Fire Dynamics Simulator: Advances on simulation capability for complex geometry

Marcos Vanella[a,b], Randall McDermott[b], Glenn Forney[b], Kevin McGrattan[b]

[a] *The George Washington University*
[b] *National Institute of Standards and Technology*

Thunderhead Engineering Fire and Evacuation Modelling Technical Conference 2016. Torremolinos, Spain. November 16th-18th, 2016.

Contents

Motivation, Objective

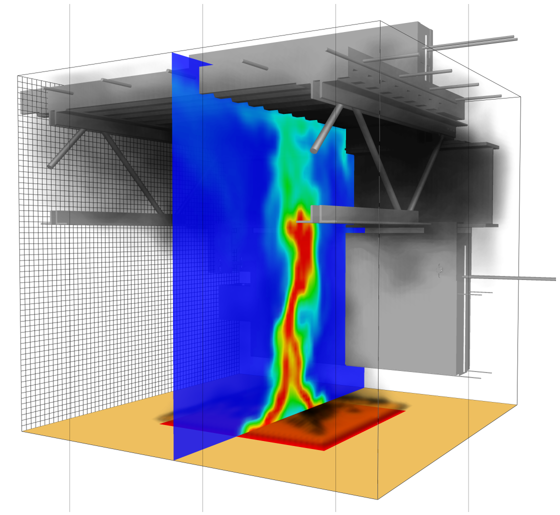
The Fire Dynamics Simulator* (FDS) is used in:

- performance-based design of fire protection systems,
- forensic work,
- Simulation of wild land fire scenarios.

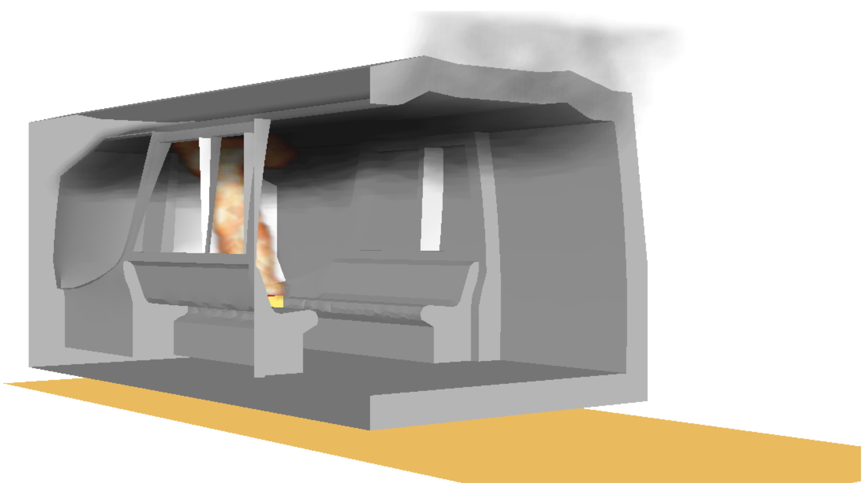
Uses block-wise structured, rectilinear grids for gas phase, and “lego-block” geometries to represent internal boundaries.

Objective:

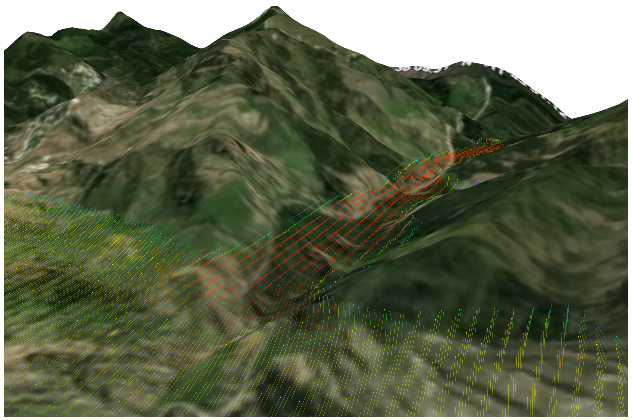
- Develop an efficient, conservative numerical scheme for treatment of complex geometry within FDS.



Fire-Structure Interaction: 12 MW fire load on a steel/concrete floor connection assembly.



LES of 800 KW propane fire in open train cart. Geometry courtesy of Fabian Braennstroem (Bombardier).



