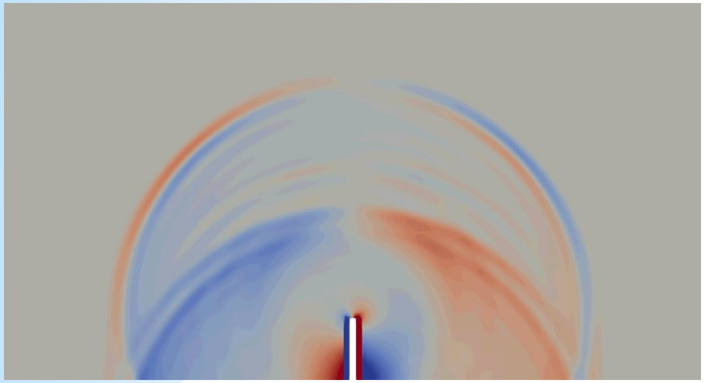


Mathematical modelling of unsteady problems of mechanics of continua using the CABARET method in OpenFOAM framework

V.M. Goloviznin, M.A. Zaitsev, S.A. Karabasov

IBRAE

Mathematical modelling of unsteady problems of mechanics of continua using the CABARET method in OpenFOAM framework

Contents:

- Mathematical modelling of unsteady problems using the CABARET(case for elastic media)
- Development of unified algorithms of fluid-structure interaction problem solution including acoustics applictions
- Calculation results including hybrid hexa & tetra mesh

Problem solved:

- unsteady backstep flow
- unsteady t-junction flow
- unsteady jet flow of mixing multicomponent gas
- unsteady acoustic emission of oscillating beam

Mathematical modelling of unsteady problems using the CABARET(case for elastic media) momentum equation

$$\rho\frac{\partial u}{\partial t}=\frac{\partial\sigma_{xx}}{\partial x}+\frac{\partial\sigma_{xy}}{\partial y}+\frac{\partial\sigma_{xz}}{\partial z}$$

$$\rho\frac{\partial v}{\partial t}=\frac{\partial\sigma_{xy}}{\partial x}+\frac{\partial\sigma_{yy}}{\partial y}+\frac{\partial\sigma_{yz}}{\partial z}$$

$$\rho\frac{\partial w}{\partial t}=\frac{\partial\sigma_{xz}}{\partial x}+\frac{\partial\sigma_{yz}}{\partial y}+\frac{\partial\sigma_{zz}}{\partial z}$$

ρ - density; u, v, w velocity components; x, y, z coordinates; σ_{ij}-componets of Cauchy stress tensor.

OpenFOAM formulation for time step dt2:
u=u-dt2*fvc::surfaceIntegrate(ss & mesh.Sf())/Rofon;
ss - stress tensor defined at faces(surfaceTensorField ss);
Rofon – density
mesh.Sf() – face vector

Equation of state

$$\frac{\partial \sigma_{xx}}{\partial t} = 2\mu \frac{\partial u}{\partial x} + \lambda\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right)$$
$$\frac{\partial \sigma_{yy}}{\partial t} = 2\mu \frac{\partial v}{\partial y} + \lambda\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right)$$
$$\frac{\partial \sigma_{zz}}{\partial t} = 2\mu \frac{\partial w}{\partial z} + \lambda\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right)$$
$$\frac{\partial \sigma_{xy}}{\partial t} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)$$
$$\frac{\partial \sigma_{xz}}{\partial t} = \mu\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right)$$
$$\frac{\partial \sigma_{yz}}{\partial t} = \mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right)$$

λ,μ - Lame constans of elastic media;

OpenFOAM formulation for time step dt2:

```
volTensorField gradU = fvc::surfaceIntegrate(us*mesh.Sf());
            s=s-dt2*(2.0*mu*symm(gradU)+lambda*I*tr(gradU));
```

s – stress tensor in cells;

us - velocity vector in faces(surfaceVectorField us);

lambda, mu – Lame constans

Eigen values

Plane problem in X direction

$$\frac{\partial}{\partial t}\begin{Bmatrix} u \\ v \\ \sigma_{xx} \\ \sigma_{xy} \\ \sigma_{yy} \end{Bmatrix} + \begin{bmatrix} 0 & 0 & -\frac{1}{\rho} & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{\rho} & 0 \\ -\lambda-2\mu & 0 & 0 & 0 & 0 \\ 0 & -\mu & 0 & 0 & 0 \\ -\lambda & 0 & 0 & 0 & 0 \end{bmatrix} \frac{\partial}{\partial x}\begin{Bmatrix} u \\ v \\ \sigma_{xx} \\ \sigma_{xy} \\ \sigma_{yy} \end{Bmatrix} + \begin{bmatrix} 0 & 0 & 0 & -\frac{1}{\rho} & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{\rho} \\ 0 & -\lambda & 0 & 0 & 0 \\ -\mu & 0 & 0 & 0 & 0 \\ 0 & -\lambda-2\mu & 0 & 0 & 0 \end{bmatrix} \frac{\partial}{\partial y}\begin{Bmatrix} u \\ v \\ \sigma_{xx} \\ \sigma_{xy} \\ \sigma_{yy} \end{Bmatrix} = 0$$

$$\det\begin{bmatrix} 0-\Lambda & 0 & -\frac{1}{\rho} & 0 & 0 \\ 0 & 0-\Lambda & 0 & -\frac{1}{\rho} & 0 \\ -\lambda-2\mu & 0 & 0-\Lambda & 0 & 0 \\ 0 & -\mu & 0 & 0-\Lambda & 0 \\ -\lambda & 0 & 0 & 0 & 0-\Lambda \end{bmatrix} = 0$$

$$\Lambda^5 - \Lambda^3 \frac{1}{\rho}(\lambda+3\mu) + \Lambda \frac{1}{\rho^2}(\lambda+2\mu)\mu = 0$$

