Stable Fluid Solver

Verrà qui presentato un semplice e versatile solver in 2D e 3D per le equazioni di Navier-Stokes di un fluido incomprimibile. Proposto per la prima volta da Jos Stam nel ‘97, si basa sul metodo di Chorin e sul metodo delle caratteristiche. L’evoluzione temporale ad ogni passaggio viene calcolata tramite un’operazione di time-splitting.

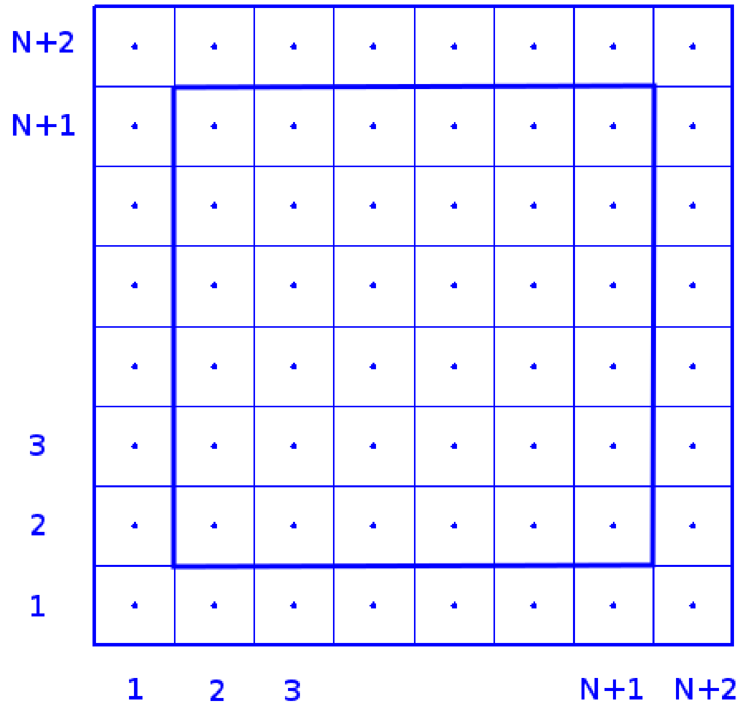
Dinamica dei fluidi

$$\frac{\partial \boldsymbol{u}}{\partial t} = -(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - \frac{1}{\rho}\nabla p + \nu \nabla^2 \boldsymbol{u} + \boldsymbol{f} \qquad \nabla \cdot \boldsymbol{u} = 0$$

$$\frac{\partial \rho}{\partial t} = -(\boldsymbol{u} \cdot \nabla)\rho + \kappa \nabla^2 \rho + S$$

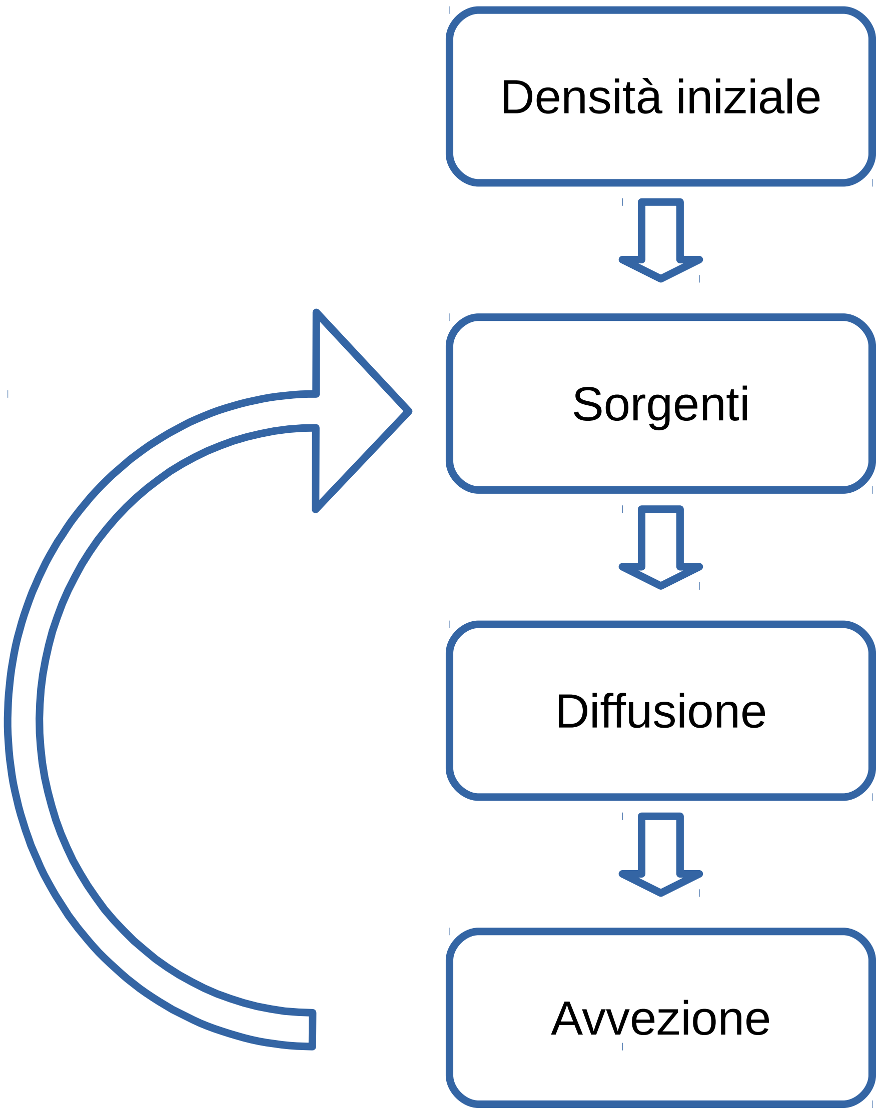
Equazione di Navier-Stokes per un fluido incomprimibile ed equazione per una densità all’interno di un campo di velocità

