orbits

Figure 2: Solutions of the system of ODEs.

Matlab code for this problem

```
%% Problem 1b
close all
F = @(t,V) [-4*V(2)+V(1).*(1-V(1).^2-V(2).^2).*(4-V(1).^2-V(2).^2);...
                4*V(1)+V(2).*(1-V(1).^2-V(2).^2).*(4-V(1).^2-V(2).^2)];
theta=pi/5;
R=[0.1; 1.2; 1.8];
tspan=[0 10];
N=300;
a=3;
for i=1:length(R)
    V0 = [R(i)*cos(theta); R(i)*sin(theta)];
    [t,V] = rk4(F,tspan,V0,N);
    figure(1)
    plot(V(:,1),V(:,2),’r’)
    hold on
    plot(V0(1),V0(2),’r*’)
    axis([-a a -a a])
    grid on
end
Theta=0:pi/4:2*pi;
for i=1:length(Theta)
    V0 = [2*cos(Theta(i)); 2*sin(Theta(i))];
    [t,V] = rk4(F,[0 1],V0,N);
    figure(1)
    l(1)=plot(V(:,1),V(:,2),’r’);
    hold on
    l(2)=plot(V0(1),V0(2),’r*’);
    axis([-a a -a a])
    grid on
end

figure(1)
hold on
l(3)=plot(cos(0:0.001:2*pi),sin(0:0.001:2*pi),’--b’);
plot(2*cos(0:0.001:2*pi),2*sin(0:0.001:2*pi),’--b’);
xlabel(’$x(t)$’,’interpreter’,’latex’)
ylabel(’$y(t)$’,’interpreter’,’latex’)
pbaspect([1 1 1])
set(gca,’fontsize’,14)
%legend([l(1) l(2) l(3)],{’ Orbits’,’ Initial condtions’,’ Limit circle’},...
% ’Interpreter’,’latex’,’fontsize’,legendfontsize,’location’,’SouthEast’)
txt=’Latex/FIGURES/P1_2’;
saveas(gcf,txt,’epsc’)

% Plot vector field
step=0.1;
b=2;
[x,y] = meshgrid(-b:step:b,-b:step:b);
u=-4*y+x.*(1-x.^2-y.^2).*(4-x.^2-y.^2);
v=4*x+y.*(1-x.^2-y.^2).*(4-x.^2-y.^2);
figure(2)
q=quiver(x,y,u,v,’r’);
set(q,’AutoScale’,’on’, ’AutoScaleFactor’, 3)
hold on
% step=0.1;
% b=2.1;
% [x,y] = meshgrid(-b:step:b,-b:step:b);
% u=-4*y+x.*(1-x.^2-y.^2).*(4-x.^2-y.^2);
% v=4*x+y.*(1-x.^2-y.^2).*(4-x.^2-y.^2);
% q=quiver(x,y,u,v,’r’);
% step=0.05;
% b=1.2;
% [x,y] = meshgrid(-b:step:b,-b:step:b);
% u=-4*y+x.*(1-x.^2-y.^2).*(4-x.^2-y.^2);
% v=4*x+y.*(1-x.^2-y.^2).*(4-x.^2-y.^2);
% q=quiver(x,y,u,v,’r’);
% set(q,’AutoScale’,’on’, ’AutoScaleFactor’, 0.5)
plot(cos(0:0.001:2*pi),sin(0:0.001:2*pi),’--b’);
plot(2*cos(0:0.001:2*pi),2*sin(0:0.001:2*pi),'--b');
xlabel('$x(t)$','interpreter','latex')
ylabel('$y(t)$','interpreter','latex')
pbaspect([1 1 1])
grid on
a=2.1;
axis([-a a -a a])
set(gca,'fontsize',14)
txt='Latex/FIGURES/P1_2field';
saveas(gcf,txt,'epsc')
```

Problem 2

Consider the wave propagation PDE solved in HW 3,

$$\begin{aligned} u_t &= p_x, \\ p_t &= u_x, \quad x \in (0,\pi), t > 0, \end{aligned}$$

with boundary conditions $u(t,0) = u(t,\pi) = 0$ **and initial conditions** $u(0,x) = e^{-30(x-\pi/2)^2}$ **and** $p(0,x) = 0$**.**

---

a) **Show that the eigenvalues of the differential operator $\mathcal{L}$ are integers in the imaginary axis.**

Proof. By making use of the PDE,

$$u_{xx} = \lambda p_x = \lambda^2 u.$$

Let $\lambda^2 = -k^2$. Then,

$$u = A\cos(kx) + iB\sin(kx).$$

Applying the boundary conditions,

$$u(t,0) = A = 0,$$

and

$$u(t,\pi) = iB\sin(k\pi) = 0,$$

which implies either $B = 0$ (trivial solution) or $\sin(k\pi) = 0$. Hence we get

$$k = 0, \pm 1, \pm 2, ... \ \Rightarrow \ \lambda = 0, \pm i, \pm 2i, ...$$

since $\lambda = \pm\sqrt{-k^2} = \pm ki$.

b) **Consider the finite difference discretization used in HW3 for the spatial derivatives. The eigenvectors of the discrete FD matrix are**

$$V^k = \begin{bmatrix} i\sin(kx_1) \\ i\sin(kx_2) \\ \vdots \\ i\sin(kx_{N-1}) \\ cos(kx_{1/2}) \\ cos(kx_{3/2}) \\ \vdots \\ cos(kx_{N-1/2}) \end{bmatrix}, \quad k = -(N-1), \ldots, -1, 0, 1, \ldots, N-1.$$

Find the analytical expression for the eigenvalues and check it with `eig(M)` in *Matlab*.