Invariants

$$I_+ = u - \frac{\sigma_{xx}}{\rho c_1}$$

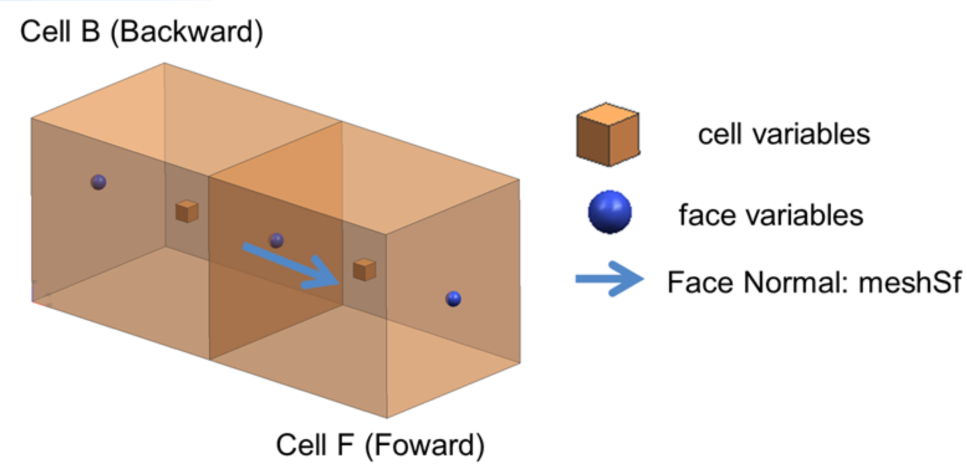
$$I_- = u + \frac{\sigma_{xx}}{\rho c_1}$$

$$J_+ = v - \frac{\sigma_{xy}}{\rho c_2}$$

$$J_- = v + \frac{\sigma_{xy}}{\rho c_2}$$

Eigen values are equal positive and negative values of longitudinal and transverse wave velocity. Zero eigen value is for Y-direction stress invariant.

Space and time stencils



OpenFOAM formulation:
Time loop
Phase 1;
Phase 2;
Phase 3;
Loop end
Phase 2 is external function.
Boundaries are OpenFOAM codedMixed type.

Cell computations(Phase 1 & 3)

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix}_{new} = \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} + \frac{\Delta t}{2\rho V} \sum_{k=1}^{6} \begin{Bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_{zz} \end{Bmatrix} \begin{Bmatrix} S_x \\ S_y \\ S_z \end{Bmatrix}$$

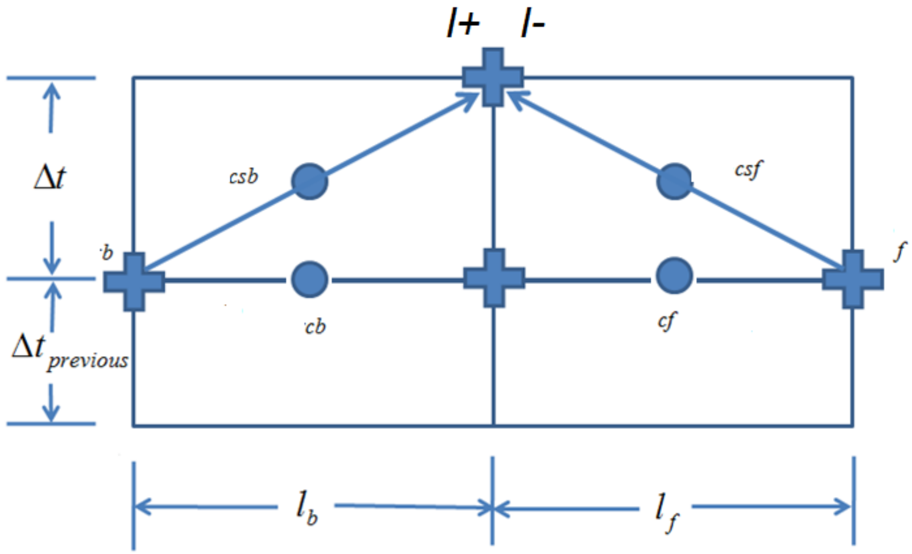
$$\begin{Bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} & \frac{\partial w}{\partial z} \end{Bmatrix} = \frac{1}{V} \sum_{k=1}^{6} \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} \begin{Bmatrix} S_x & S_y & S_z \end{Bmatrix}$$

$$\Delta = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}$$

$$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{xy} \\ \sigma_{xz} \\ \sigma_{yy} \\ \sigma_{yz} \\ \sigma_{zz} \end{Bmatrix}_{new} = \begin{Bmatrix} \sigma_{xx} \\ \sigma_{xy} \\ \sigma_{xz} \\ \sigma_{yy} \\ \sigma_{yz} \\ \sigma_{zz} \end{Bmatrix} + \frac{\Delta t}{2} \begin{Bmatrix} \lambda\Delta + 2\mu \frac{\partial u}{\partial x} \\ \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \\ \mu\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) \\ \lambda\Delta + 2\mu \frac{\partial v}{\partial y} \\ \mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right) \\ \lambda\Delta + 2\mu \frac{\partial w}{\partial z} \end{Bmatrix}$$

V – cell volume, Δt – time step.