2D: Griglia



Griglia a volumi finiti di dimensione NxN con uno strato esterno per tener conto delle condizioni al contorno. I valori di densità e velocità sono calcolati al centro delle celle.

2D: Evoluzione della densità



Time-splitting:

i vari termini dell’equazione vengono valutati separatamente introducendo dei passi temporali intermedi.

2D: Diffusione

$$\frac{\partial \rho}{\partial t} = \kappa \nabla^2 \rho$$

$$\frac{\rho_c^k - \rho_c^{k+1}}{\Delta t} = -\kappa \left(\frac{\rho_e^{k+1} + \rho_w^{k+1} - 2\rho_c^{k+1}}{h^2} + \frac{\rho_n^{k+1} + \rho_s^{k+1} - 2\rho_c^{k+1}}{h^2} \right)$$

$$\rho_c^{k+1} - \rho_c^k = \frac{\kappa}{h^2} \Delta t \left(\rho_e^{k+1} + \rho_w^{k+1} + \rho_n^{k+1} + \rho_s^{k+1} - 4\rho_c^{k+1} \right)$$

$$\rho_c^{k+1} = \frac{\rho_c^k + a\left(\rho_e^{k+1} + \rho_w^{k+1} + \rho_n^{k+1} + \rho_s^{k+1}\right)}{1 + 4a} \qquad a = \frac{\kappa}{h^2} \Delta t$$

Discretizzazione BTCS (Backward Time, Centered Space Method) o alternativamente BT e volumi finiti.

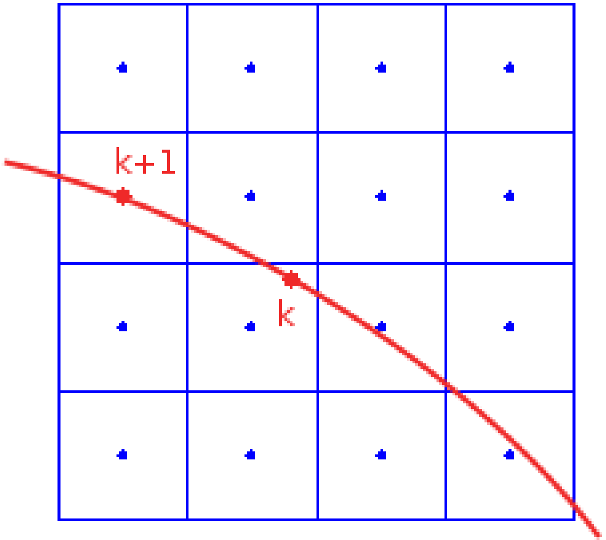
2D: Diffusione (codice)

```
a=dt*diff/(h**2)

do q=1,20
  do j=2,M+1
    do i=2,L+1
      x(i,j) = (x0(i,j) + a*(x(i-1,j)+x(i+1,j) &
                 +x(i,j-1)+x(i,j+1)))/(1+4*a)
    end do
  end do
  call set_bnd_2D(L,M,b,x)
end do
```

Il sistema risultante viene risolto iterativamente (Gauss-Seidel), la subroutine set_bnd si occupa delle condizioni al contorno.

2D: Avvezione



Metodo delle curve caratteristiche:

Il nuovo valore viene calcolato BT ripercorrendo le linee di campo ed effettuando una media dei valori delle celle adiacenti. Stabile, il valore non sarà mai maggiore del massimo presente nel campo precedente.

2D: Avvezione (codice)

```
dt0 = dt/h;

do j=2,M+1
  do i=2,L+1
    x = i-dt0*u(i,j)
    y = j-dt0*v(i,j)

    if (x<1.5) then
      x=1.5;
    endif
    if (x>L+1.5) then
      x=L+1.5
    endif
    i0=int(x)
    i1=i0+1

    if (y<1.5) then
      y=1.5
    endif
```