From the matrix M given and the k-th eigenvector above, we can compute MV^k,

$$MV^k = \begin{cases} -V^k_{j+N-1} + V^k_{j+n} & j \leq N-1 \\ V^k_1 & j = N \\ -V^k_{j-N} + V^k_{j-N+1} & N+1 \leq j < 2N-1 \\ -V^k_{N-1} & j = 2N-1 \end{cases} ,$$

where V^k represents the k-th eigenvector, and j is the column inside the vector. From here we can solve for the eigenvalues for the different values of j. Let $j \leq N-1$. Then,

$$\begin{aligned} MV^k &= \frac{1}{h}\left[-V^k_{j+N-1} + V^k_{j+n}\right] \\ &= -\cos(kx_{j-1/2}) + \cos(kx_{j+1/2}) \\ &= -\cos\left(k\left(j - \frac{1}{2}\right)h\right) + \cos\left(k\left(j + \frac{1}{2}\right)h\right) \\ &= -2\sin(kjh)\sin\left(k\frac{h}{2}\right), \end{aligned}$$

where we have used the identity $\cos(a+b) - \cos(a-b) = -2\sin(a)(b)$. Then, since we are looking for the eigenvalue that satisfies $MV^k = \lambda^k V^k$, we have

$$\begin{aligned} -2\sin(kjh)\sin\left(k\frac{h}{2}\right) &= ih\sin(kjh)\lambda_j \\ \Rightarrow \lambda^k_j &= \frac{2i}{h}\sin\left(k\frac{h}{2}\right) = \lambda^k. \end{aligned}$$

Note that the k-th eigenvalue does not depend on j. This is the desired result since the k-th eigenvalue is the same for the each entry of the k-th eigenvector.

Next let $j = N$. Then,

$$MV^k = \frac{1}{h}V^k_1 = \frac{i}{h}\sin(kh) ,$$

and

$$\lambda^k_N V^k = \lambda^k_N \cos(kx_{1/2}) = \lambda^k_N \cos\left(k\frac{h}{2}\right) ,$$

which implies

$$\begin{aligned} &\frac{i}{h}\sin(kh) = \lambda^k_N \cos\left(k\frac{h}{2}\right) \\ &\Rightarrow i\sin(kh) = \lambda^k_N h\cos\left(k\frac{h}{2}\right) \\ &\Rightarrow 2i\sin\left(k\frac{h}{2}\right)\cos\left(k\frac{h}{2}\right) = \lambda^k_N h\cos\left(k\frac{h}{2}\right) \\ &\Rightarrow \lambda^k_N = \frac{2i}{h}\sin\left(k\frac{h}{2}\right) = \lambda^k. \end{aligned}$$

Now, let $j \in [N+1, 2N-1)$, then

$$\begin{aligned} MV^k &= \frac{1}{h}\left[-V^k_{j-N} + V^k_{j-N+1}\right] \\ &= \frac{i}{h}\left[-\sin\left(kx_{j-N}\right) + \sin\left(kx_{j-N+1}\right)\right] \\ &= \frac{i}{h}\left[\sin(k(j-N+1)h) - \sin(k(j-N)h)\right] \ , \end{aligned}$$

and

$$\lambda^k_j V^k_j = \lambda^k_j \cos(kx_{j-N+1/2}) = \lambda^k_j \cos(k(j-N+1/2)h) \ .$$

This gives us

$$\begin{aligned} &\frac{i}{h}\left[\sin(k(j-N+1)h) - \sin(k(j-N)h)\right] = \lambda^k_j \cos(k(j-N+1/2)h) \\ &\Rightarrow \frac{2i}{h}\cos(k(j-N+1/2)h)\sin\left(k\frac{h}{2}\right) = \lambda^k_j \cos(k(j-N+1/2)h) \\ &\Rightarrow \lambda^k_j = \frac{2i}{h}\sin\left(k\frac{h}{2}\right) = \lambda^k. \end{aligned}$$

Finally, for the last entry of the vector, let $j = 2N-1$, then

$$\begin{aligned} MV^k &= -\frac{1}{h}V^k_{N-1} = -\frac{i}{h}\sin(kx_{N-1}) = -\frac{i}{h}\sin(k(N-1)h) = -\frac{i}{h}\sin(k\pi - kh) \\ &= -\frac{i}{h}\sin(k\pi)\cos(kh) + \frac{i}{h}\sin(kh)\cos(k\pi) = \frac{i}{h}\sin(kh)\cos(k\pi) \ , \end{aligned}$$

and

$$\begin{aligned} \lambda^k_{2N-1}V^k &= \lambda^k_{2N-1}\cos(k(N-1/2)h) = \lambda^k_{2N-1}\cos\left(knh - k\frac{h}{2}\right) = \lambda^k_{2N-1}\cos\left(k\pi - k\frac{h}{2}\right) \\ &= \lambda^k_{2N-1}\left[\cos(k\pi)\cos\left(k\frac{h}{2}\right) - \sin(k\pi)\sin\left(k\frac{h}{2}\right)\right] = \lambda^k_{2N-1}\cos(k\pi)\cos\left(k\frac{h}{2}\right) \ . \end{aligned}$$

Joining the two results we get the following equation,

$$\begin{aligned} &\lambda^k_{2N-1}\cos\left(k\pi - k\frac{h}{2}\right) = \frac{i}{h}\sin(kh)\cos(k\pi) \\ &\Rightarrow \lambda^k_{2N-1} = \frac{2i}{h}\sin\left(k\frac{h}{2}\right) = \lambda^k. \end{aligned}$$

