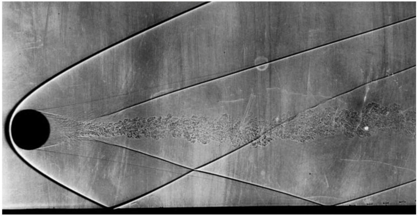
[302.044] Numerical Methods in Fluid Dynamics

Transport phenomena in weak formulation

F. Romanò & P.-E. des Boscs
Institute of Fluid Mechanics and Heat Transfer, TU Wien



December 14th 2017, Vienna, Austria

francesco.romano@tuwien.ac.at
pierre-emmanuel.boscs@tuwien.ac.at

Advection Equation

Hyperbolic PDEs in FreeFEM++

Advection Equation

Advection Equation

Hyperbolic PDEs in FreeFEM++

Advection Equation

- Strong form:

Advection Equation

Hyperbolic PDEs in FreeFEM++

Advection Equation

- Strong form:

$$\begin{cases} \partial_t u - \mathbf{a} \cdot \nabla u(\mathbf{x},\mathbf{t}) = f(\mathbf{x},\mathbf{t}) \quad , \quad \mathbf{x} \in \Omega \, , \; t > 0 \\ \\ +\text{Dirichlet BCs} \end{cases}$$

Advection Equation

Hyperbolic PDEs in FreeFEM++

Advection Equation

- Strong form:

$$\begin{cases} \partial_t u - \mathbf{a} \cdot \nabla u(\mathbf{x},\mathbf{t}) = f(\mathbf{x},\mathbf{t}) \quad , \quad \mathbf{x} \in \Omega \ , \ t > 0 \\ \\ +\text{Dirichlet BCs} \end{cases}$$

- Weak form:

Advection Equation

Hyperbolic PDEs in FreeFEM++

Advection Equation

- Strong form:

$$\begin{cases} \partial_t u - \mathbf{a} \cdot \nabla u(\mathbf{x},\mathbf{t}) = f(\mathbf{x},\mathbf{t}) \quad , \quad \mathbf{x} \in \Omega \ , \ t > 0 \\ +\text{Dirichlet BCs} \end{cases}$$

- Weak form:

$$\begin{cases} \int_\Omega v \partial_t u d\Omega - \int_\Omega v\mathbf{a} \cdot \nabla u d\Omega = \int_\Omega f v d\Omega \quad , \quad \mathbf{x} \in \Omega \\ u|_{\Gamma = \partial\Omega_{dep}} = g \end{cases}$$

Advection Equation

Hyperbolic PDEs in FreeFEM++

Characteristic-Galerkin Method

Advection Equation

Hyperbolic PDEs in FreeFEM++

Characteristic-Galerkin Method

- For unforced advection ($f = 0$):

Advection Equation

Hyperbolic PDEs in FreeFEM++

[302.044] Numerical Methods in Fluid Dynamics

Characteristic-Galerkin Method

- For unforced advection ($f = 0$):

$$\dot{\mathbf{X}}(\tau) = \mathbf{a}\left(\mathbf{X}(\tau), \tau\right)$$

Advection Equation

[302.044] Numerical Methods in Fluid Dynamics

Hyperbolic PDEs in FreeFEM++

Characteristic-Galerkin Method

- For unforced advection ($f = 0$):

$$\dot{\mathbf{X}}(\tau) = \mathbf{a}\left(\mathbf{X}(\tau), \tau\right)$$

- On the characteristic line reference frame:

Advection Equation

Hyperbolic PDEs in FreeFEM++

Characteristic-Galerkin Method

- For unforced advection ($f = 0$):

$$\dot{\mathbf{X}}(\tau) = \mathbf{a}\left(\mathbf{X}(\tau), \tau\right)$$

- On the characteristic line reference frame:

$$\frac{D}{Dt}\left[u(\mathbf{X}(t), t)\right] = f(\mathbf{X}(t), t)$$

where $\frac{D}{Dt}\left[\mathbf{X}(t), t\right] = \mathbf{a}\left(\mathbf{X}(t), t\right)$

Advection Equation

Hyperbolic PDEs in FreeFEM++

Characteristic-Galerkin Method

- For unforced advection ($f = 0$):

$$\dot{\mathbf{X}}(\tau) = \mathbf{a}\left(\mathbf{X}(\tau), \tau\right)$$

- On the characteristic line reference frame:

$$\frac{D}{Dt}\left[u(\mathbf{X}(t), t))\right] = f(\mathbf{X}(t), t))$$

where $\frac{D}{Dt}\left[\mathbf{X}(t), t)\right] = \mathbf{a}\left(\mathbf{X}(t), t\right)$