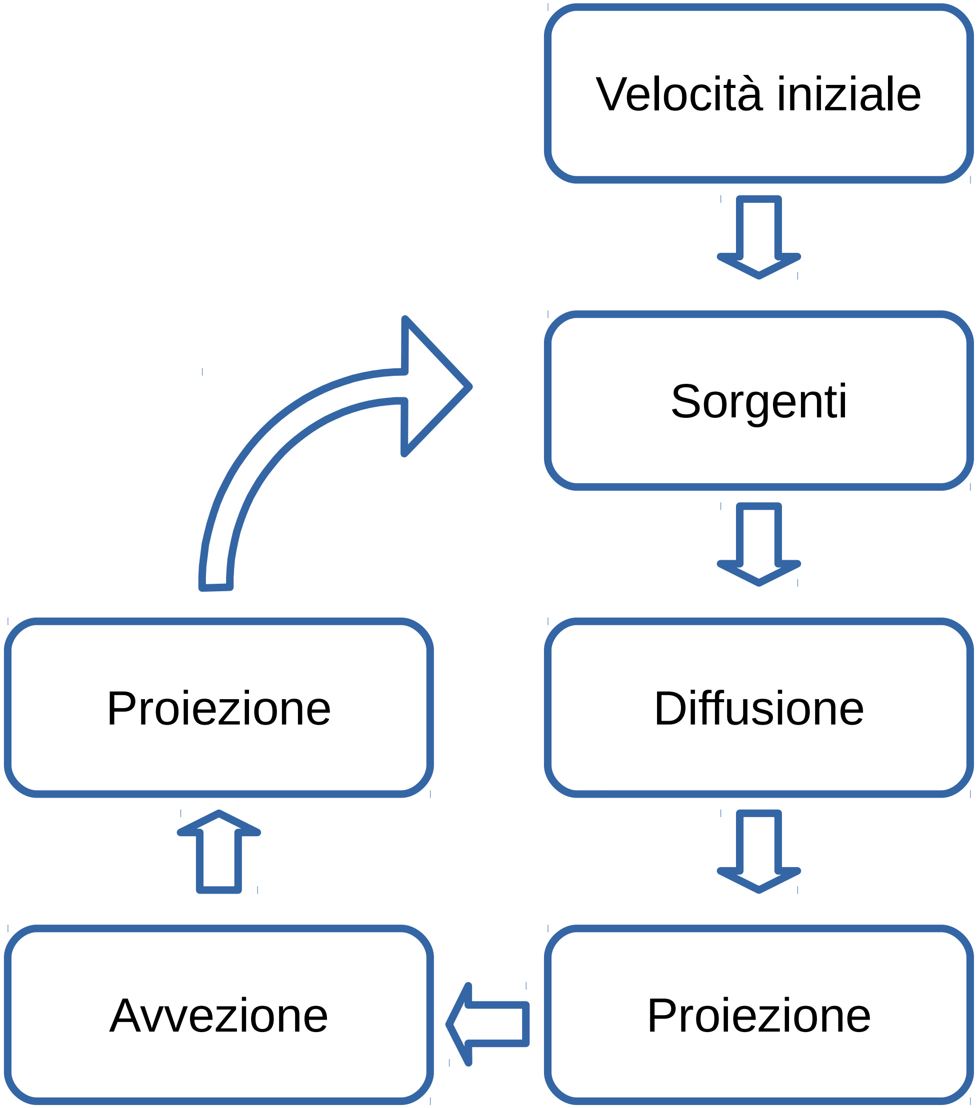
```
    if (y>M+1.5) then
      y=M+1.5
    endif
    j0=int(y)
    j1=j0+1


    r1 = x-i0
    r0 = 1-r1
    s1 = y-j0
    s0 = 1-s1

    d(i,j) = r0*(s0*d0(i0,j0)+s1*d0(i0,j1)) &
                +r1*(s0*d0(i1,j0)+s1*d0(i1,j1))
  end do
end do
call set_bnd_2D(L,M,b,d);
```

Viene effettuata una interpolazione lineare dei valori delle quattro celle adiacenti al punto trovato.

2D: Evoluzione delle velocità



Si diffonde e propaga il campo su sé stesso, l'operazione di proiezione invece accoppia l'equazione di continuità per il fluido incomprimibile.

2D: Proiezione

$$\boldsymbol{w} = \boldsymbol{u} + \nabla q \qquad\qquad \nabla \cdot \boldsymbol{w} = \nabla^2 q$$

$$\boldsymbol{u} = P\,\boldsymbol{w} = \boldsymbol{w} - \nabla q$$

Un campo vettoriale ***w*** può essere decomposto nella somma di un campo soleinodale e di uno irrotazionale (th. di Helmholz), il campo ***u*** quindi soddisfa l'eq. di continuità per un fluido incomprimibile. L'eq. di Navier-Stokes si riduce quindi alla forma seguente (Chorin):

$$\frac{\partial \boldsymbol{u}}{\partial t} = P\left(-(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \nu \nabla^2 \boldsymbol{u} + \boldsymbol{f}\right)$$

2D: Pressure Poisson equation

$$\nabla \cdot \boldsymbol{w} = \nabla^2 q$$

$$\nabla \cdot \boldsymbol{w} = d = \frac{u_e - u_w}{2h} + \frac{v_n - v_s}{2h}$$

$$h^2 d = q_n + q_s + q_e + q_w - 4 q_c$$

$$q_c = \frac{q_n + q_s + q_e + q_w - h^2 d}{4}$$

Si risolve l’equazione di Poisson integrando sui volumi. Per il sistema risultante si procede iterativamente (Gauss-Seidel)