Thus,

$$\boxed{\lambda^k = \frac{2i}{h}\sin\left(k\frac{h}{2}\right)} \ .$$

Matlab code for this part

```
%% Problem 2b
clear variables
close all

N=100;
h=pi/N;

Du = gallery(’tridiag’,N,-1,1,0)/h; % In sparse form.
Du(:,end)=[];
Dp = gallery(’tridiag’,N,0,-1,1)/h; % In sparse form.
Dp(end,:)=[];
Z1 = zeros(N-1,N-1);
Z2 = zeros(N,N);

M = [Z1 Dp ; Du Z2];

E=eig(full(M));
E=sort(E);
k=-(N-1):(N-1);
E_analytic=2*1i/h*sin(k*h/2)’;
E_analytic=sort(E_analytic);

difference=norm(E-E_analytic,inf)
% The expression in part b is correct
```

c) **Show that the eigenvalues of M are $\mathcal{O}(h^2)$.**

Proof. We can prove the state ment by simply Taylor expanding the previous expression for the eigenvalues :

$$\begin{aligned}\lambda_k(h) &= \lambda_k(0) + h\lambda_k'(0) + \frac{h^2}{2}\lambda_k''(0) + \frac{h^3}{6}\lambda_k'''(0) + \cdots \\ &= \frac{ik}{h}h - \frac{h^3}{6}i\frac{k^3}{4h} + \mathcal{O}(h^4) \\ &= ik\left[1 + \frac{k^2h^2}{6} + \mathcal{O}(h^4)\right].\end{aligned}$$

Note that we have dropped the k for convenience. Thus,

$$\lambda_k(h) = ik\left[1 + \mathcal{O}((kh)^2)\right].$$

d) **Show that**

$$\lambda_{\pm(N-1)} = \pm\frac{2i}{h}\cos(h/2), \quad \text{and} \quad |\lambda_{\pm(N-1)}| < \frac{2}{h} = \frac{2}{\pi}N\ .$$

By simply plugging ±$(N-1)$ into the eigenvalue equation,

$$\begin{aligned}\lambda_{\pm(N-1)} &= \frac{2i}{h}\sin\left(\pm N\frac{h}{2} \mp \frac{h}{2}\right) = \frac{2i}{h}\sin\left(\pm\frac{\pi}{2} \mp \frac{h}{2}\right)\\ &= \frac{2i}{h}\left[\sin\left(\pm\frac{\pi}{2}\right)\cos\left(\mp\frac{h}{2}\right) + \cos\left(\pm\frac{\pi}{2}\right)\sin\left(\mp\frac{h}{2}\right)\right]\\ &= \pm\frac{2i}{h}\cos\left(\frac{h}{2}\right)\ .\end{aligned}$$

Using this, we note that $0 \leq |\cos\left(\frac{\pi}{2N}\right)| < 1$ and $|i| = 1$, so,

$$|\lambda_{\pm(N-1)}| = \left|\pm\frac{2i}{h}\cos\left(\frac{h}{2}\right)\right| < \frac{2}{h} = \frac{2N}{\pi}\ .$$

e) **Use RK4 to time-step this PDE. Plot the stability region of RK4 and find a stable Δt. Find the growth function g for this method and plot $|g|$ using contour.**

To obtain the growth function $g(z)$ let $z = \Delta t\lambda$ and $f(t, y) = \lambda y$. Then,

$$\begin{aligned}k_1 &= zy^n\\ k_2 &= \left(z + \frac{z^2}{2}\right)y^n\\ k_3 &= \left(z + \frac{1}{2}\left(z^2 + \frac{z^3}{2}\right)\right)y^n\\ k_4 &= \left(z + z\left(z + \frac{1}{2}\left(z^2 + \frac{z^3}{2}\right)\right)\right)y^n\ ,\end{aligned}$$

and

$$\begin{aligned}y^{n+1} &= y^n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)y^n\\ &= \left(1 + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)\right)y^n\\ &= \left[1 + \frac{1}{6}\left(6z + 3z^2 + z^3 + \frac{z^4}{4}\right)\right]y^n.\end{aligned}$$

Hence,

$$g(z) = 1 + \frac{1}{6}\left(6z + 3z^2 + z^3 + \frac{z^4}{4}\right).$$

We can plot $g(z)$ in *Matlab* to obtain the stability region, shown in part f).

f) **Assuming that the boundary of the stability region crosses the imaginary axis approximately at $\pm 2.81i$, estimate the stability requirement on Δt (use part (4) to find c so that $\Delta t < ch$ for stability). For $N = 100$, plot the eigenvalues of M multiplied by Δt together with the stability region and verify that they fall inside the stability region.**

For stability we know that

$$|\Delta t \lambda_k| < 2.81,$$

for all the eigenvalues λ_k. Since the largest eigenvalue is going to be the one from part d),

$$\frac{2.81}{|\lambda_k|} > \frac{2.81}{|\lambda_{\pm(N-1)}|} > \frac{2.81}{2}h > \Delta t.$$

Thus,

$$\Delta t < 1.405h.$$

The figure below shows that $1.405h\lambda_k$ falls within the stability region for all λ_k.

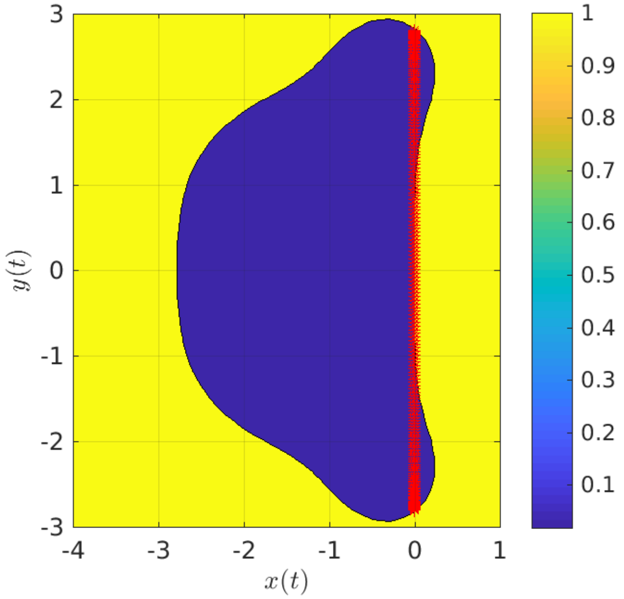


Figure 3: Stability region and eigenvalues.