Velocity vectors (35 m/s [78 mph] max [red]) for a wind field in Mill Creek Canyon, Utah. 4 km x 4 km horizontal domain, 1 km vertical. 40 m grid resolution on a single mesh.

* K. McGrattan et al. Fire Dynamics Simulator, Tech. Ref. Guide, NIST. Sixth Ed., Sept. (2013).

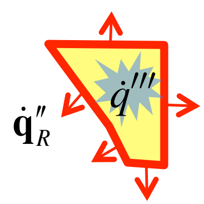
Motivation, Objective

Spatial discretization and time marching in FDS, work areas:

Scalar transport
$$\left\{ \frac{\partial \rho Y_\alpha}{\partial t} + \nabla\cdot\left(\rho Y_\alpha \mathbf{u}\right) = -\nabla\cdot\left(-\rho D_\alpha \nabla Y_\alpha\right) + \dot{m}_\alpha''' \longrightarrow \rho^{n+1}, Y_\alpha^{n+1} \right.$$

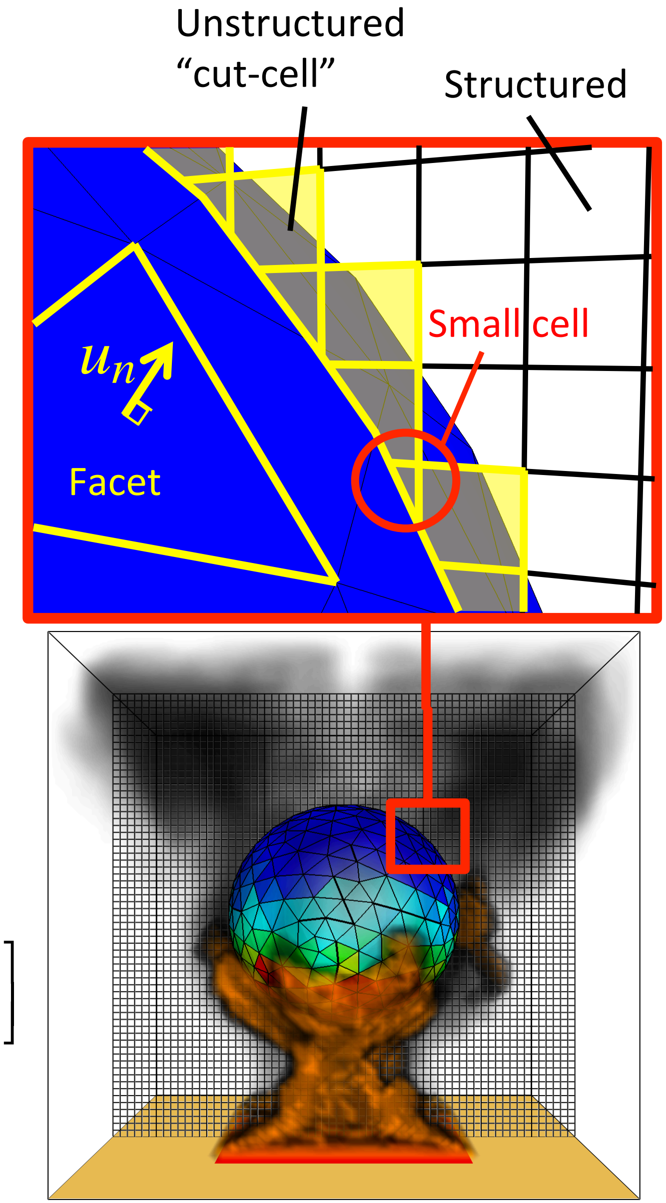
EOS
$$\left\{ \bar{W}^{n+1} = \left[\sum_{\alpha=1}^{n_\alpha} Y_\alpha^{n+1} / W_\alpha\right]^{-1}, \quad T^{n+1} = \frac{\bar{p}\bar{W}^{n+1}}{\rho^{n+1} R} \longrightarrow T^{n+1} \right.$$

Combustion, Radiation



Divergence Constraint*
$$\left\{ \nabla\cdot\mathbf{u} = \frac{1}{\rho h_s}\left[\frac{D}{Dt}\left(\bar{p} - \rho h_s\right) + \dot{q}''' - \nabla\cdot\dot{\mathbf{q}}''\right] \longrightarrow \left(\nabla\cdot\mathbf{u}\right)^{n+1} \right.$$