Face computations



c – invariant value, l – distance to opposite face, indices b и f are for backward and forward face invariant value, cb и cf are for backward and forward cell invariant value, csb и csf are for backward and forward cell invariant value on intermediate time step.

$$I_+^{\max} = \max(I, I_b, I_{cb}) + 2(I_{csb} - I_{cb}) + c\frac{\Delta t}{l}(I - I_b)$$

$$I_+^{\min} = \min(I, I_b, I_{cb}) + 2(I_{csb} - I_{cb}) + c\frac{\Delta t}{l}(I - I_b)$$

$$I_+^{new} = \begin{cases} I_+^{\max} & 2I_{csb} - I_b > I_+^{\max} \\ 2I_{csb} - I_b & I_+^{\min} <= 2I_{csb} - I_b <= I_+^{\max} \\ I_+^{\min} & 2I_{csb} - I_b < I_+^{\min} \end{cases}$$

$$I_-^{\max} = \max(I, I_f, I_{cf}) + 2(I_{csf} - I_{cf}) + c\frac{\Delta t}{l}(I_f - I)$$

$$I_-^{\min} = \min(I, I_f, I_{cf}) + 2(I_{csf} - I_{cf}) + c\frac{\Delta t}{l}(I_f - I)$$

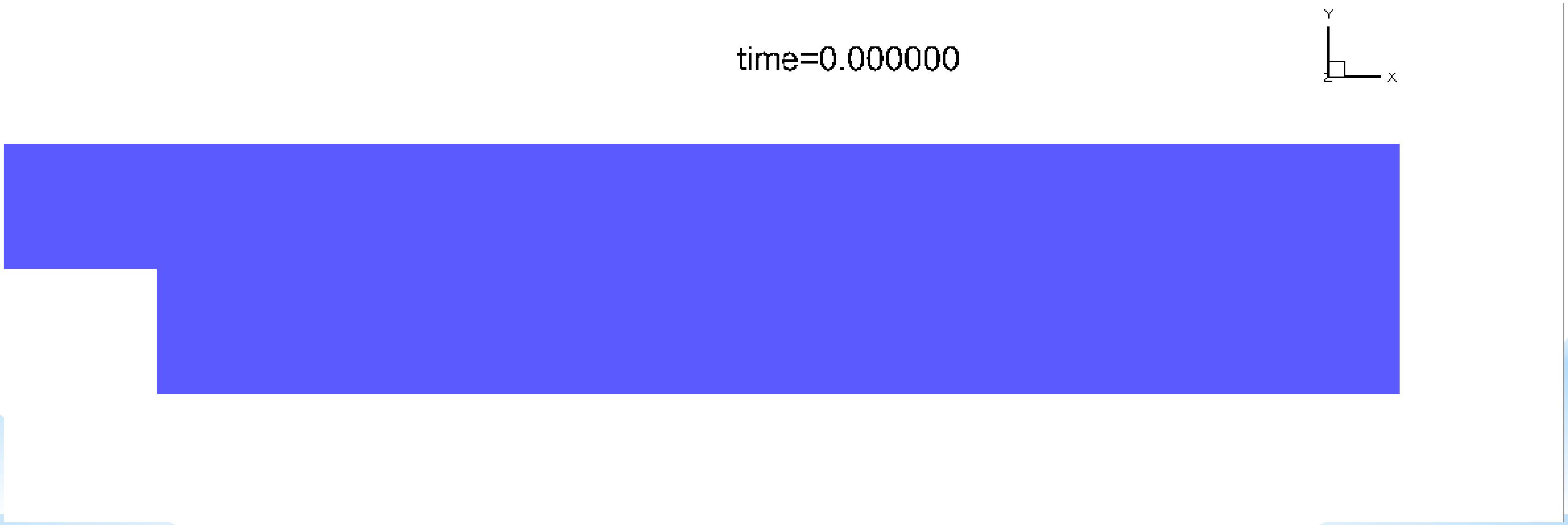
$$I_-^{new} = \begin{cases} I_-^{\max} & 2I_{csb} - I_b > I_-^{\max} \\ 2I_{csf} - I_f & I_-^{\min} <= 2I_{csb} - I_b <= I_-^{\max} \\ I_-^{\min} & 2I_{csb} - I_b < I_-^{\min} \end{cases}$$

New velocity values –half summ of invariants with indices “+” and “-”. Stresses – half difference, multiplied by factor ρc.