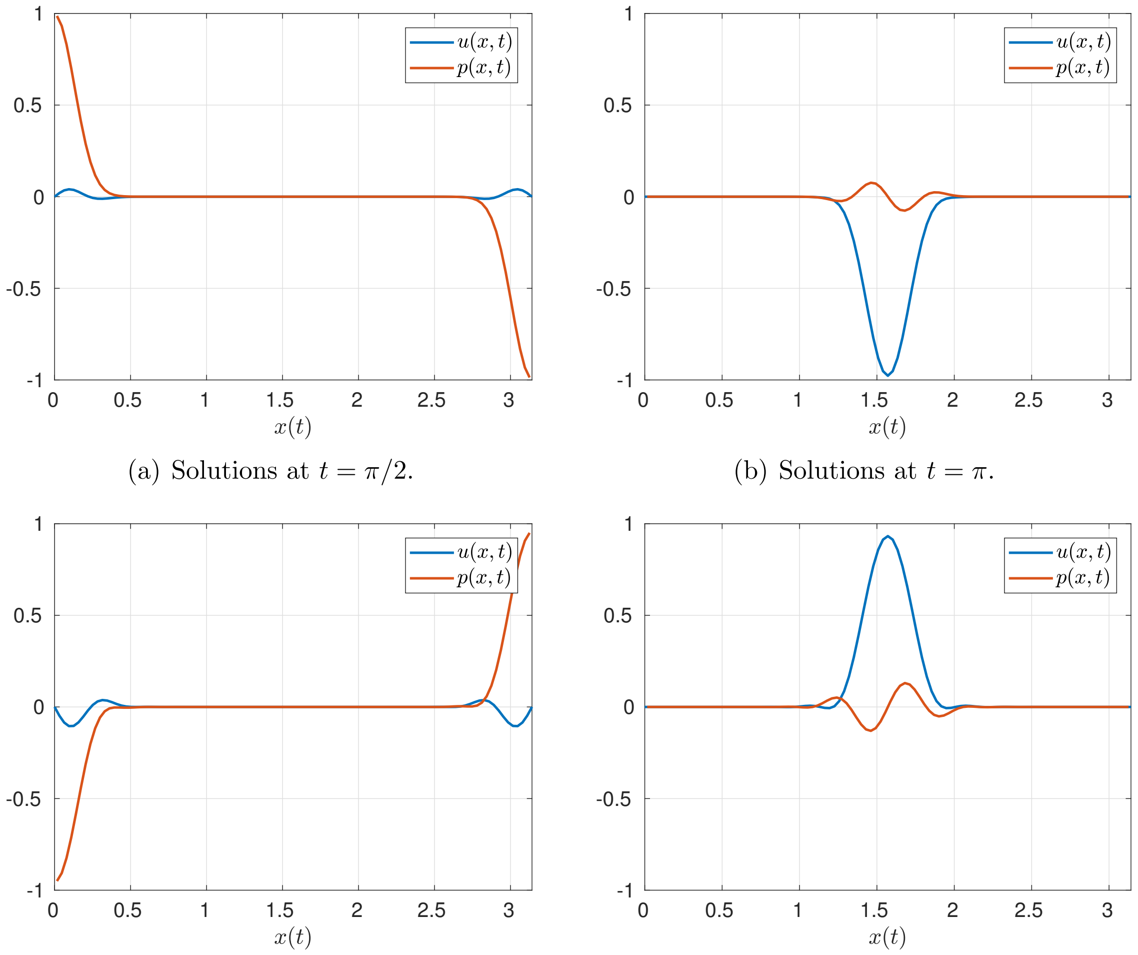
Matlab code for this part

```
a=-20;
b=20;
c=-20;
```

```
d=20;
dx=0.1;
dy=dx;

[zr,zi]=meshgrid(a:dx:b,c:dy:d);
z=zr+1i*zi;
g=1+1/6*(6*z+3*z.^2+z.^3+z.^4/4);
dt=1.405*h;
lambda=dt*E;

figure
contourf(zr,zi,abs(g),[-inf 1 inf], 'k'), colorbar
hold on
grid on
plot(real(lambda),imag(lambda),'r*')
xlabel('$x(t)$','interpreter','latex','fontsize',16)
ylabel('$y(t)$','interpreter','latex','fontsize',16)
axis('image', [-4 1 -3 3])
set(gca,'fontsize',12)
txt='Latex/FIGURES/P2_ef';
saveas(gcf,txt,'epsc')
```

g) **With $N = 100$, solve the PDE using** `RK4` **and plot your solution at times $t = \pi/2,\ \pi, 3,\ \pi/2,\ 2\pi$.**

We can see the solutions at those values of time in the next figure.



(a) Solutions at $t = \pi/2$.

(b) Solutions at $t = \pi$.

(c) Solutions at $t = 3\pi/2$.

(d) Solutions at $t = 2\pi$.

Figure 4: Solutions of the acoustic equation.