Momentum + IBM+
$$\left\{ \begin{array}{l} \dfrac{\partial \mathbf{u}}{\partial t} = -\left(\mathbf{F} + \nabla H^{n-1}\right) \longrightarrow \mathbf{F}_{IB} = -\left(\dfrac{\partial \mathbf{u}}{\partial t}\right)_D - \nabla H^{n-1} \\ \dfrac{\partial}{\partial t}\left(\nabla\cdot\mathbf{u}\right) \cong \dfrac{\left(\nabla\cdot\mathbf{u}\right)^{n+1} - \nabla\cdot\mathbf{u}^n}{\Delta t}, \quad \Delta H = -\left[\nabla\cdot\mathbf{F} + \dfrac{\partial}{\partial t}\left(\nabla\cdot\mathbf{u}\right)\right] \\ \dfrac{\partial \mathbf{u}}{\partial t} = -\left(\mathbf{F} + \nabla H\right) \longrightarrow \mathbf{u}^{n+1} \quad \Delta H^n \end{array} \right.$$



* R. J. McDermott. J. Comput. Phys. 274, pp. 413-431 (2014); + E. A. Fadlun et al. J. Comput. Phys. 161, pp. 35-60 (2000).

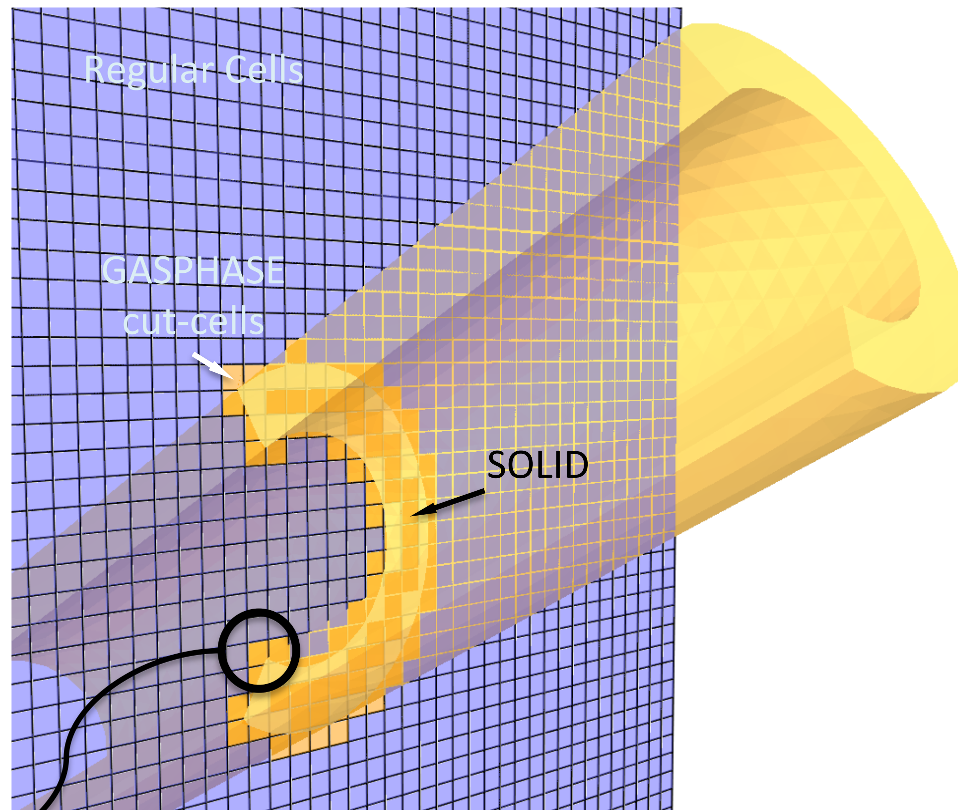
Computational Geometry

Objective:

- Define cut-cell volumes of Cartesian cells intersected by body.
- Robust, general, parallelizable.
- Ideally efficient for moving object problem.

