Applied Numerical Methods : Project 8

GOEYNS Florentin & WEICKER David

18th December 2015

Matlab codes

```
function [U,r,t] = highCyl(Nx,Nt,tend)
%HIGHCYL Solves the problem in project 8 where the cylinder is considered
%to have infinite height.
%INPUT : - Nx is the number of spatial unknown ==> (Nx+1)*h = R
%        - Nt is the number of time steps
%        - tend is the final time ==> deltaT*Nt = tend
%OUTPUT : - U is a matrix containing the solution
%           U(:,i) is the solution for time = t(i)
%         - r is a row vector containing the spatial discretization
%         - t is a row vector containing the time discretization
%
%Goyens Florentin & Weicker David

nu = 1e-6;
omega = 1;
R = 0.04;
h = R/(Nx+1);
alpha = nu/(h*h);

e = (1:Nx)';
a = 1-1./(2*(e+1));
b = -2-1./(e.*e);
c = 1+1./(2*(e-1));
A = alpha*spdiags([a b c],-1:1,Nx,Nx);

U = zeros(Nt+1,Nx+2);
r = 0:h:R;
U(1,:) = omega*r;

deltaT = tend/Nt;
t = 0:deltaT:tend;

%First iteration takes initial condition into account
f = [zeros(1,Nx-1) nu*R*omega*(1/(h*h)+1/(2*h*h*Nx))]';
U(2,2:Nx+1) = (eye(Nx)-0.5*deltaT*A)\((eye(Nx)+0.5*deltaT*A)*U(1,2:Nx+1)'+0.5*deltaT*f↩
    );

%Following iterations
for i=2:Nt
    U(i+1,2:Nx+1) = (eye(Nx)-0.5*deltaT*A)\((eye(Nx)+0.5*deltaT*A)*U(i,2:Nx+1)');
end
end
```

```
function [U,r,z,t] = nsCyl(Nr,Nt,tend)
%NSCYL Solves the problem in project 8 for height H = 0.08 m
%INPUT : - Nr is the number of spatial unknown ==> (Nr+1)*h = R
%        - Nt is the number of time steps
%        - tend is the final time ==> deltaT*Nt = tend
%OUTPUT : - U is a matrix containing the solution
%           U(:,:,i) is the solution for time = t(i)
%          - r is a row vector containing the spatial discretization in r
%          - z is a row vector containing the spatial discretization in z
%          - t is a row vector containing the time discretization
%
%Goyens Florentin & Weicker David
nu = 1e-6;
omega = 1;
R = 0.04;
H = 0.08;

h = R/(Nr+1);
r = 0:h:R;
z = 0:h:H;
Nz = length(z)-2;
alpha = nu/(h*h);

e = zeros(Nr*Nz,1);
for i=0:Nz-1
    e(1+i*Nr:(i+1)*Nr) = (1:Nr)';
end

%Construction of A and B
a = ones(Nr*Nz+1,1);
b = [1-1./(2*e(2:end)); 0; 0];
c = [-4-1./(e.*e); 0];
d = [0; 1+1./(2*e)];
A = alpha*spdiags([a b c d a],[-Nr -1 0 1 Nr],Nr*Nz,Nr*Nz);
B = zeros(Nr*Nz,1);
for m = 1:Nz-1
    A(Nr*m,Nr*m+1) = 0;
    A(1+m*Nr,m*Nr) = 0;
    B(m*Nr) = alpha*(1+1/(2*Nr))*R*omega;
end
B(Nz*Nr) = B(Nz*Nr) + alpha*(1+1/(2*Nr))*R*omega;
for m = 1:Nr
    B(m) = B(m) + nu*omega*m/h;
    B(m+(Nz-1)*Nr) = B(m+(Nz-1)*Nr) + nu*omega*m/h;
end

U = zeros(Nz+2,Nr+2,Nt+1);
U(:,:,1) = omega*ones(Nz+2,1)*r;
u = zeros(Nr*Nz,Nt+1);
u(:,1) = reshape(U(2:end-1,2:end-1,1)',Nr*Nz,1);

deltaT = tend/Nt;
t = 0:deltaT:tend;
%First iteration
I = eye(Nr*Nz);
u(:,2) = (I-0.5*deltaT*A)\((I+0.5*deltaT*A)*u(:,1)+0.5*deltaT*B);
U(2:end-1,2:end-1,2) = reshape(u(:,2),Nr,Nz)';

%Following iterations
for i = 2:Nt
    u(:,i+1) = (I-0.5*deltaT*A)\((I+0.5*deltaT*A)*u(:,i));
    U(2:end-1,2:end-1,i+1) = reshape(u(:,i+1),Nr,Nz)';
end
end
```

```
%Plot graphs of the solution in one spatial dimension
%
%Goyens Florentin & Weicker David
close all;
clear all;

[U,r,t] = highCyl(10,100,200);