Matlab code for this part

```
%% Problem 2g
clear variables
close all
format long
clc

linewidth=1.7;
legendfontsize=14;
axisfontsize=14;

N=100;
L=pi;
h=L/N;
```

```
xu=linspace(0,pi,N+1);
xp=linspace(0+h/2,pi-h/2,N);

Du = gallery(’tridiag’,N,-1,1,0); % In sparse form.
Du(:,end)=[];
Du=Du/h;
Dp = gallery(’tridiag’,N,0,-1,1); % In sparse form.
Dp(end,:)=[];
Dp=Dp/h;
Z1 = zeros(N-1,N-1);
Z2 = zeros(N,N);
M = [Z1 Dp ; Du Z2];

u0 = exp(-30*(xu-pi/2).^2)’;
p0 = zeros(N,1);
V0 = [u0(2:end-1) ; p0];

t = 0:pi/200:2*pi;
dydt = @(t,V) M*V;
[t,V] = rk4(dydt,[0 2*pi],V0,length(t)-1);
% Plots
figure(1)
plot(xu,[0 V(101,1:N-1) 0],’linewidth’,linewidth)
hold on
plot(xp,V(101,N:2*N-1),’linewidth’,linewidth)
grid on
axis([0 pi -1 1])
xlabel(’$x(t)$’,’interpreter’,’latex’)
legend({’ $u(x,t)$’,’ $p(x,t)$’},...
 ’Interpreter’,’latex’,’fontsize’,legendfontsize)
set(gca,’fontsize’,axisfontsize)
txt=’Latex/FIGURES/P2_g1’;
saveas(gcf,txt,’epsc’)


figure(2)
plot(xu,[0 V(201,1:N-1) 0],’linewidth’,linewidth)
hold on
plot(xp,V(201,N:2*N-1),’linewidth’,linewidth)
grid on
axis([0 pi -1 1])
xlabel(’$x(t)$’,’interpreter’,’latex’)
legend({’ $u(x,t)$’,’ $p(x,t)$’},...
 ’Interpreter’,’latex’,’fontsize’,legendfontsize)
set(gca,’fontsize’,axisfontsize)
```

```
txt='Latex/FIGURES/P2_g2';
saveas(gcf,txt,'epsc')

figure(3)
plot(xu,[0 V(301,1:N-1) 0],'linewidth',linewidth)
hold on
plot(xp,V(301,N:2*N-1),'linewidth',linewidth)
grid on
axis([0 pi -1 1])
xlabel('$x(t)$','interpreter','latex')
legend({' $u(x,t)$',' $p(x,t)$'},...
 'Interpreter','latex','fontsize',legendfontsize)
set(gca,'fontsize',axisfontsize)
txt='Latex/FIGURES/P2_g3';
saveas(gcf,txt,'epsc')

figure(4)
plot(xu,[0 V(401,1:N-1) 0],'linewidth',linewidth)
hold on
plot(xp,V(401,N:2*N-1),'linewidth',linewidth)
grid on
axis([0 pi -1 1])
xlabel('$x(t)$','interpreter','latex')
legend({' $u(x,t)$',' $p(x,t)$'},...
 'Interpreter','latex','fontsize',legendfontsize)
set(gca,'fontsize',axisfontsize)
txt='Latex/FIGURES/P2_g4';
saveas(gcf,txt,'epsc')
```

h) **With $N = 100$, run you code with Δt that is slightly bigger than the optimal (stable) choice, is the computation stable?**

With a choice of $\Delta t = 1.471h$, the solution is unstable as we can see in the next figure.

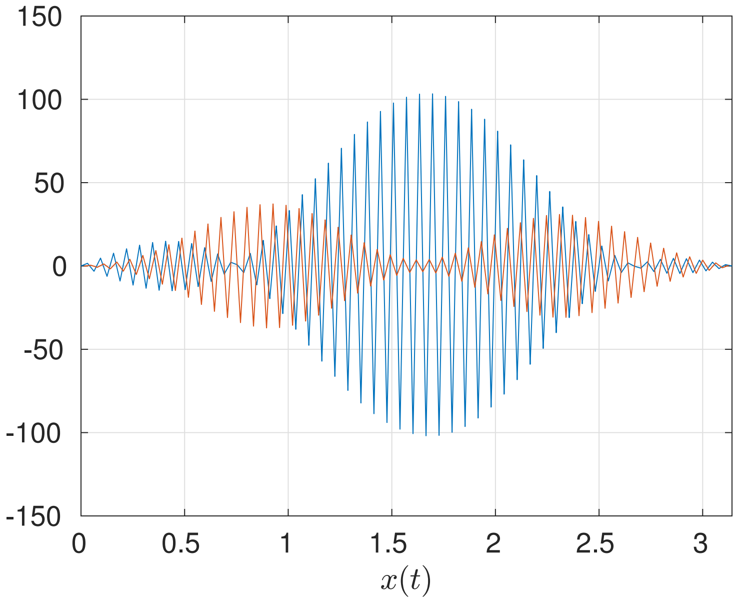


Figure 5: Unstable solution.

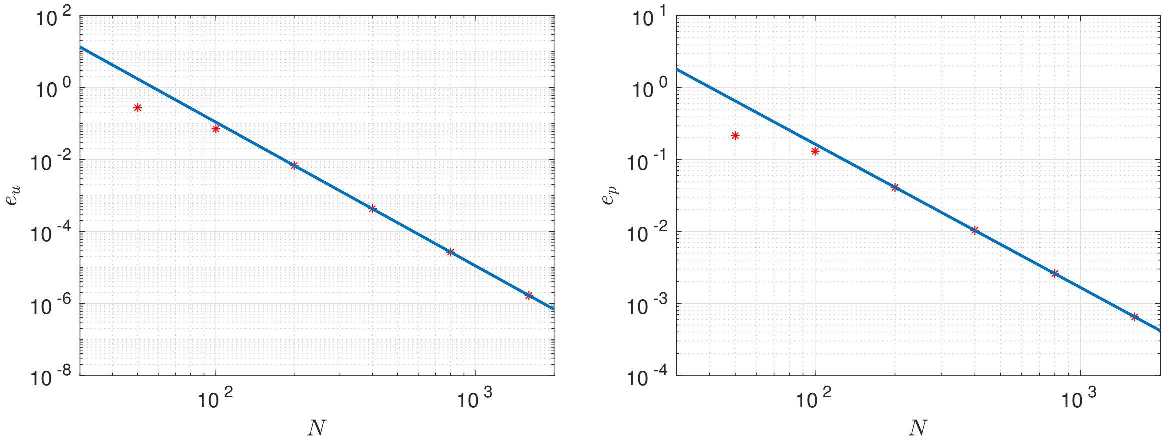
Matlab code for this part

```
%% Problem 2h
t = 0:1.48*h:2*pi;
dydt = @(t,V) M*V;
[t,V] = rk4(dydt,[0 2*pi],V0,length(t)-1);
figure
plot(xu,[0 V(end,1:N-1) 0])
hold on
plot(xp,V(end,N:2*N-1))
grid on
% axis([0 pi -1 1])
xlim([0 pi])
xlabel('$x(t)$','interpreter','latex')
set(gca,'fontsize',axisfontsize)
txt='Latex/FIGURES/P2_h';
saveas(gcf,txt,'epsc')
```

i) **Using the stability criteria for choosing Δt, plot the error at $t = 2\pi$ for several values of N and verify that the error decays as $\mathcal{O}(h^2)$. Conclude that the error is dominated by the spatial discretization in this case.**

