Projekt

Die Navier-Stokes Gleichungen beschreiben die Strömung in einer Flüssigkeit. Sie sind gegeben durch

$$\begin{aligned} \rho \mathbf{u}_t + \rho \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p &= \mu \Delta \mathbf{u} \\ \operatorname{div} \mathbf{u} &= 0, \end{aligned} \tag{1}$$

mit dem unbekannten Geschwindigkeitsfeld, bzw. Druck, $\mathbf{u} = (u, v)^T$ und p. ρ und μ sind physikalische Konstanten (Dichte und Viskosität).

(1) Finden Sie eine Koordinaten-Transformation, um (1) wie folgt zu schreiben

$$\begin{aligned} \mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p &= \nu \Delta \mathbf{u} \\ \operatorname{div} \mathbf{u} &= 0, \end{aligned} \tag{2}$$

mit $\nu = 1/\text{Re}$, wobei Re die Reynolds-Zahl heisst.

(2) Wir definieren zunächst die Vortizität $w := \operatorname{curl} \mathbf{u}$. Wenden Sie den curl-Operator auf (2) an, um zu zeigen, dass diese Gleichung sich wie folgt schreiben lässt

$$w_t + \mathbf{u} \cdot \nabla w = \nu \Delta w. \tag{3}$$

(3) Um die Unbekannte $\mathbf{u} = (u, v)^T$ zu berechnen, müssen wir zwei Poisson-Gleichungen lösen. Wie finden wir diesen Gleichungen?

(4) Wir betrachten jetzt (2) auf dem Gebiet $[0, 2\pi]^2$ mit periodischen Randbedingungen. Schreiben Sie eine Matlab-Funktion NAVIERSTOKES.M, um diese Gleichung mit einer Fourier-Kollokation-Spektalmethode zu lösen:

```
function W=NavierStokes(w0,nu,T,N,l)
%% Loeser fuer eine 2d Navier-Stokes Glg
%% u_t+u grad u+grad p=nu Delta u mit period. RB.
%% Zeit Intervall [0,T], N Schritten
%% Speichern von Daten jeden l Schritt
%% fft2 und ifft2 werden benutzt
%% leap frog in der Zeit
%% w0=Anfangsvortizitaet (2D-Matrix)
%% W=3D-Matrix von Vortizitaet Schnappschuss
%% (W(x,y,j)-->j. Schnappschuss von dem Feld
%% an der Stelle (x,y)).
```

Hinweis:

- Mit w^0 lösen Sie die Poisson-Gleichungen (im Fourier-Raum), um u^0 und v^0 zu bekommen.
- Um den ersten Schritt in der Zeit zu bekommen, benutzen wir das explizite Euler-Verfahren auf (3) (immer im Fourier-Raum):

$$\frac{\widehat{w}^1 - \widehat{w}^0}{\Delta t} = -(\widehat{u^0 w_x^0 + v^0 w_y^0}) + \nu \widehat{\Delta w^1}.$$

  (Achtung, der Teil mit dem Laplace-Operator ist implizit gegeben).
- Berechnung von u^1, v^1 und w^1.

  Wir werden auch die Normalisierung MA=MAX(MAX(ABS(W))) benutzen.
- Wir benutzen jetzt das leap-frog Verfahren in der Zeit solange

  J<=N & MAX(MAX(ABS(W)))<=1.2*MA (falls die zweite Bedingung nicht erfüllt ist, geben Sie eine WARNING und brechen Sie ab: es gibt numerische Instabilität).