2D: Proiezione (codice)

```
! -h*div(u)
do j=2,M+1
  do i=2,L+1
    div(i,j) = -0.5*h*(u(i+1,j)-u(i-1,j)+v(i,j+1)-v(i,j-1))
    p(i,j) = 0
  end do
end do

call set_bnd_2D(L,M,0,div)
call set_bnd_2D(L,M,0,p)

! ppe
do q=1,20
  do j=2,M+1
    do i=2,L+1
      p(i,j) = (div(i,j)+p(i-1,j)+p(i+1,j) &
                         +p(i,j-1)+p(i,j+1))/4
    end do
  end do
  call set_bnd_2D(L,M,0,p)
end do

! u-grad(p)
do j=2,M+1
  do i=2,L+1
    u(i,j) = u(i,j)-0.5*(p(i+1,j)-p(i-1,j))/h
    v(i,j) = v(i,j)-0.5*(p(i,j+1)-p(i,j-1))/h
  end do
end do

call set_bnd_2D(L,M,1,u)
call set_bnd_2D(L,M,2,v)
```

2D: Condizioni al contorno (codice)

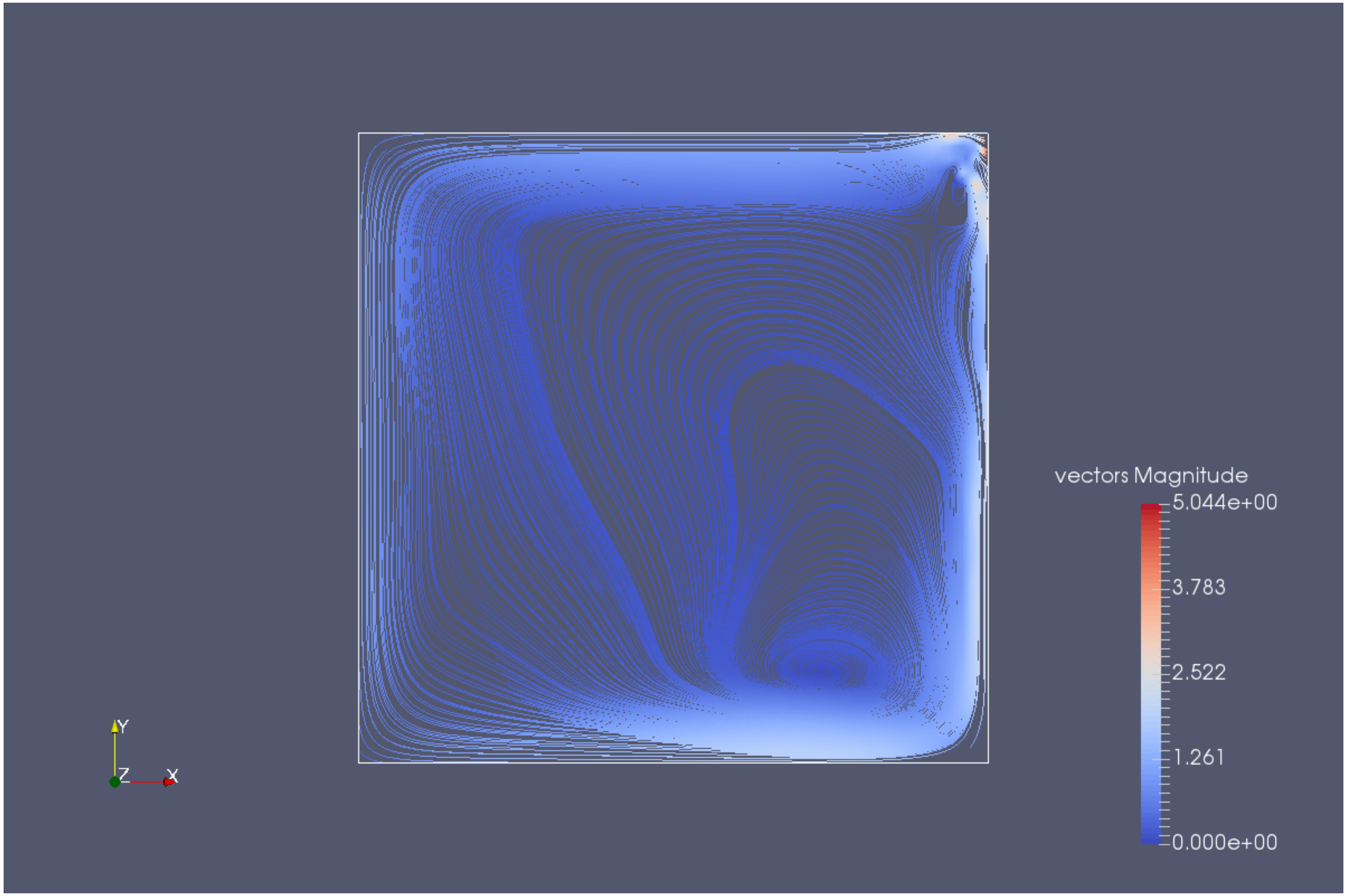
```
! lati
do j=2,M+1
  if(b==1) then
    x(1,j) = -x(2,j)
    x(L+2,j) = -x(L+1,j)
  else
    x(1,j) = x(2,j)
    x(L+2,j) = x(L+1,j)
  endif
end do

do i=2,L+1
  if(b==2) then
    x(i,1) = -x(i,2)
    x(i,M+2) = -x(i,M+1)
  else
    x(i,1) = x(i,2)
    x(i,M+2) = x(i,M+1)
  endif
end do

! vertici
x(1,1) = 0.5*(x(2,1) + x(1,2))
x(1,M+2) = 0.5*(x(2,M+2) + x(1,M+1))
x(L+2,1) = 0.5*(x(L+1,1) + x(L+2,2))
x(L+2,M+2) = 0.5*(x(L+1,M+2) + x(L+2,M+1))
```