Unsteady backstep flow problem

Linear compressible fluid

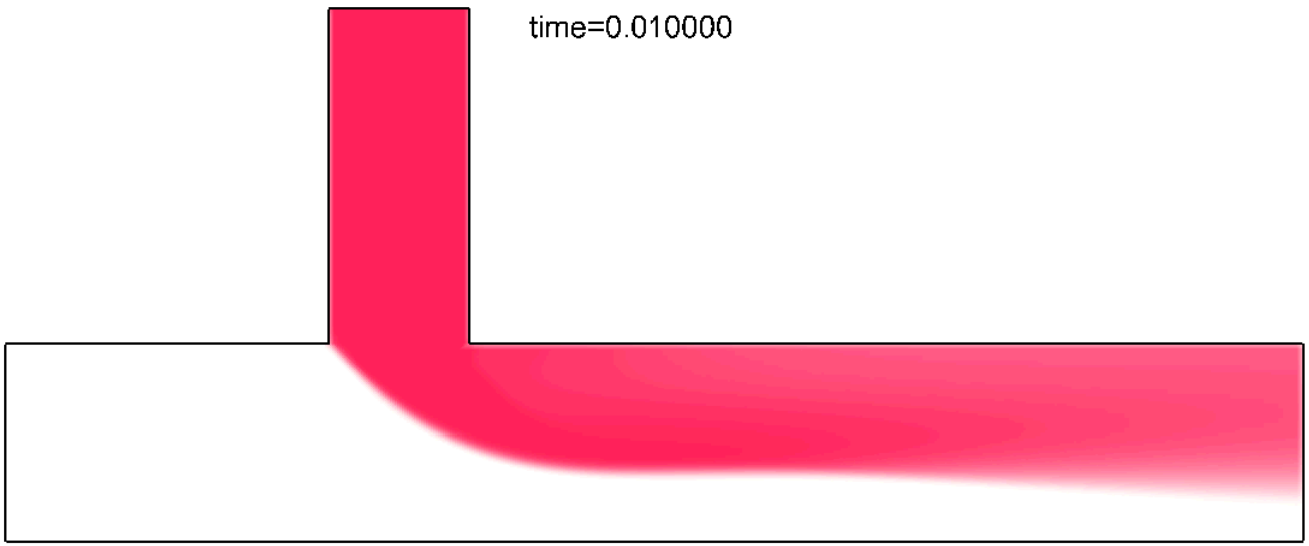
```
p=p-dt2*rss*fvc::surfaceIntegrate(mesh.Sf() & us);
u=u-dt2*fvc::surfaceIntegrate((mesh.Sf() & us)*us+ps*mesh.Sf()/Rofon)
    +dt2*fvc::laplacian(nu, u)+g*dt2*t;
```



Unsteady T-junktion problem

Linear compressible fluid

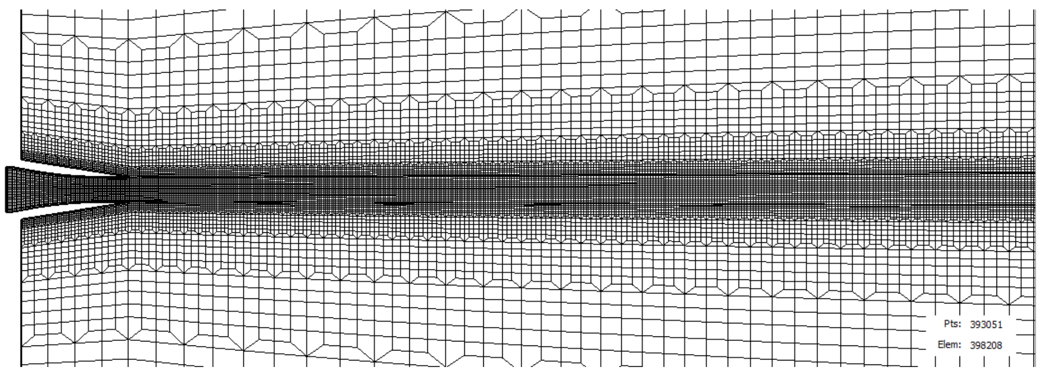
```
         p=p-dt2*rss*fvc::surfaceIntegrate(mesh.Sf() & us);
         t=t-dt2*fvc::surfaceIntegrate((mesh.Sf() & us)*ts)
        +dt2*fvc::laplacian(kappa, t);
     u=u-dt2*fvc::surfaceIntegrate((mesh.Sf() & us)*us+ps*mesh.Sf()/Rofon)
        +dt2*fvc::laplacian(nu, u)+g*dt2*t;
```