  Dieses Verfahren lautet

$$\frac{\widehat{w}^{n+1} - \widehat{w}^{n-1}}{2\Delta t} = -(\widehat{u^n w_x^n + v^n w_y^n}) + \nu \widehat{\Delta w^{n+1}}.$$

  Die Daten werden jeden l. Schritt gespeichert mit:

```
if (j-1)/l==floor((j-1)/l),
    q=q+1;
    W(:,:,q)=w;
end
```

(5) Um die Daten zu plotten, schreiben Sie eine Matlab-Funktion PLOTNS.M:

```
function M=plotNS(W)
%% W=3D-Matrix von Vortizitaet Schnappschuss
%% M=Matlab movie
```

Hinweis:

Das Skelett für diese Funktion liegt auf der Homepage der Vorlesung.

```
function M=plotNS(W)
vecMax=16; % # auf Punkten um Pfeile zu zeichnen
N=size(W,3);
[n,m]=size(W(:,:,1));
if n<=vecMax,
  % Dx und Dy in Fourier
  % Laplace-Operator in Fourier
  % Laplace-Loeser
end
x=0:2*pi/n:2*pi*(n-1)/n;
```

```
for j=1:N,
  image(x,x,W(:,:,j),'CDataMapping','scaled');
  if n<=vecMax, % um Pfeile zu zeichnen
    % Berechnung von w-hut:
    what=fft2(W(:,:,j));
    % Berechnung von u und v
    hold on;
    % Plot von Pfeile:
    quiver(x,x,v,u);
    hold off;
  end
  xlabel('y');ylabel('x');
  M(j)=getframe; % um einen Film zu machen
  % Benutzen Sie movie(M), um den Film zu spielen
end
```

(6) Schreiben Sie einen Matlab-Code MAIN.M, um die Navier-Stokes Gleichungen mit folgenden Anfangsdaten zu lösen:

- Konstante Geschwindigkeit $\mathbf{u} = (1, 0)$ oder $\mathbf{u} = (0, 1)$. Fliesst die Flüssigkeit in die richtige Richtung?
- *Kelvin-Helmholtz Instabilität:* Beschreibt das Einspritzen einer Flüssigkeit in einer Flüssigkeit im Ruhezustand.
- *Wirbel-Dipole:* Beschreibt zwei Wirbel, die sich in entgegengesetzte Richtungen drehen.
- Versuchen Sie die Anfangsdaten zu finden, um *Zwei Wirbel-Dipole* zu beschreiben.

Hinweis:

Das Skelett für diesen Code befindet sich auf der Homepage der Vorlesung.

```
clear all
n=6;
m=2^n;
h=2*pi/m;
x=(0:h:2*pi-h)';
X=repmat(x,1,m);
Y=repmat(x',m,1);
k=[0:m/2-1 -m/2:-1];
 % Dx und Dy im Fourierraum

 %% easy test
u0=1;v0=0;
w0=real(ifft2(Dx.*fft2(v0)-Dy.*fft2(u0)));
w=NavierStokes(w0,0.003,10,400,10);
```

```
M=plotNS(w);

 %% Kelvin-Helmholtz Instabilitaet
u0=0.5.*(1+tanh(10.* ...
   (1-4./(2.*pi).*abs(pi-Y)))).* ...
   (1+0.5.*sin(2.*X));
v0=0;
w0=real(ifft2(Dx.*fft2(v0)-Dy.*fft2(u0)));
w=NavierStokes(w0,0.001,50,1000,10);
M=plotNS(w);

 %% Wirbel-Dipole
psi0=0.01;
 % Wirbel1
xv=2.*pi./4;
yv=2.*pi./2+0.05;
lv=0.4.*sqrt(2).*min([xv,yv,2.*pi-xv,2.*pi-yv]);
u01=-2.*(Y-yv)./(lv.^2).*psi0.*...
    exp(-((X-xv).^2+(Y-yv).^2)./lv.^2);
v01=2.*(X-xv)./(lv.^2).*psi0.*...
    exp(-((X-xv).^2+(Y-yv).^2)./lv.^2);
 % Wirbel2
xv=2.*pi./4;
yv=2.*pi./2-0.05;
lv=0.4.*sqrt(2).*min([xv,yv,2.*pi-xv,2.*pi-yv]);
u02=2.*(Y-yv)./(lv.^2).*psi0.*...
    exp(-((X-xv).^2+(Y-yv).^2)./lv.^2);
v02=-2.*(X-xv)./(lv.^2).*psi0.*...
    exp(-((X-xv).^2+(Y-yv).^2)./lv.^2);
 % Dipole
u0=u01+u02;
v0=v01+v02;

w0=real(ifft2(Dx.*fft2(v0)-Dy.*fft2(u0)));
w=NavierStokes(w0,0.001,2000,500,10);
M=plotNS(w);
```