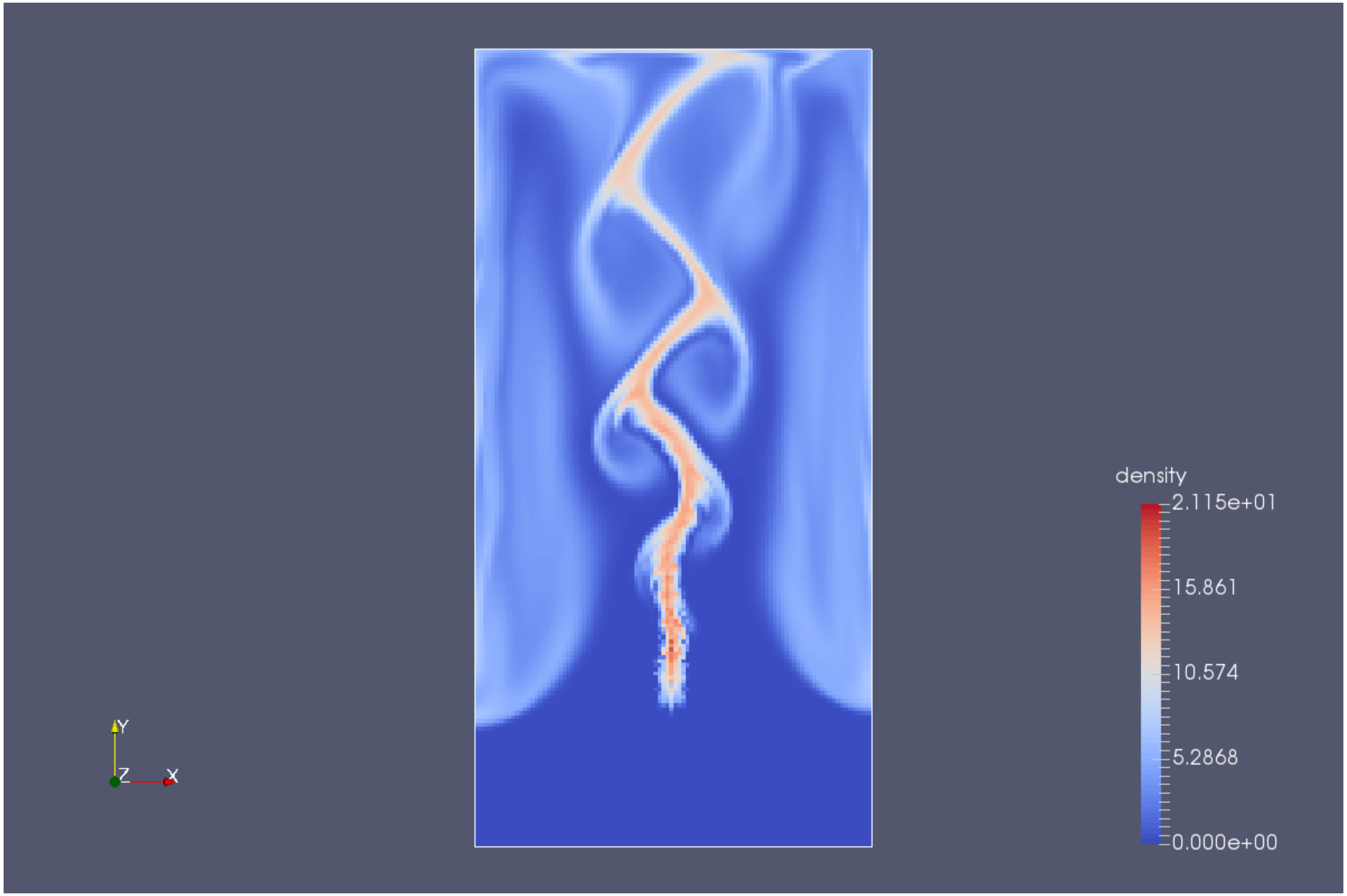
Condizioni al contorno per una scatola chiusa: conservazione della massa per la densità e velocità perpendicolare nulla per le velocità.