Unsteady jet problem

Ideal gas

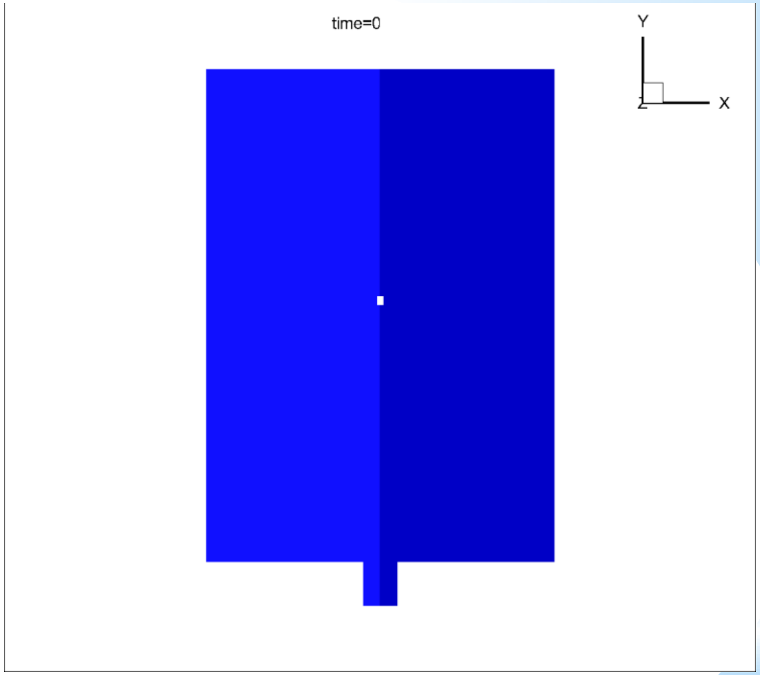
```
r=r-dt2*fvc::surfaceIntegrate(rnews*(mesh.Sf() & unews));
 e=(e*r-dt2*fvc::surfaceIntegrate((mesh.Sf() &
unews)*(rnews*Cv*tnews+pnews)))/r
u=(u*r-dt2*fvc::surfaceIntegrate((mesh.Sf() &
unews)*unews*rnews+pnews*mesh.Sf()))/r
 +dt2*fvc::laplacian(nu, u);
```



Unsteady jet flow of mixing multicomponent gas problem

Multicomponent ideal gas

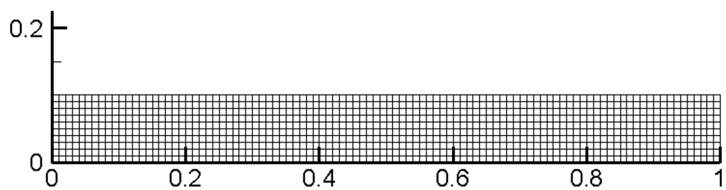
```
rnew=r-dt2*fvc::surfaceIntegrate(rnews*(mesh.Sf() & unews));
hnew=h-dt2*fvc::surfaceIntegrate(hnews*(mesh.Sf() & unews));
wnew=w-dt2*fvc::surfaceIntegrate(wnews*(mesh.Sf() & unews));
unews)*rnews*enews))/rnew
enew=(e*r-dt2*fvc::surfaceIntegrate((mesh.Sf() & unews)*(((rnews-hs-ws)*Cva+Cvh*hs+Cvw*w
dt2*fvc::laplacian(kappa, ((r-h-w)*Cva+Cvh*h+Cvw*w)*t))/r;
unew=(u*r-dt2*fvc::surfaceIntegrate((mesh.Sf() &
unews)*unews*rnews+pnews*mesh.Sf()))/rnew
 +dt2*fvc::laplacian(nu, u)+g*dt2;
pnew=((Cph-Cvh)*hnew+(Cpw-Cvw)*wnew+(Cpa-Cva)*(rnew-hnew-wnew))/
 (Cvh*hnew+Cvw*wnew+Cva*(rnew-hnew-wnew))
*rnew*(enew-mag(unew)*mag(unew)/2);
<------>tnew=pnew/((Cph-Cvh)*hnew+(Cpw-Cvw)*wnew+(Cpa-Cva)*(rnew-hnew-wnew));
```

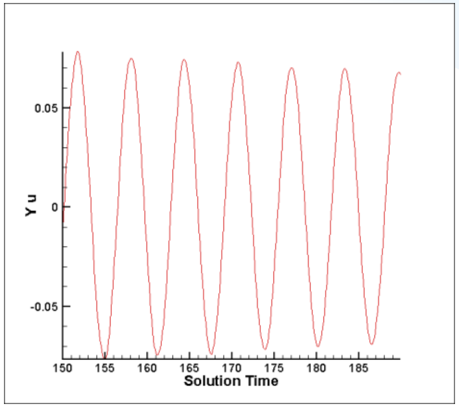


Beam vibration problem

Elastic media

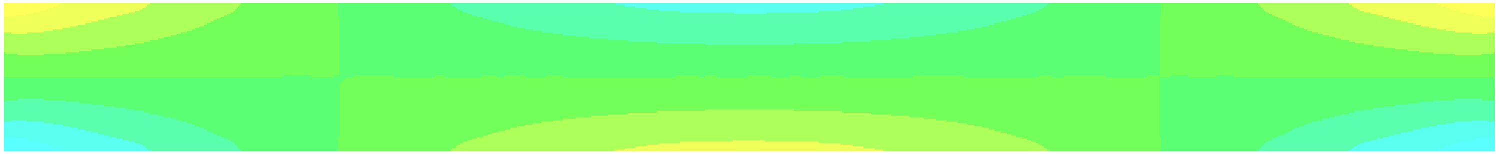
```
volTensorField gradU = fvc::surfaceIntegrate(us*mesh.Sf());
        s=s-dt2*(2.0*mu*symm(gradU)+lambda*I*tr(gradU));
        u=u-dt2*fvc::surfaceIntegrate(ss & mesh.Sf())/Rofon;
```