(7) Am Ende, sollten Sie schönen Bildern bekommen:

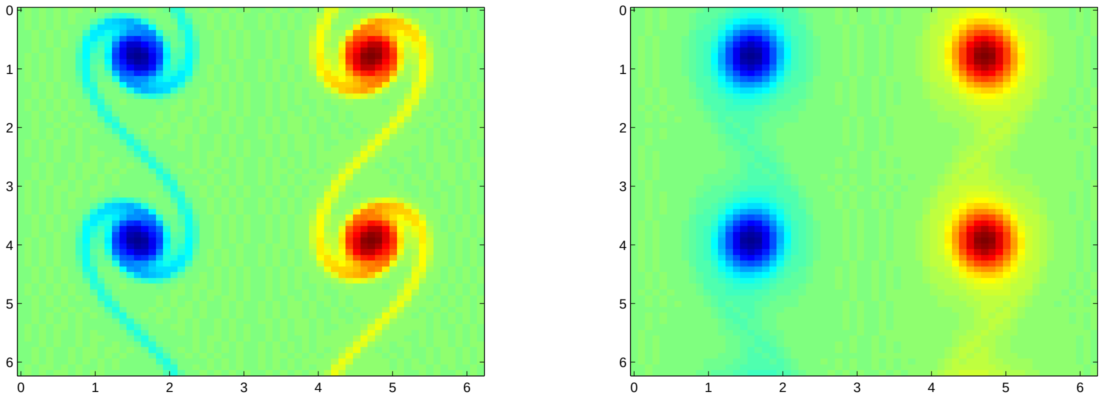

Abbildung 1: Kelvin-Helmholtz Instabilität

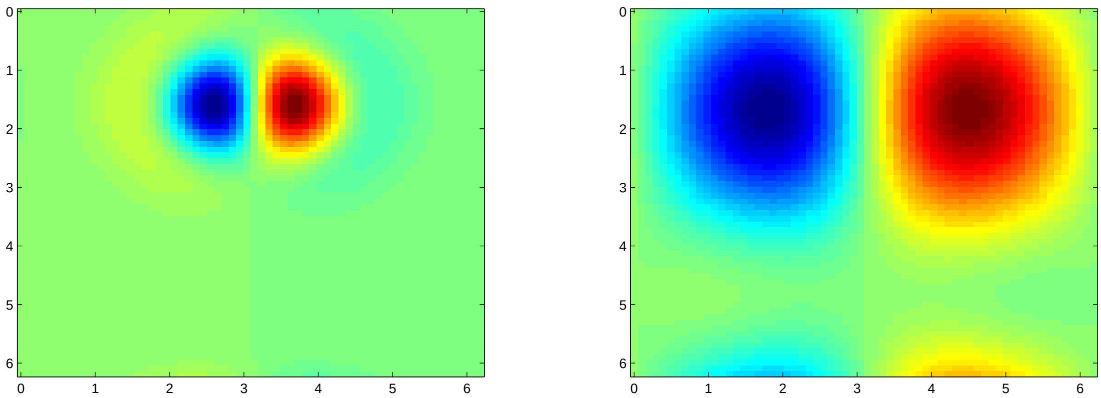

Abbildung 2: Wirbel-Dipole

Bis Ende Juni 2008 erwarten wir eine kurze Präsentation (auf dem Rechner) und ein kleinen Bericht (LaTeX geschrieben) von Ihrem Projekt.