2D: Risultati



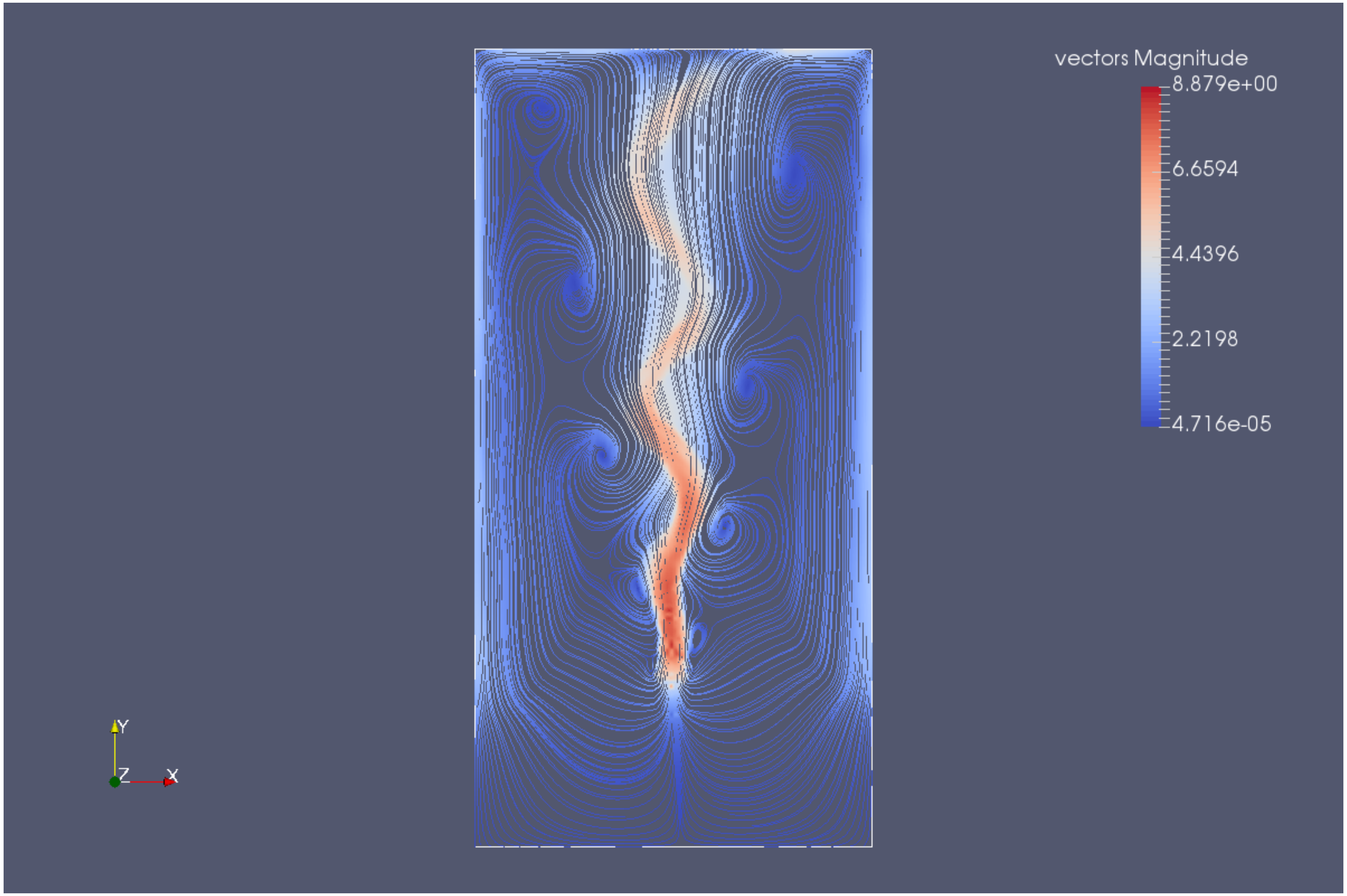
Slip-slit: 50x50, t=1000, v=10.0

2D: Risultati



Stream, densità: 100x200, t=200, v=5.0, r=5

2D: Risultati



Stream, velocità: 100x200, t=200, v=5.0, r=5

Modello 3D

La versione 3D del solver ha la stessa struttura e sfrutta le stesse funzioni della versione bidimensionale tranne per piccole modifiche.
Verranno in seguito mostrate le principali differenze.

3D: Diffusione

$$\frac{\partial \rho}{\partial t} = \kappa \nabla^2 \rho$$

$$\frac{\rho_c^k - \rho_c^{k+1}}{\Delta t} = -\kappa \left(\frac{\rho_e^{k+1} + \rho_w^{k+1} - 2\rho_c^{k+1}}{h^2} + \frac{\rho_n^{k+1} + \rho_s^{k+1} - 2\rho_c^{k+1}}{h^2} + \frac{\rho_u^{k+1} + \rho_d^{k+1} - 2\rho_c^{k+1}}{h^2} \right)$$

$$\rho_c^{k+1} - \rho_c^k = \frac{\kappa}{h^2} \Delta t \left(\rho_e^{k+1} + \rho_w^{k+1} + \rho_n^{k+1} + \rho_s^{k+1} + \rho_u^{k+1} + \rho_d^{k+1} - 6\rho_c^{k+1} \right)$$

$$\rho_c^{k+1} = \frac{\rho_c^k + a\left(\rho_e^{k+1} + \rho_w^{k+1} + \rho_n^{k+1} + \rho_s^{k+1} + \rho_u^{k+1} + \rho_d^{k+1} \right)}{1 + 6a} \qquad a = \frac{\kappa}{h^2} \Delta t$$

Discretizzazione BTCS (Backward Time, Centered Space Method) o alternativamente BT e volumi finiti.