Data Management:

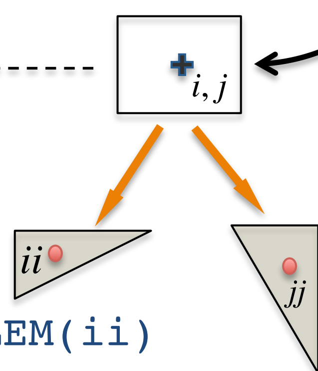
- Work by Eulerian mesh block. Body surfaces defined by triangulations.
- Hierarchical data structures are defined, capable of arbitrary number of cut-faces and cut-cells per Cartesian counterparts.



Smokeview Visualization inclined C-beam mid-plane. Obtained with computational geometry engine in FDS.

- Cartesian Level ----------------

`IBM_CUTCELL(m)`



- Cut-cell Level ---------

`IBM_CUTCELL(m)%CCELEM(ii)`

Scheme:

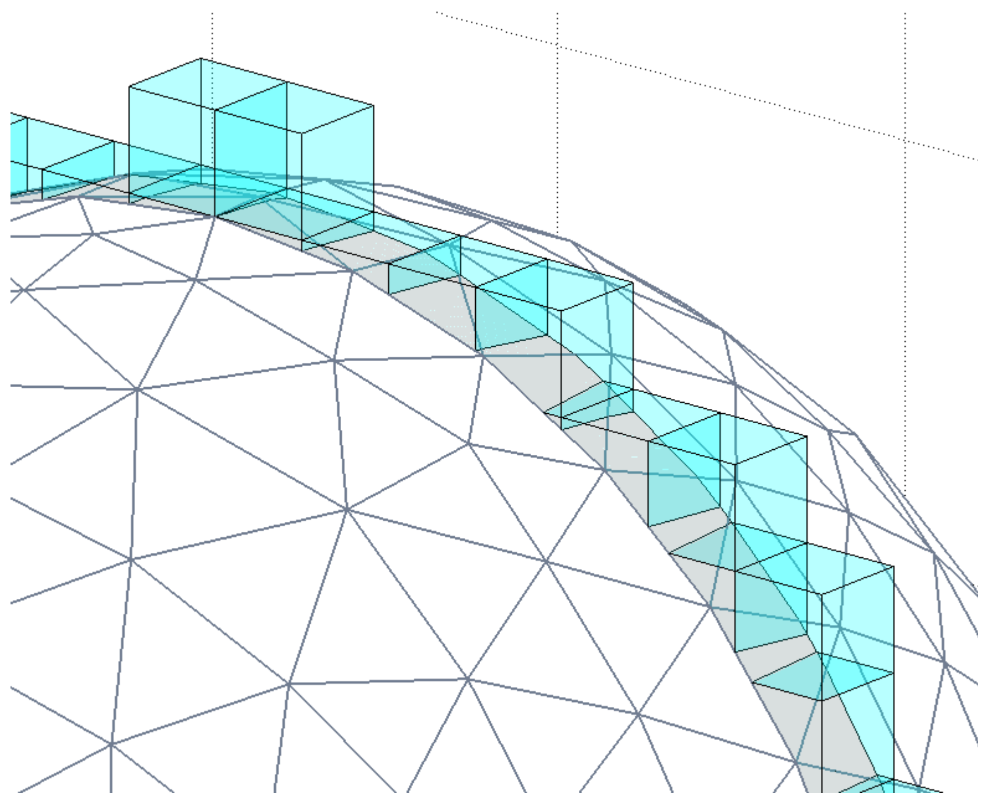
- Body-plane intersection elements (segments, triangles) are defined for all Cartesian grid planes. Intersections along surface triangles also defined.

- Cut-faces on Cartesian planes are defined by joining segments. Same for cut-faces along triangles.

- Working by Cartesian cell, cut face sets are found for each **cut-cell** volume.

- Area and volume properties are computed for each cut-face and cell.

- Interpolation stencils are found for centroids (IBM).



Cut-cell definition on original Matlab implementation.