The spatial discretization dominates the error since the method used is second order in space and fourth order in time, which means the spatial discretization leads the error. However, we can see in the following figure that the observed order of convergence is not the theoretical one. We have obtained a observed order of convergence of 3.998 for

u and 1.993 for p using the infinity norm for the errors of both. For the interpolation we have not used the first two data points.



(a) Error of u with N.

(b) Error of p with N.

Figure 6: Order of convergence.

Matlab code for this part

```
%% Problem 2i
clear variables
close all
format long
clc
%Calculate exact solution with a large number of N
axisfontsize=14;
L=pi;

% Check order of convergence
Nvector=[50 100 200 400 800 1600];
for i=1:length(Nvector)
    N=Nvector(i)
    h=L/N;
    xu=linspace(0,pi,N+1);
    xp=linspace(0+h/2,pi-h/2,N);

    Du = gallery('tridiag',N,-1,1,0); % In sparse form.
    Du(:,end)=[];
    Du=Du/h;
    Dp = gallery('tridiag',N,0,-1,1); % In sparse form.
    Dp(end,:)=[];
    Dp=Dp/h;
    Z1 = zeros(N-1,N-1);
    Z2 = zeros(N,N);
    M = [Z1 Dp ; Du Z2];

    u0 = exp(-30*(xu-pi/2).^2)’;
    p0 = zeros(N,1);
    V0 = [u0(2:end-1) ; p0];
    dt=h;
    t = 0:dt:2*pi;
    dydt = @(t,V) M*V;
    [t,V] = rk4(dydt,[0 2*pi],V0,length(t)-1);

    difNormu(i)=norm(V(end,1:N-1)’-u0(2:end-1),inf);
    index=find(xp>=pi/2,1);
    difNormp(i)=norm(V(end,N:2*N-1)’-p0,inf);
end
% Order of convergence of u and p
hvector=pi/1600:1e-5:pi/50;
NvectorCont=20:500:7e3;

% Fitting curves for the error.
% Ignore first 2 points. No asymptotic regime.

cu=polyfit(log(Nvector(3:end)),log(difNormu(3:end)),1);
ufit=exp(cu(2))*NvectorCont.^cu(1);
cp=polyfit(log(Nvector(3:end)),log(difNormp(3:end)),1);
pfit=exp(cp(2))*NvectorCont.^cp(1);
cu(1)
cp(1)

figure
loglog(Nvector,difNormu,’r*’)
hold on
loglog(NvectorCont,ufit,’linewidth’,2)
grid on
xlim([30 2000])
set(gca,’fontsize’,axisfontsize)
xlabel(’$N$’,’interpreter’,’latex’,’fontsize’,16)
ylabel(’$e_u$’,’interpreter’,’latex’,’fontsize’,16)
txt=’Latex/FIGURES/P2_iu’;
saveas(gcf,txt,’epsc’)

figure
loglog(Nvector,difNormp,’r*’)
hold on
```

```
loglog(NvectorCont,pfit,'linewidth',2)
grid on
xlim([30 2000])
set(gca,'fontsize',axisfontsize)
xlabel('$N$','interpreter','latex','fontsize',16)
ylabel('$e_p$','interpreter','latex','fontsize',16)
txt='Latex/FIGURES/P2_ip';
saveas(gcf,txt,'epsc')
```

j) **Repeat part g), but use 3rd order Adams-Bashforth for time-stepping. Use `RK4` to compute the first 2 time levels. Use a Δt that is close to the stability limit for `AB3`.** First we calculate the stability region for 3rd order Adams-Bashforth, which has the following form,

$$y^{n+1} = y^n = \frac{\Delta t}{12}(23f^n - 16f^{n-1} + 5f^{n-2}).$$

We begin with the substitution $f^n = \lambda y^n$:

$$y^{n+1} = y^n = \frac{\lambda\Delta t}{12}(23y^n - 16y^{n-1} + 5y^{n-2}).$$

Note that $y^n = gy^{n-1}$ and let $z = \lambda\Delta t$,

$$gy^n = y^n + \frac{z}{12}\left(23y^n - 16\frac{y^n}{g} + 5\frac{y^n}{g^2}\right).$$

Factoring out y^n we get

$$g^3 = g^2 + \frac{z}{12}(23g^2 - 16g + 5),$$

and solving for z,

$$z = 12\left(\frac{g^3 - g^2}{23g^2 - 16g + 5}\right).$$

Using Matlab, we find the stability region crosses the imaginary axis just shy of ±0.72. Thus we calculate a stable time step to be

$$\Delta t \le 0.36h.$$

We will choose in the code $\Delta t = 0.36h$. In the next figure we can see how all the eigenvalues are inside the stability region for the time step chosen.