3D: Avvezione

```
    dt0 = dt/h;

    do k=2,N+1
      do j=2,M+1
        do i=2,L+1
          x = i-dt0*u(i,j,k)
          y = j-dt0*v(i,j,k)
          z = k-dt0*w(i,j,k)

          if (x<1.5) then
            x=1.5;
          endif
          if (x>L+1.5) then
            x=L+1.5
          endif
          i0=int(x)
          i1=i0+1

          if (y<1.5) then
            y=1.5
          endif
          if (y>M+1.5) then
            y=M+1.5
          endif
          j0=int(y)
          j1=j0+1
```

```
          if (z<1.5) then
            z=1.5
          endif
          if (z>N+1.5) then
            z=N+1.5
          endif
          k0=int(z)
          k1=k0+1

          r1 = x-i0
          r0 = 1-r1
          s1 = y-j0
          s0 = 1-s1
          t1 = z-k0
          t0 = 1-t1

          d(i,j,k) = r0*(s0*(t0*d0(i0,j0,k0)+t1*d0(i0,j0,k1)) &
                        +s1*(t0*d0(i0,j1,k0)+t1*d0(i0,j1,k1))) &
                    +r1*(s0*(t0*d0(i1,j0,k0)+t1*d0(i1,j0,k1)) &
                        +s1*(t0*d0(i1,j1,k0)+t1*d0(i1,j1,k1)))
        end do
      end do
    end do
    call set_bnd_3D(L,M,N,b,d);
```

Come per il caso 2D viene effettuata una interpolazione lineare dei valori delle otto celle adiacenti al punto trovato BT.

3D: Pressure Poisson equation

$$\nabla \cdot \boldsymbol{w} = \nabla^2 q$$

$$\nabla \cdot \boldsymbol{w} = d = \frac{u_e - u_w}{2h} + \frac{v_n - v_s}{2h} + \frac{w_u - w_d}{2h}$$

$$h^2 d = q_n + q_s + q_e + q_w + q_u + q_d - 6 q_c$$

$$q_c = \frac{q_n + q_s + q_e + q_w + q_u + q_d - h^2 d}{6}$$

Per il sistema risultante si procede iterativamente. E' stato usato sia il metodo di Gauss-Seidel sia, per una migliore convergenza, il metodo del gradiente coniugato.

3D: Gradiente coniugato

$$6q_c - q_n - q_s - q_e - q_w - q_u - q_d = h^2 d$$

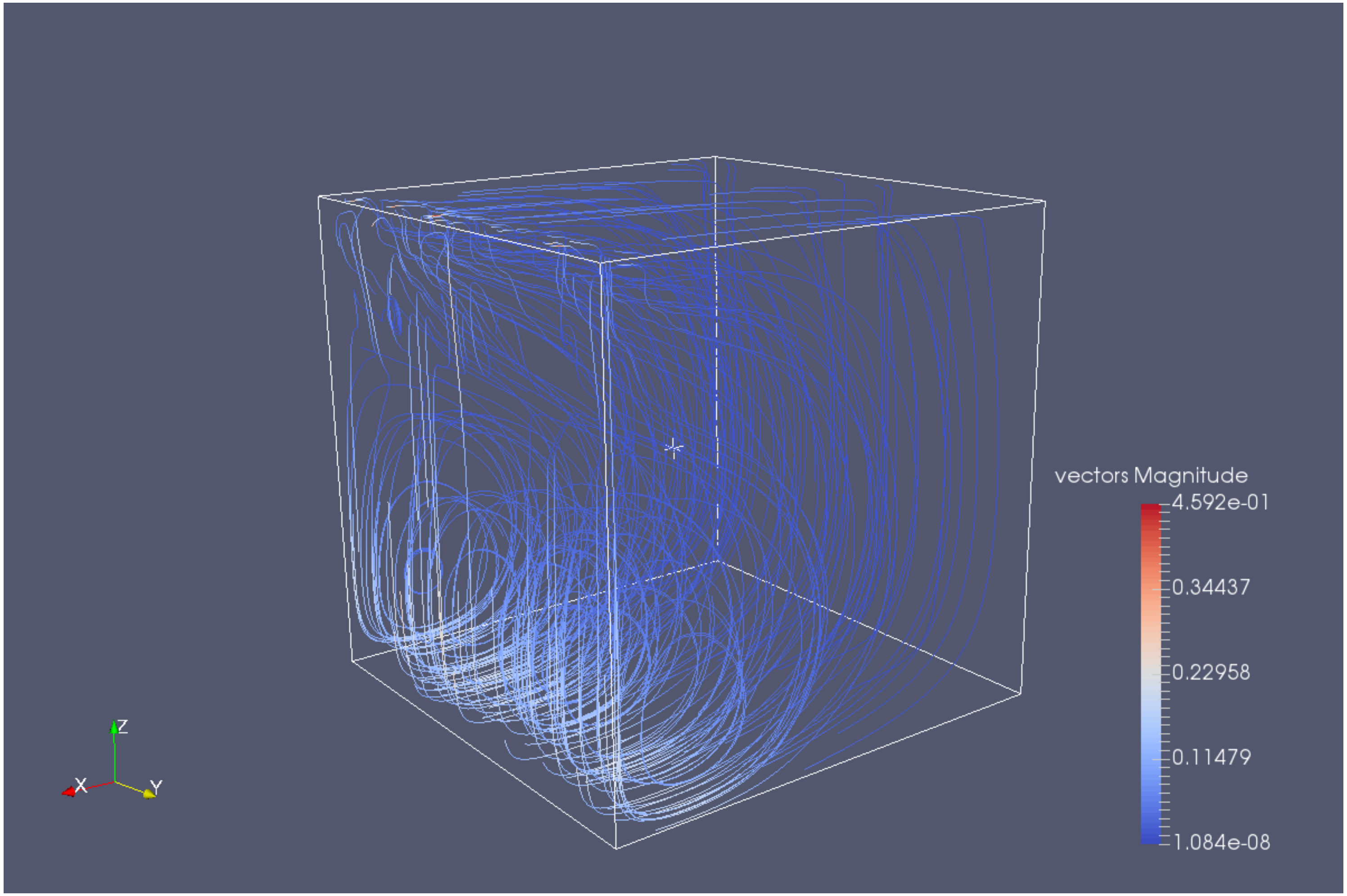
La matrice del sistema da risolvere *(LxNxM)x(LxMxN)* contiene al suo interno le condizioni al contorno ($q_{bordo}=q_c$): ai termini diagonali vicini al bordo viene direttamente sottratta la cella del contorno. La matrice avendo principalmente termini nulli viene memorizzata in formato CSR.