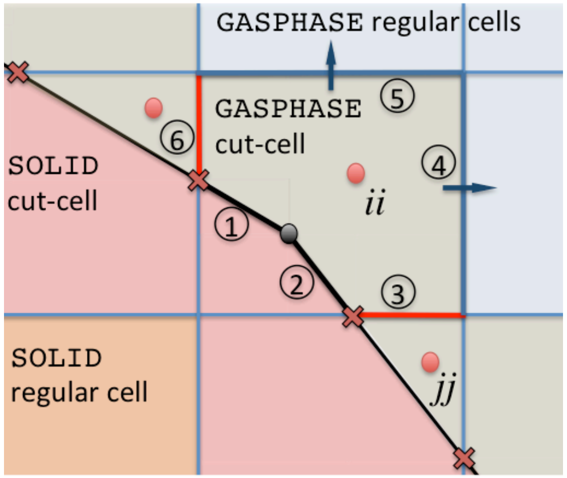
Scalar Transport

Based on mass fractions: $\frac{\partial \rho Y_\alpha}{\partial t} + \nabla \cdot (\rho Y_\alpha \mathbf{u}) = -\nabla \cdot \mathbf{J}_\alpha + \dot{m}_\alpha''' \; ; \; \alpha = 1,\ldots,N$ on domain + Ics, Bcs

Take: $\mathbf{J}_\alpha = -\rho D_\alpha \nabla Y_\alpha = -\left(D_\alpha \nabla(\rho Y_\alpha) - \frac{D_\alpha}{\rho} \nabla \rho \; (\rho Y_\alpha) \right)$

Then:

$$\frac{\partial \rho Y_\alpha}{\partial t} + \nabla \cdot (\mathbf{u}' \rho Y_\alpha) = \nabla \cdot (D_\alpha \nabla(\rho Y_\alpha)) + \dot{m}_\alpha''' \; ; \; \mathbf{u}' = \mathbf{u} + \frac{D_\alpha}{\rho} \nabla \rho$$



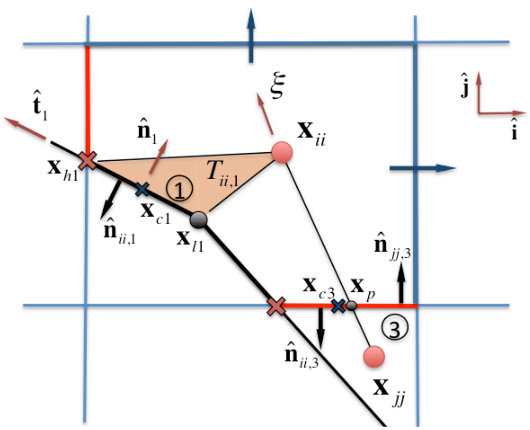
Finite Volume method: Divide the domain on $ii \rightarrow (i,j) \in \Im$ cells.

- **Advection:**

$$\int_{\Omega_{ii}} \nabla \cdot (\mathbf{u}' \rho Y_\alpha) d\Omega = \int_{\partial \Omega_{ii}} (\mathbf{u}' \rho Y_\alpha) \cdot \hat{\mathbf{n}}_{ii} d\partial\Omega = \sum_{k=1}^{nfc} (\mathbf{u}' \rho Y_\alpha)_k \cdot \hat{\mathbf{n}}_{ii,k} A_k$$

For a given face (k=4, cut-cell *ii*):

$$(\mathbf{u}' \rho Y_\alpha)_k \cdot \hat{\mathbf{n}}_{ii,k} A_k = \left[\overline{(\rho Y_\alpha)}_k^{fl} \mathbf{u}_k + \overline{(\rho Y_\alpha)}_k^{lin} \left(\frac{D_\alpha}{\rho} \nabla \rho \right)_k \right] \cdot \hat{\mathbf{n}}_{ii,k} A_k$$

