San Diego State University

Department of Mathematics and Statistics
Math 693b
Advanced Numerical Analysis

Codes for Final Project

Matteo Polimeno

Professor:
Dr. Peter Blomgren

May 1^{st}, 2018

Contents

1 KdV

1.1 Code

```
%RK4 with integrating factor
%set up grid and soliton initial data
clear all;
clc;
N=2^8; dt=.4/N^2; x=(2*pi/N)*(-N/2:N/2-1)';
A=25; B=16; C=36; clf,drawnow
u=3*A^2*sech(.5*(A*(x+2))).^2+3*B^2*sech(.5*(B*(x+1))).^2; %KdV with sup
%u=3*B^2*sech(.5*(B*(x+1))).^2; %single soliton %uncomment when necessary

%fourier transform of initial conditions
w=fft(u);
%k-vector
k=[0:N/2-1 0 -N/2+1:-1]';
%linear opeator
ik3=1i*k.^3;

%solve PDE and plot
tmax=0.006;
nplt=floor((tmax/25)/dt);
nmax=round(tmax/dt);
udata=u;tdata=0;
tic

for n=1:nmax
    t=n*dt;g=-.5i*dt*k;%kdv
    E=exp(dt*ik3/2); E2=E.^2;
    a=g.*fft(real(ifft(w)).^2);
    b=g.*fft(real(ifft(E.*(w+a/2))).^2);
    c=g.*fft(real(ifft(E.*w+b/2)).^2);
    d=g.*fft(real(ifft(E2.*w+E.*c)).^2);
    w=E2.*w+(E2.*a+2*E.*(b+c)+d)/6;
     if mod(n,nplt)==0
          u=real(ifft(w));
          udata=[udata u]; tdata=[tdata t];
     end
end
```

```
toc %print elapsed time
nmax %print # of steps involved
waterfall(x,tdata,udata'), view(-20, 25),rotate3d on
xlabel x, ylabel t, axis([-pi pi 0 tmax 0 2000]), grid off
title('Solution to KdV with 1-soliton initial data')
```

1.2 KdV-Movie

```
%RK4 with integrating factor
%set up grid and 2-soliton initial data
clear all;
clc;
N=2^8; dt=.4/N^2; x=(2*pi/N)*(-N/2:N/2-1)';
A=25; B=16;clf,drawnow
u=3*A^2*sech(.5*(A*(x+2))).^2+3*B^2*sech(.5*(B*(x+1))).^2; %KdV with sup
%u=cos(pi*x);
w=fft(u); k=[0:N/2-1 0 -N/2+1:-1]';
ik3=1i*k.^3;
%solve PDE and plot
tmax=0.004; %kdv and burgers
nplt=floor((tmax/25)/dt);
nmax=round(tmax/dt);
%udata=u;tdata=0;%h=waitbar(0,'please wait...');

v = VideoWriter('Kdv_moviewaterfall.avi');
open(v)
for n = 1:nmax
    t=n*dt;g=-.5i*dt*k;%kdv
    E=exp(dt*ik3/2); E2=E.^2;
    a=g.*fft(real(ifft(w)).^2);
    b=g.*fft(real(ifft(E.*(w+a/2))).^2);
    c=g.*fft(real(ifft(E.*w+b/2)).^2);
    d=g.*fft(real(ifft(E2.*w+E.*c)).^2);
    w=E2.*w+(E2.*a+2*E.*(b+c)+d)/6;
    u=real(ifft(w));
    plot(x,u,'linewidth',2.5)
    axis([-pi pi 0 2000])
    M(n) = getframe;
    %%this movie is cooler but was not able to save the file
    %udata=[udata u]; tdata=[tdata t];
    %pcolor(x,tdata,udata')
    %shading flat; colormap('jet');
    %title(['KdV-Solitons superposition, t=' num2str(tdata(n))])
    %writeVideo(v,M(n));
end
close(v)
```

2 KPII

2.1 Runge-Kutta Time stepper

```
function J = rk4exp2(w,dt,g,E,KT)
E2=E.^2;
a=g.*reshape((fft2(real(ifft2(reshape(w.',KT,KT).')).^2))',KT^2,1); %you
b=g.*reshape((fft2(real(ifft2(reshape((E.*(w+a/2)).',KT,KT).')).^2))',KT^
c=g.*reshape((fft2(real(ifft2(reshape((E.*w+b/2).',KT,KT).')).^2))',KT^2
d=g.*reshape((fft2(real(ifft2(reshape((E2.*w+E.*c).',KT,KT).')).^2))',KT^

J=E2.*w+(E2.*a+2*E.*(b+c)+d)/6;
end
```