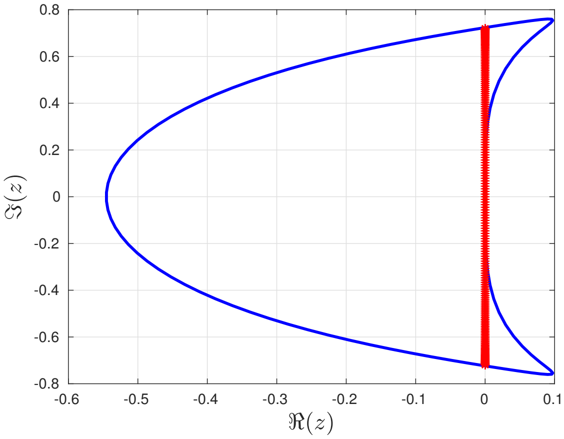
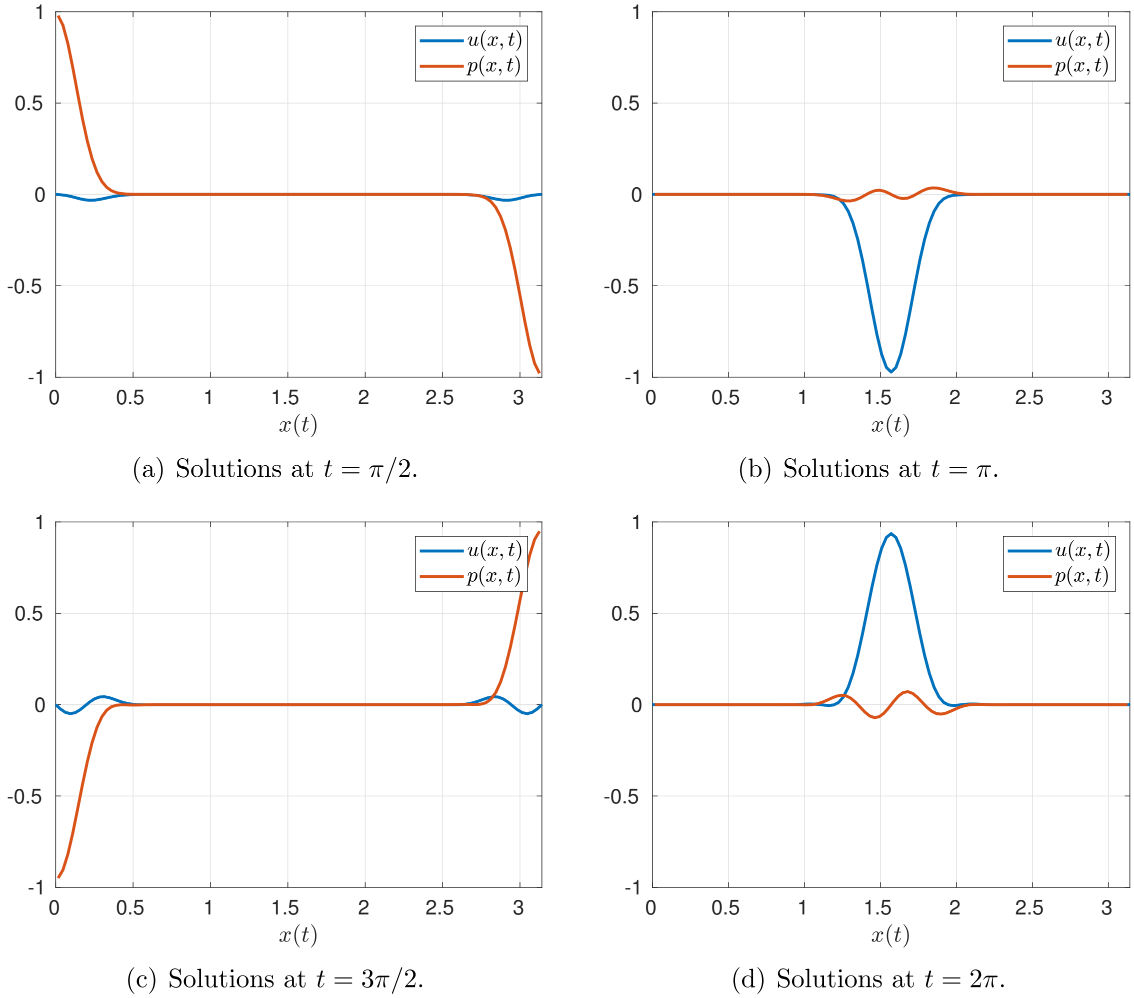


Figure 7: Stability region and eigenvalues.

Finally, in the next figure we can see that we obtained the same solution as in part g).



(a) Solutions at $t = \pi/2$.

(b) Solutions at $t = \pi$.

(c) Solutions at $t = 3\pi/2$.

(d) Solutions at $t = 2\pi$.

Figure 8: Solutions of the acoustic equation.