- Using a 2-points FT scheme for discretizing the total derivative:

$$\frac{u^{n+1}(\mathbf{X}^{n+1}) - u^n(\mathbf{X}^n)}{\Delta t} = f^n$$

where $\mathbf{X}^{n+1} = \mathbf{X}((k+1)\Delta t) = \mathbf{x}$

Advection Equation

[302.044] Numerical Methods in Fluid Dynamics

Hyperbolic PDEs in FreeFEM++

Taylor Expansion

Advection Equation

Hyperbolic PDEs in FreeFEM++

Taylor Expansion

$$\begin{aligned} u(\mathbf{X}(k\Delta t)) =& u(\mathbf{X}((k+1)\Delta t)) - \\ & \Delta t \sum_{i=1}^{N_{dim}} \frac{\partial \left[u^k(\mathbf{X}((k+1)\Delta t)) \right]}{\partial x_i} \frac{D\left[\mathbf{X}_i((k+1)\Delta t)\right]}{Dt} + o(\Delta t) \\ =& u(\mathbf{X}((k+1)\Delta t)) - \Delta t \mathbf{a}^n(\mathbf{x}) \cdot \nabla u^n(\mathbf{x}) + o(\Delta t) \\ & \longrightarrow u^n(\mathbf{x} - \mathbf{a}\Delta t) = u^n(\mathbf{x}) - \Delta t \mathbf{a}^n(\mathbf{x}) \cdot \nabla u^n(\mathbf{x}) + o(\Delta t) \end{aligned}$$

Advection Equation

Hyperbolic PDEs in FreeFEM++

Taylor Expansion

$$
\begin{aligned}
u(\mathbf{X}(k\Delta t)) =& u(\mathbf{X}((k+1)\Delta t)) - \\
& \Delta t \sum_{i=1}^{N_{dim}} \frac{\partial \left[u^k(\mathbf{X}((k+1)\Delta t)) \right]}{\partial x_i} \frac{D\left[\mathbf{X}_i((k+1)\Delta t)\right]}{Dt} + o(\Delta t) \\
=& u(\mathbf{X}((k+1)\Delta t)) - \Delta t \mathbf{a}^n(\mathbf{x}) \cdot \nabla u^n(\mathbf{x}) + o(\Delta t) \\
& \longrightarrow u^n(\mathbf{x} - \mathbf{a}\Delta t) = u^n(\mathbf{x}) - \Delta t \mathbf{a}^n(\mathbf{x}) \cdot \nabla u^n(\mathbf{x}) + o(\Delta t)
\end{aligned}
$$

$$
\boxed{u^n(\mathbf{x} - \mathbf{a}\Delta t) \approx u^n(\mathbf{x}) - \Delta t \mathbf{a}^n(\mathbf{x}) \cdot \nabla u^n(\mathbf{x})}
$$

Advection Equation

Hyperbolic PDEs in FreeFEM++

Taylor Expansion

$$
\begin{aligned}
u(\mathbf{X}(k\Delta t)) =& u(\mathbf{X}((k+1)\Delta t)) - \\
& \Delta t \sum_{i=1}^{N_{dim}} \frac{\partial \left[u^k(\mathbf{X}((k+1)\Delta t)) \right]}{\partial x_i} \frac{D\left[\mathbf{X}_i((k+1)\Delta t)\right]}{Dt} + o(\Delta t) \\
=& u(\mathbf{X}((k+1)\Delta t)) - \Delta t \mathbf{a}^n(\mathbf{x}) \cdot \nabla u^n(\mathbf{x}) + o(\Delta t) \\
& \longrightarrow u^n(\mathbf{x} - \mathbf{a}\Delta t) = u^n(\mathbf{x}) - \Delta t \mathbf{a}^n(\mathbf{x}) \cdot \nabla u^n(\mathbf{x}) + o(\Delta t)
\end{aligned}
$$

$$
\boxed{u^n(\mathbf{x} - \mathbf{a}\Delta t) \approx u^n(\mathbf{x}) - \Delta t \mathbf{a}^n(\mathbf{x}) \cdot \nabla u^n(\mathbf{x})}
$$

- Advection Equation: FreeFEM++ `convect` operator:

$$
\texttt{convect}(\mathbf{a}, -\Delta t, u^n) = u^n(\mathbf{x}) - \Delta t \mathbf{a}^n(\mathbf{x}) \cdot \nabla u^n(\mathbf{x}) \approx u^{n+1}(\mathbf{x})
$$