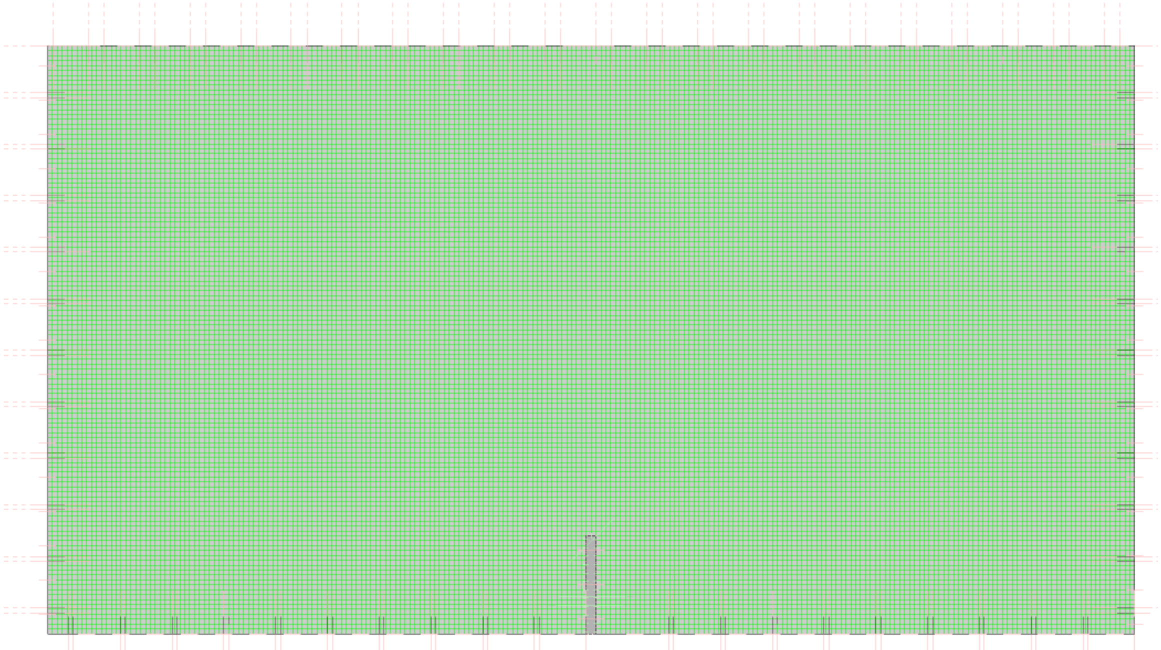


Left and right ends of beam are fully constrained. Uniform pressure is applied at top surface.

time=190.000000



Unsteady acoustic emission of oscillating beam



Beam has uniform horizontal velocity of value 0.1 m/sec at initial time.

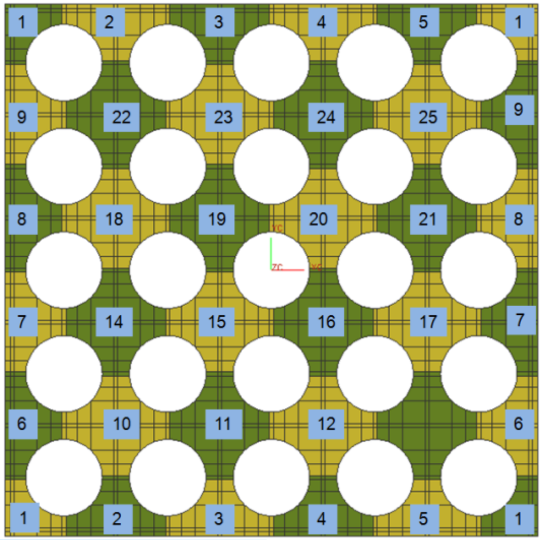
```
├─0
│ ├─fluid
│ │ └─polyMesh
│ └─solid
│   └─polyMesh
├─constant
│ ├─fluid
│ └─solid
├─dynamicCode
│ ├─fixedDisplacementStress
│ │ ├─lnInclude
│ │ └─Make
│ │   ├─cygwin64mingw-w64DPInt32Opt
│ │   └─linux64GccDPInt32Opt
│ ├─fluidPressure
│ │ ├─lnInclude
│ │ └─Make
│ │   ├─cygwin64mingw-w64DPInt32Opt
│ │   └─linux64GccDPInt32Opt
│ ├─fluidVelocity
│ │ ├─lnInclude
│ │ └─Make
│ │   ├─cygwin64mingw-w64DPInt32Opt
│ │   └─linux64GccDPInt32Opt
│ ├─platforms
│ │ ├─cygwin64mingw-w64DPInt32Opt
│ │ │ └─lib
│ │ └─linux64GccDPInt32Opt
│ │   └─lib
│ ├─solidStress
│ │ ├─lnInclude
│ │ └─Make
│ │   ├─cygwin64mingw-w64DPInt32Opt
│ │   └─linux64GccDPInt32Opt
│ └─solidVelocity
│   ├─lnInclude
│   └─Make
│     ├─cygwin64mingw-w64DPInt32Opt
│     └─linux64GccDPInt32Opt
├─ifoam_201709100600
│ └─Make
│   └─linux64GccDPInt32Opt
└─system
  ├─fluid
  └─solid
```