3D: Condizioni al contorno

Condizioni al contorno per una scatola chiusa: conservazione della massa per la densità e velocità perpendicolare nulla per le velocità. Le condizioni vengono imposte sulle facce, poi si esegue una media sugli spigoli, infine una media sui vertici.

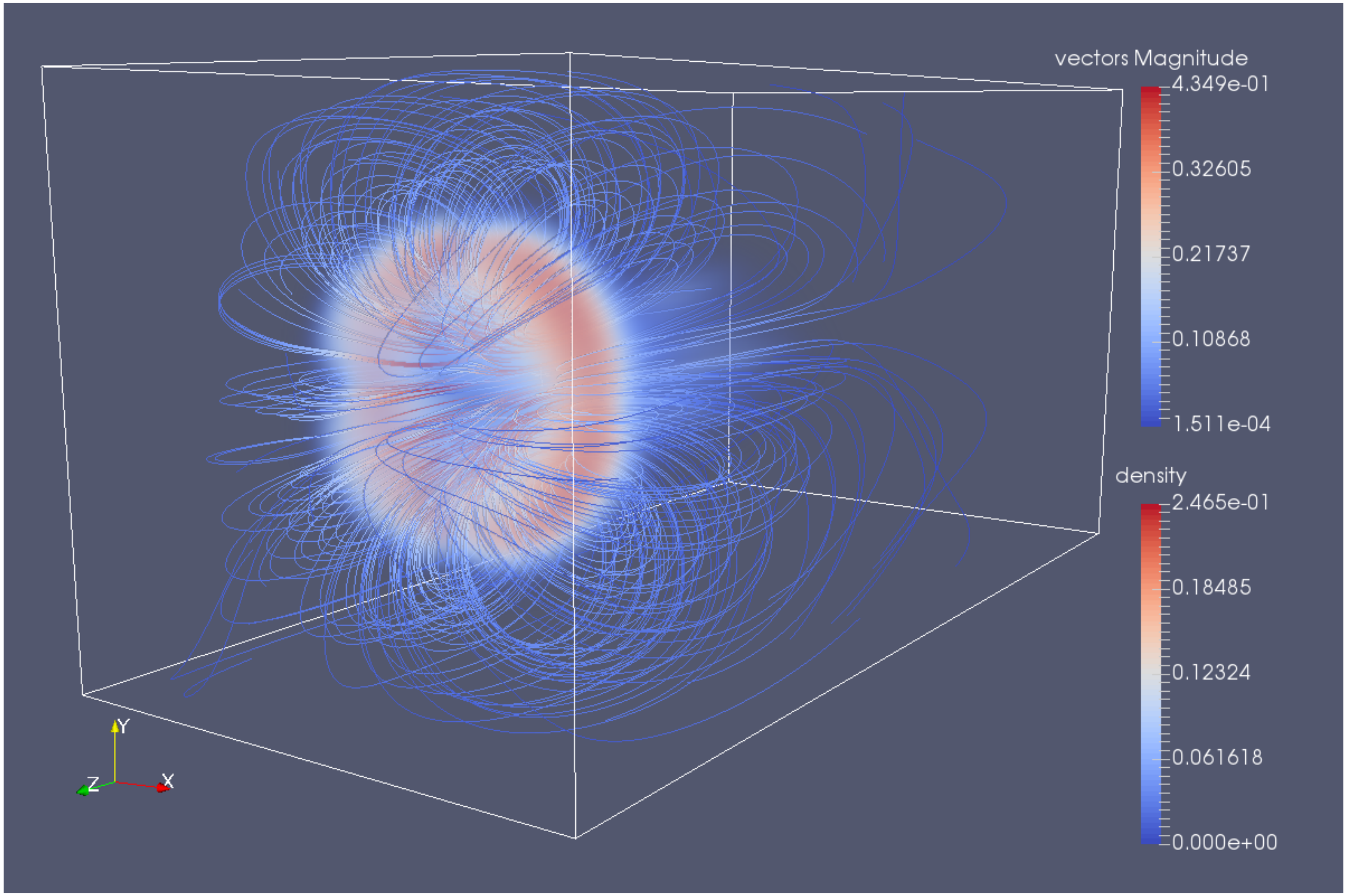
Presente come nella versione 2D la possibilità di inserire una velocità costante su una faccia.

3D: Risultati



Slip-slit: 40x40x40, t=500, v=10.0

3D: Risultati



Ring: 50x50x100, t=150, v=10.0, r=5