Advection Equation

[302.044] Numerical Methods in Fluid Dynamics

Hyperbolic PDEs in FreeFEM++

2-D Advection Equation

- First case: $\Omega = [0,1] \times [0,1]$ and $\mathbf{a}$ constant

```
mesh Omega = square(60,20);
fespace Vh(Omega,P1);
Vh u0 = exp(-10*((x-0.3)^2 +(y-0.3)^2));
Vh a1 = 1, a2 = 1; u;
real dt = 0.0017,t=0;
for (int i=0; i<200; i++) {
t += dt;
u = convect([a1,a2],-dt,u0);
u0 = u;
plot(u,fill=1,wait=0);
};
```

Advection Equation

Object Oriented Programming in FreeFEM++

FreeFEM++: Object Oriented Programming

- First case: $\Omega = [0,1] \times [0,1]$ and **a** constant

Mesh Block

```
mesh Omega = square(60,20);
```

Functional Space Block

```
fespace Vh(Omega,P1);
Vh u0 = exp(-10*((x-0.3)^2 +(y-0.3)^2));
Vh a1 = 1, a2 = 1; u;
```

Solver Block

```
real dt = 0.0017,t=0;
for (int i=0; i<200; i++) {
t += dt; u = convect([a1,a2],-dt,u0);
u0 = u; plot(u,fill=1,wait=0); };
```

Advection Equation

Object Oriented Programming in FreeFEM++

FreeFEM++: Object Oriented Programming

- Second case: $\Omega = \left\{x^2 + y^2 \leq 1\right\}$ and **a** variable

```
border C(t=0, 2*pi) {x=cos(t); y=sin(t);};
mesh Omega = buildmesh(C(200));
fespace Vh(Omega,P1);
Vh u0 = exp(-10*((x-0.3)^2 +(y-0.3)^2));
Vh a1 = y, a2 = -x; u;
real dt = 0.17,t=0;
for (int i=0; i<200; i++) {
t += dt;
u = convect([a1,a2],-dt,u0);
u0 = u;
plot(u,fill=1,wait=0);
};
```

Advection Equation

[302.044] Numerical Methods in Fluid Dynamics

Object Oriented Programming in FreeFEM++

FreeFEM++: Object Oriented Programming

- Second case: $\Omega = \left\{x^2 + y^2 \leq 1\right\}$ and **a** variable

Mesh Block Changed

```
border C(t=0,2*pi) {x=cos(t); y=sin(t);};
mesh Omega = buildmesh(C(200));
```

Functional Space Block: Changed

```
fespace Vh(Omega,P1);
Vh u0 = exp(-10*((x-0.3)^2 +(y-0.3)^2));
Vh a1 = y, a2 = -x; u;
```

Solver Block: Unchanged

Wave Equation

Hyperbolic PDEs in FreeFEM++

Advection Equation

Wave Equation

Hyperbolic PDEs in FreeFEM++

Advection Equation

- Strong form:

Wave Equation

Hyperbolic PDEs in FreeFEM++

Advection Equation

- Strong form:

$$\begin{cases} \partial_{tt} p - c^2 \nabla^2 p(\mathbf{x},\mathbf{t}) = f(\mathbf{x},\mathbf{t}) \quad , \quad \mathbf{x} \in \Omega \ , \ t > 0 \\ \\ +\text{Dirichlet BCs} \end{cases}$$

Wave Equation

Hyperbolic PDEs in FreeFEM++

Advection Equation

- Strong form:

$$\begin{cases} \partial_{tt} p - c^2 \nabla^2 p(\mathbf{x},\mathbf{t}) = f(\mathbf{x},\mathbf{t}) \quad , \quad \mathbf{x} \in \Omega \ , \ t > 0 \\ \\ +\text{Dirichlet BCs} \end{cases}$$

- Weak form:

Wave Equation

Hyperbolic PDEs in FreeFEM++

Advection Equation

- Strong form:

$$\begin{cases} \partial_{tt} p - c^2 \nabla^2 p(\mathbf{x},\mathbf{t}) = f(\mathbf{x},\mathbf{t}) \quad , \quad \mathbf{x} \in \Omega \ , \ t > 0 \\ \text{+Dirichlet BCs} \end{cases}$$