surf(r,t,U);
title('Fluid velocity in cylinder');
xlabel('r [m]');
ylabel('Time [s]');
zlabel('Speed [m/s]');
```

```
%Plot graphs of two spatial dimensions for different times
%
%Goyens Florentin & Weicker David
close all;
clear all;

[U,r,z,t] = nsCyl(9,200,200);
j = [0 20 50 100];

figure;
for i=1:4
    subplot(2,2,i);
    contourf(r,z,U(:,:,j(i)+1),0:0.0005:0.04,'edgecolor','None');caxis([0 0.04]);↩
        colorbar;
    axis equal;
    title(sprintf('Time=%d',j(i)));xlabel('r [m]');ylabel('z [m]');
end
```

```
%Makes a little animation for the speed in one spatial dimension (the
%height of the cylinder is considered to be infinite)
%
%Goyens Florentin & Weicker David
close all;
clear all;

Nx = 20;
Nt = 200;
tend = 200;

ht = tend/Nt;
[U,r,t] = highCyl(Nx,Nt,tend);

n = Nt+1;
for i=1:n
    titre = sprintf('Time t=%f',(i-1)*ht);
    plot(r,U(i,:));
    title(titre);xlabel('r [m]');ylabel('Speed [m/s]');axis([0 0.04 0 0.04]);
    F(i) = getframe(gcf);
end
movie2avi(F,'anim1D','compression','None');
```

```
%Makes a little animation for the speed in two spatial dimension (we no
%long consider that the height of the cylinder is infinite)
%
%Goyens Florentin & Weicker David
close all;
clear all;

Nr = 10;
Nt = 100;
```

```
tend = 200;

ht = tend/Nt;
[U,r,z,t] = nsCyl(Nr,Nt,tend);

n = Nt+1;
for i=1:n
    titre = sprintf('Time t=%f',(i-1)*ht);
    contourf(r,z,U(:,:,i),0:0.005:0.04);colorbar;caxis([0 0.04]);
    title(titre);xlabel('r [m]');ylabel('z [m]');axis equal;
    F(i) = getframe(gcf);
end
movie(F);
movie2avi(F,'anim2D','compression','None');
```

```
function [] = compare(height)
%COMPARE Plots graphs to compare the solution at differents heights
%with the solution obtained for an infinite height
%INPUT : height is a vector containing the heights we want to compare
%
%Goyens Florentin & Weicker David
close all;

Nr = 9;
h = 0.04/(Nr+1);
Nt = 100;
tend = 200;
Z = round(height/h)+1;

M{1} = 'Infinite height';
for i = 2:length(height)+1
    M{i} = sprintf('z = %f',height(i-1));
end

[u,~,~] = highCyl(Nr,Nt,tend);
[U,r,~,~] = nsCyl(Nr,Nt,tend);
Ntcomp = [1 26 51 76];

figure;
for i = 1:4
    subplot(2,2,i);
    titre = sprintf('Time = %f',tend*(Ntcomp(i)-1)/Nt);
    plot(r,u(Ntcomp(i),:),r,U(Z,:,Ntcomp(i)));xlabel('r [m]');ylabel('Speed [m/s]');
    title(titre);legend(M);
    axis([0 0.04 0 0.04]);
end



end
```

```
function [] = animComp(height)
%ANIMCOMP Makes an animation to compare the solution at differents heights
%with the solution obtained for an infinite height
%INPUT : height is a vector containing the heights we want to compare
%
%Goyens Florentin & Weicker David
close all;

Nr = 9;
h = 0.04/(Nr+1);
Nt = 100;
tend = 200;
Z = round(height/h)+1;

M{1} = 'Infinite height';
```

```
for i = 2:length(height)+1
    M{i} = sprintf('z = %f',height(i-1));
end

ht = tend/Nt;
[U,~,~,~] = nsCyl(Nr,Nt,tend);
[u,r,~] = highCyl(Nr,Nt,tend);

n = Nt+1;
for i=1:n
    titre = sprintf('Time t=%f',(i-1)*ht);
    plot(r,u(i,:),r,U(Z,:,i));
    title(titre);xlabel('r [m]');ylabel('Speed [m/s]');axis([0 0.04 0 0.04]);legend(M)↩
        ;
    F(i) = getframe(gcf);
end
movie2avi(F,'animCom','compression','None');
end
```