Matlab code for this part

```
%% Problem 2j
clear variables
close all
clc

legendfontsize=14;
axisfontsize=14;

L=pi;
N=100;
h=L/N;

theta=linspace(0,2*pi,N);
g=exp(1i*theta);
z=12*(g.^3-g.^2)./(23*g.^2-16*g+5);

xu=linspace(0,pi,N+1);
xp=linspace(0+h/2,pi-h/2,N);

Du = gallery('tridiag',N,-1,1,0); % In sparse form.
Du(:,end)=[];
Du=Du/h;
Dp = gallery('tridiag',N,0,-1,1); % In sparse form.
Dp(end,:)=[];
Dp=Dp/h;
Z1 = zeros(N-1,N-1);
Z2 = zeros(N,N);
M = [Z1 Dp ; Du Z2];

e=eig(full(M));
dt=0.36*h;
lambdadt=dt*e;

%Plot eigenvalues in stability region
figure
plot(z,'b','linewidth',2)
hold on
plot(real(lambdadt),imag(lambdadt),'r*')
grid on
xlabel('$\Re(z)$','interpreter','latex'...
    ,'fontsize',16)
ylabel('$\Im(z)$','interpreter','latex'...
    ,'fontsize',16)
```

```
txt=’Latex/FIGURES/P2_jRegion’;
saveas(gcf,txt,’epsc’)


u0 = exp(-30*(xu-pi/2).^2)’;
p0 = zeros(N,1);
V0 = [u0(2:end-1) ; p0];


F2=@(t,v) M*v;

[t,V]=rk4(F2,[0 2*dt],V0,2);

F1=@(v) M*v;
f0=F1(V(1,:)’);
f1=F1(V(2,:)’);
f=F1(V(3,:)’);

V=V(3,:)’+dt/12*(23*f-16*f1+5*f0);

t=t(end);
outputTime=[pi/2 pi 3*pi/2 2*pi];
n=1;
while t<2*pi
    if (t < outputTime(n) && t+dt >= outputTime(n))
        dt=outputTime(n)-t;
    else
        dt=0.36*h;
    end
    f=F1(V);
    V=V+dt/12*(23*f-16*f1+5*f0);
    %Advance variables
    f0=f1;
    f1=f;
    t=t+dt;
    if (t==outputTime(n))
        figure
        plot(xu’,[0; V(1:N-1); 0],’linewidth’,2)
        hold on
        plot(xp’,V(N:2*N-1),’linewidth’,2)
        grid on
        legend({’ $u(x,t)$’,’ $p(x,t)$’},...
        ’Interpreter’,’latex’,’fontsize’,legendfontsize)
        set(gca,’fontsize’,axisfontsize)
        xlabel(’$x(t)$’,’interpreter’,’latex’,’fontsize’,16)
        axis([0 pi -1 1])
        txt=['Latex/FIGURES/P2_j' num2str(n)];
        saveas(gcf,txt,'epsc')
        n=n+1;
    end
end
```

Problem 3

Consider the predictor-corrector time-stepping scheme:

$$\begin{aligned} y^p_{n+1} &= y_n + \frac{\Delta t}{12}(23f_n - 16f_{n-1} + 5f_{n-2}), \\ y_{n+1} &= y_n + \frac{\Delta t}{12}(5f(t_{n+1}, y^p_{n+1}) + 8f)n - f_{n-1}) \ . \end{aligned}$$

where $f_n = f(t_n, y_n)$ and $y'(t) = f(t, y)$, $y(0) = y_0$. Plot the stability region for this scheme.

Let $f_n = \lambda y^n$ and $a = \frac{\Delta t \lambda}{12}$. We start by simply substituting y^p_{n+1} from the first equation into the second,

$$y^{n+1} = y^n + a\left[5(y^n + a(23y^n - 16y^{n-1} + 5y^{n-1})) + 8y^n - y^{n-1}\right] \ .$$

Further let $y^{n+1} = gy^n$,

$$gy^n = y^n + a\left[5y^n + 115ay^n - 80\frac{a}{g}y^n + 25\frac{a}{g^2}y^n\right] + 8ay^n - \frac{a}{g}y^n.$$

Reorganizing the terms, we obtain a second degree polynomial for a,

$$(115g^2 - 80g + 25)a^2 + (13g^2 - g)a + (g^2 - g^3) = 0,$$

which gives us the solution

$$a = \frac{-(13g^2 - g) \pm \sqrt{(13g^2 - g)^2 - 4(115g^2 - 80g + 25)(g^2 - g^3)}}{2(115g^2 - 80g + 25)}.$$

Note that $z = \Delta t \lambda = 12a$,

$$z = 12\frac{-(13g^2 - g) \pm \sqrt{(13g^2 - g)^2 - 4(115g^2 - 80g + 25)(g^2 - g^3)}}{2(115g^2 - 80g + 25)}.$$

We plot the previos in *Matlab* and obatin stability region shown in the next figure.

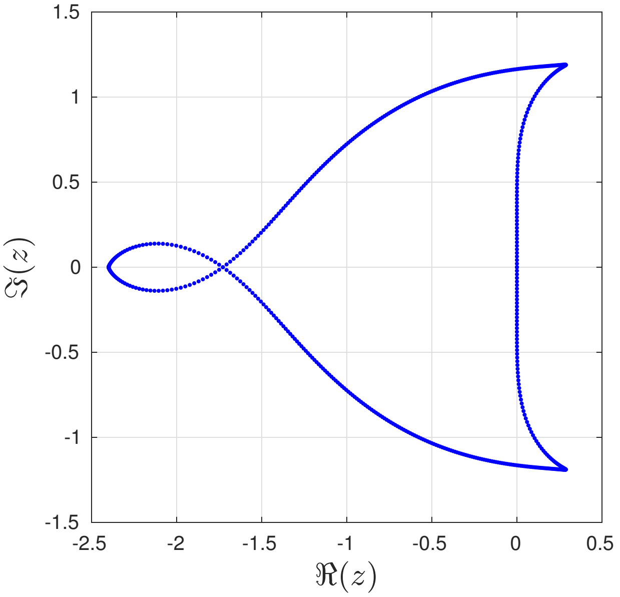


Figure 9: Stability region.

Matlab code for this problem

```
%% Problem 3
clear variables
close all
clc
N=300;
theta = linspace(0,2*pi,N);
g=exp(1i*theta);
a=115*g.^2-80*g+25;
b=13*g.^2-g;
c=g.^2-g.^3;
sq=sqrt(b.^2-4*a.*c);
zp=12*(-b+sq)./(2*a);
zn=12*(-b-sq)./(2*a);
figure
plot(real(zp),imag(zp),'b.')
hold on
plot(real(zn),imag(zn),'b.')
grid on
axis('image', [-2.5 0.5 -1.5 1.5])
xlabel('$\Re(z)$','interpreter','latex'...
    ,'fontsize',16)
ylabel('$\Im(z)$','interpreter','latex'...
    ,'fontsize',16)
txt='Latex/FIGURES/P3';
```

```
saveas(gcf,txt,'epsc')
```