OpenFOAM data structure

Pressure animation during acoustic emission

Parallel computations and scalability

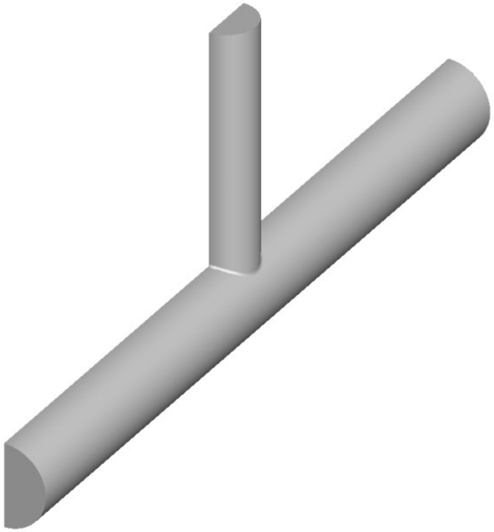
Program	Program name	Problem	Processor core number	Cell number	Operating speed
Linear compressible fluid	rhoCabaretFoam	t-junction	64	231517	9.95243373
Ideal gas	gasCabaretFoam	jet flow	32	400000	16.7
Elastic media	solidCabaretFoam	beam vibration	1	1000	7.5
Multicomponent gas	mixingCabaretFoa m	multicomponent jet	32	9588	50
Fluid structure interaction	fsiCabaretFoam	acoustic emission of oscillating beam	1	26600	34.54

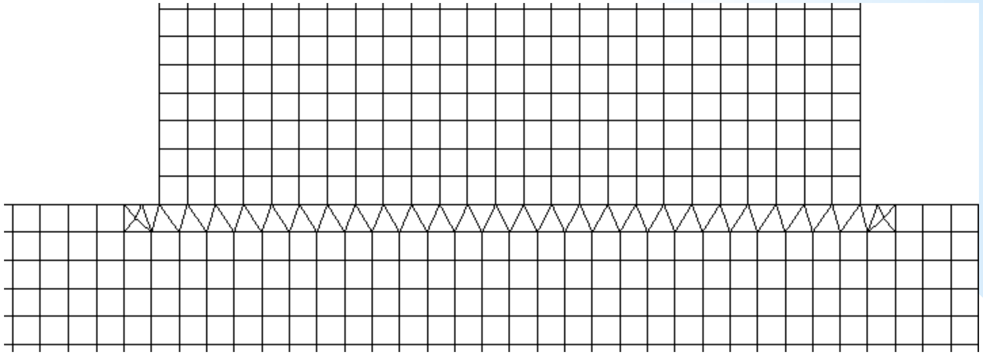


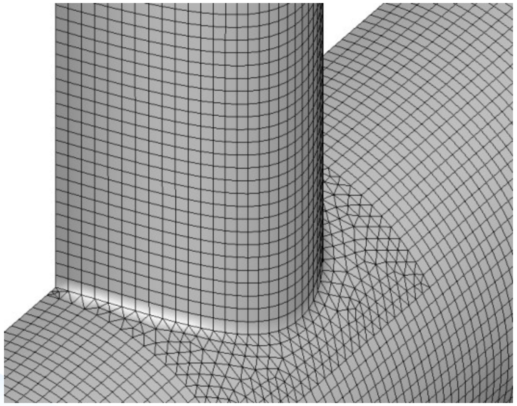
V.M. Goloviznin, M.A. Zaitsev and S.A. Karabasov, "A highly scalable hybrid mesh CABARET MILES method for MATIS-H problem," Proc. of the CFD4NRS-4 WORKSHOP, p. 104, Daejon, South Korea (2012).

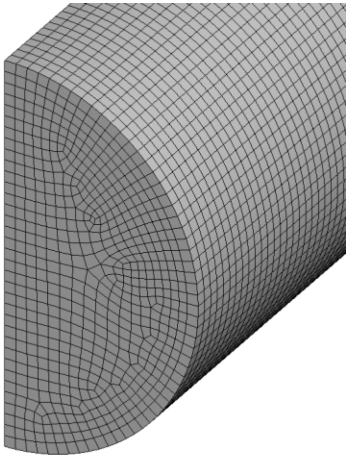
Parallel computations were made on mesh up to 40 millions cells on 4096 processors cores.

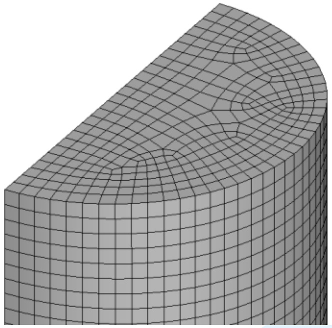
Hybrid hexa & tetra mesh for t-junction problem











Hybrid hexa & tetra mesh space stencil

Opposite face value is only for hexa cell. To make uniform computations it needs to define fictive opposite face value.