- Weak form:

$$\boxed{\begin{cases} \int_\Omega v \partial_{tt} p d\Omega + \int_\Omega c^2 \nabla p \cdot \nabla v d\Omega = \int_\Omega f v d\Omega \quad , \quad \mathbf{x} \in \Omega \\ p|_{\Gamma = \partial\Omega_{dep}} = g \end{cases}}$$

Wave Equation

Hyperbolic PDEs in FreeFEM++

2-D Wave Equation

- First case: $\Omega = \left\{ x^2 + y^2 \leq 1 \right\}$ constant

```
border Gamma(t=0, 2*pi) { x=cos(t); y=sin(t); };
mesh Omega = buildmesh(Gamma(200));
fespace Vh(Omega,P2);
Vh u, uh;
Vh uprev  = 5*exp(-100*((x)^2+(y)^2 ));
Vh uprev2 = 5*exp(-100*((x)^2+(y)^2 ));
real dt = 0.01,t=0;
macro Grad(u)[dx(u),dy(u)]
problem solveWave(u,uh)
= int2d(Omega)(2*Grad(u){'} *Grad(uh) + u*uh/dt/dt)
+ int2d(Omega)((-2*uprev + uprev2) * uh/dt/dt)
+ on(1,2,3,4,u=0);
```

Wave Equation

Hyperbolic PDEs in FreeFEM++

2-D Wave Equation

- First case: $\Omega = \left\{x^2 + y^2 \leq 1\right\}$ constant

```
real[int] viso(60);
for(int i=0;i<viso.n;i++)
        viso[i]=i*0.05-1.1;
for (int i=0; i<5000; i++)
{
 t += dt; solveWave; uprev2 = uprev; uprev = u;
 if (i%10==0)
  {
    plot(u,fill=1,wait=0,viso=viso(0:viso.n-1),dim=3,
        value=true,cmm="t = "+t+" i "+i );
  }
};
```

Isentropic Inviscid Flow

Hyperbolic PDEs in FreeFEM++

Isentropic Inviscid Flow[1]

[1] Under the hypothesis of quasi-stationarity near the boundaries.
O. Pironneau, *Corrected Nonconservative Scheme*,
Chinese Annals of Mathematics (1/6/5)

Isentropic Inviscid Flow

Hyperbolic PDEs in FreeFEM++

Isentropic Inviscid Flow[1]

- Strong form:

[1] Under the hypothesis of quasi-stationarity near the boundaries.
O. Pironneau, *Corrected Nonconservative Scheme*,
Chinese Annals of Mathematics (1/6/5)

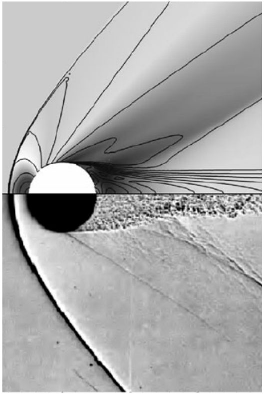
Isentropic Inviscid Flow

Hyperbolic PDEs in FreeFEM++

Isentropic Inviscid Flow[1]

- Strong form:

$$\begin{cases} \rho_{,t} + \nabla \cdot (\rho \mathbf{u}) = 0 \\ (\rho \mathbf{u})_{,t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u} + \rho^{\gamma}) = 0 \\ + \text{Dirichlet BCs} \end{cases}$$



[1] Under the hypothesis of quasi-stationarity near the boundaries.
O. Pironneau, *Corrected Nonconservative Scheme*,
Chinese Annals of Mathematics (1/6/5)

Isentropic Inviscid Flow

Hyperbolic PDEs in FreeFEM++

Isentropic Inviscid Flow

Isentropic Inviscid Flow

Hyperbolic PDEs in FreeFEM++

Isentropic Inviscid Flow

- Strong form:

Isentropic Inviscid Flow

Hyperbolic PDEs in FreeFEM++

Isentropic Inviscid Flow

- Strong form:

$$\begin{cases} \dfrac{1}{\overline{\rho^n}\Delta t}\left(\rho^{n+1} - \rho^n o X^n\right) + \nabla \cdot \mathbf{u}^{n+1} = 0 \\[2ex] \dfrac{\overline{\rho^{n^{2-\gamma}}}}{\gamma \Delta t}\left(\mathbf{u}^{n+1} - \mathbf{u}^n o \tilde{X}^n\right) + \nabla \rho^{n+1} = 0 \end{cases}$$

where $\tilde{\mathbf{X}}$ is computed with the velocity $\overline{\rho \mathbf{u}}/\overline{\rho}$.