2.2 Code Implementation

```
%KP2
clear all;
clc;
%set up grid and initial data
N=2^6;%modes
KT=N*2;
L=10;
dt=0.01;
Xmesh=linspace(-L,L,KT);
[Xxmesh,Yymesh]=meshgrid(Xmesh,Xmesh);
Xxxmesh=pi./(L*Xxmesh);
Yyymesh=pi./(L*Yymesh);
u0 = sin(Xxxmesh)*cos(Yyymesh).^3;
clf,drawnow

%kronecker tensor products
Dds = 1i.*pi/L*[0:KT/2 -KT/2+1:-1]';
Dy=kron(Dds,ones(KT,1)); %ones(KT) is not the same as np.ones(KT).
Dx=kron(ones(KT,1),Dds); %ones(KT) creates a KTxKT array in matlab. and a
Dx3=Dx.^3;
Dy2=6.*Dy.^2;

Dds1 = length(Dds(2:end));
b = ones(Dds1,1)./Dds(2:end);
```

```
Dxn1sb = [0; b];
Dxn1=kron(ones(KT,1),Dxn1sb);
Dx=(3/2).*Dx;

Lop=Dx3+(Dy2.*Dxn1);%linear operator
g=-.5i.*Dx.*dt;%nonlinearity
E=exp(dt.*Lop./2);%integrating factor (exponential)
%solve PDE and plot
tmax = 10; nmax = round(tmax/dt);

nmax=round(tmax/dt);

Xxxmesh=pi./(L*Xxmesh); %variable change
Yyymesh=pi./(L*Yymesh);

u0 = sin(Xxxmesh)*cos(Yyymesh).^3; %IC's
%reshape and fourier transform initial data
w = reshape(fft2(u0)',KT^2,1);

%call runge kutta
for n=1:nmax
    t=n*dt;
    w = rk4exp2(w,dt,g,E,KT);
    wnp1 = (real(ifft2(reshape(w.',KT,KT).')))';
end

pcolor(Xxmesh,Yymesh,wnp1), rotate3d on
colorbar
xlabel x, ylabel y
```

3 Kuramoto-Sivashinsky

```
%Kuramoto-Sivashinky equation
%it evolves into chaos
clear all;
clc;
%set up grid and initial data
N = 2^8;
x=(32*pi/N)*(-N/2:N/2-1)';%periodic grid
%u0 = cos(x/16).*(1+sin(x/16)); %paper IC
u0 = exp(-x.^2); %trefethen IC
w = fft(u0);
clf,drawnow

dt=1/40;
k = [0:N/2-1 0 -N/2+1:-1]'/16;
k2 = k.^2;%kuramoto
k4 = k.^4; %kuramoto
Lop = (k2-k4);%kuramoto LOP obtained via FFT

%solve PDE and plot
uu = u0; tt = 0;
tmax = 250; nmax = round(tmax/dt); nplt = floor((tmax/375)/dt);

nmax=round(tmax/dt);
h=waitbar(0,'please wait...');
for n=1:nmax
    t=n*dt;
    g=-.5i*k*dt;%kuramoto
    E=exp(dt*Lop/2); E2=E.^2;
    a=g.*fft(real(ifft(w)).^2);
    b=g.*fft(real(ifft(E.*(w+a/2))).^2);
    c=g.*fft(real(ifft(E.*w+b/2)).^2);
    d=g.*fft(real(ifft(E2.*w+E.*c)).^2);
    w=E2.*w+(E2.*a+2*E.*(b+c)+d)/6;
      if mod(n,nplt)==0
          u0 = real(ifft(w));
          uu = [uu,u0]; tt = [tt,t];
      end
```

end

```
surf(tt,x,uu), shading interp, lighting phong, axis tight
view([-90 90]), colormap(winter); set(gca,'zlim',[-5 50])
light('color',[0 1 1],'position',[1,1,1]) %cyan[0 1 1]
material shiny
xlabel t, ylabel x
title('Kuramoto-Sivashinky Equation for $u_{0}=e^{-x^{2}}$','interpreter'
```