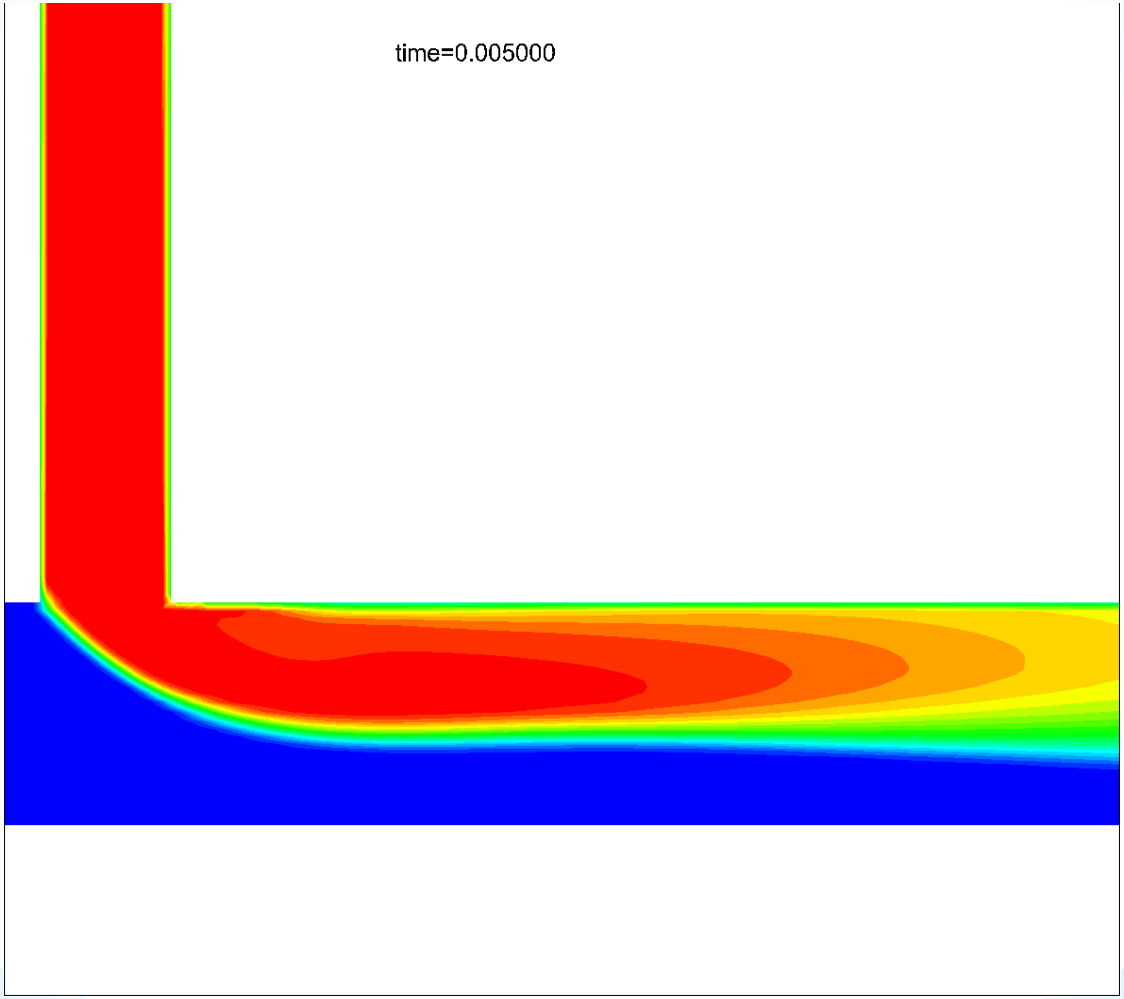
1) $$\frac{\varphi_c^{n+1/2} - \varphi_c^n}{\tau / 2} + c\,\frac{\varphi_R^n - \varphi_L^n}{h} = 0$$

2) $$\varphi_R^{n+1} = 2\varphi_c^{n+1/2} - \left(\left(\varphi_L^n + 4\varphi_c^n + \varphi_R^n\right)/3 - \varphi_R^n\right)$$

3) $$\frac{\varphi_c^{n+1} - \varphi_c^{n+1/2}}{\tau / 2} + c\,\frac{\varphi_R^{n+1} - \varphi_L^{n+1}}{h} = 0$$

Gorbachev D.J., “Adaptation of Cabaret method for abitrary mesh cells”, //Scientific conference “Tichonovskie chtenija”, Moscow State University, Moscow, Oct. 2017.

Hybrid hexa & tetra mesh space stencil temperature animation t-junction results



Conclusions

1. CABARET method realization in OpenFOAM framework for linear compressible fluid, ideal gas and elastic media have the same unique features that differentiate it from the another realizations.
2. CABARET method realization in OpenFOAM framework for linear compressible fluid structure interaction allows to create new uniform numerical algorithm with high quality computations including interface regions.
3. Operating speed in OpenFOAM framework is compatible with the same FORTRAN version of programs. Parallel numerical algorithms have parallel cluster scalability.

Thanks !