Isentropic Inviscid Flow

Hyperbolic PDEs in FreeFEM++

Isentropic Inviscid Flow

- Strong form:

$$\begin{cases} \dfrac{1}{\overline{\rho^n}\Delta t}\left(\rho^{n+1} - \rho^n o X^n\right) + \nabla \cdot \mathbf{u}^{n+1} = 0 \\[2ex] \dfrac{\overline{\rho^{n^{2-\gamma}}}}{\gamma \Delta t}\left(\mathbf{u}^{n+1} - \mathbf{u}^n o \tilde{X}^n\right) + \nabla \rho^{n+1} = 0 \end{cases}$$

  where $\tilde{\mathbf{X}}$ is computed with the velocity $\overline{\rho \mathbf{u}}/\overline{\rho}$.

- Weak form:

Isentropic Inviscid Flow

Hyperbolic PDEs in FreeFEM++

Isentropic Inviscid Flow

- Strong form:

$$\begin{cases} \dfrac{1}{\overline{\rho^n}\Delta t}\left(\rho^{n+1} - \rho^n o X^n\right) + \nabla \cdot \mathbf{u}^{n+1} = 0 \\ \dfrac{\overline{\rho^{n^{2-\gamma}}}}{\gamma \Delta t}\left(\mathbf{u}^{n+1} - \mathbf{u}^n o \tilde{X}^n\right) + \nabla \rho^{n+1} = 0 \end{cases}$$

where $\tilde{\mathbf{X}}$ is computed with the velocity $\overline{\rho \mathbf{u}}/\overline{\rho}$.

- Weak form:

$$\begin{cases} \int_\Omega \dfrac{w_h \rho_h^{n+1}}{\overline{\rho_h^n}} - \Delta t \left(\int_\Omega \mathbf{u}_h^{n+1} \cdot \nabla w_h - \int_\Gamma \mathbf{u}_h^{n+1} \cdot \mathbf{n} w_h\right) = \int_\Omega \dfrac{\rho_h^n o X^n w_h}{\overline{\rho_h^n}} \\ \int_\Omega \overline{\rho_h^{n^{2-\gamma}}} \mathbf{u}_h^{n+1} \mathbf{v}_h + \gamma \Delta t \int_\Omega \nabla \cdot (\rho^{n+1} \mathbf{v}_h) = \int_\Omega \overline{\rho^{n^{2-\gamma}}} \mathbf{u}_h^n o X^n \mathbf{v}_h \end{cases}$$

Isentropic Inviscid Flow

Hyperbolic PDEs in FreeFEM++

Isentropic Inviscid Flow

```
real pi2=atan(1.0)*2, R=0.3, x1=-3, x2=1.0, y2=1.5;
border a1(t=x1,-2-R){x=t; y=0;}
border a2(t=-2+R,x2){x=t; y=0;}
border a3(t=0,y2){x=x2;y=t;}
border a4(t=x2,x1){x=t;y=y2;}
border a5(t=y2,0){x=x1;y=t;}
border bb(t=pi2*2,0){x=-2+R*cos(t);
                     y=R*sin(t);}
mesh Th = buildmesh(a1(30)+a2(40)+a3(20)+a4(60)+a5(20)+
    bb(30));
plot(Th,wait=1);
real dt=0.02, u0=2*sqrt(1.4),
     v0=0, visc=0.00125;
```

Isentropic Inviscid Flow

Hyperbolic PDEs in FreeFEM++

Isentropic Inviscid Flow

```
fespace Vh(Th,P1);
Vh u,v,r,r1,u1,v1,rh,uh,vh,fro;
problem eul(u,v,r,uh,vh,rh)
= int2d(Th)((fro*(u*uh+v*vh)+r*rh/r1)/dt
+ (dx(r)*uh+ dy(r)*vh-dx(rh)*u-dy(rh)*v))
+ int2d(Th)(-(rh*convect([u1,v1],-dt,r1)/r1
+ fro*(uh*convect([u1,v1],-dt,u1)
+ vh*convect([u1,v1],-dt,v1)))/dt)
+ int1d(Th,3)(rh*u) + int1d(Th,4)(rh*v)
+ on(5,r=1) + on(5,u=u0) + on(1,2,5,v=0);
u1= u0; v1= v0; r1 = 1;
for(int k=0;k<200;k++){ // the time loop
fro = pow(r1,0.6)/1.4; eul;
u1=u;r1=abs(r);
v1=v*(y<1.4)+abs(v)*(y>=1.4);
// to avoid reflexion from top
plot(r,wait=0,value=1);
}
```