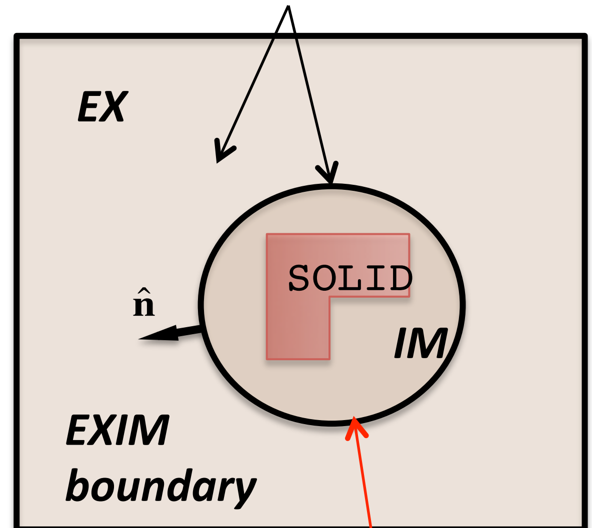


- **Diffusion:** $\int_{\Omega_{ii}} \nabla \cdot (D_\alpha \nabla(\rho Y_\alpha)) d\Omega = \int_{\partial \Omega_{ii}} (D_\alpha \nabla(\rho Y_\alpha)) \cdot \hat{\mathbf{n}}_{ii} d\partial\Omega = \sum_{k=1}^{nfc} (D_\alpha \nabla(\rho Y_\alpha))_k \cdot \hat{\mathbf{n}}_{ii,k} A_k$

Scalar Transport

$$\mathbf{u}'^{n}(\rho Y_\alpha)^n - D_\alpha^n \nabla(\rho Y_\alpha)^n$$

- Small cut-cells are problematic for explicit time integration.
- Alleviation methods tend to be arbitrary, deteriorating the solution quality.



$$\mathbf{u}'^{n}(\rho Y_\alpha)^{n+1} - D_\alpha^n \nabla(\rho Y_\alpha)^{n+1}$$

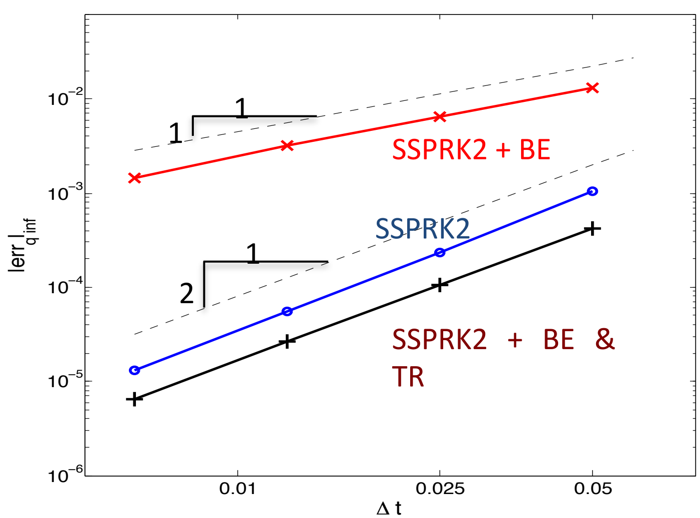
Explicit - Implicit time integration*:

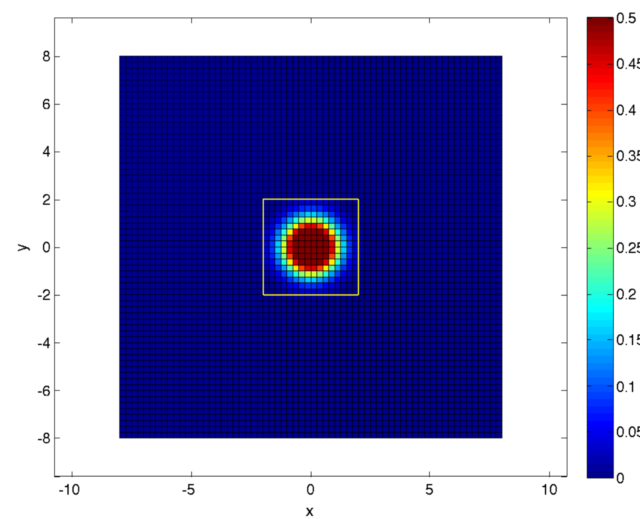
- **Explicit region**: Advance first.
- **Implicit region**: linearizing transport, i.e. implicit BE:

$$\frac{(\rho Y_\alpha)^{n+1} - (\rho Y_\alpha)^n}{\Delta t} = -\nabla \cdot \left(\boxed{\mathbf{u}'^{n}(\rho Y_\alpha)^{n+1} - D_\alpha^n \nabla(\rho Y_\alpha)^{n+1}} \right)$$

*- C.N. Dawson, T.F. Dupont. SIAM J. Numer. Analysis 31:4, pp. 1045-1061 (1994).

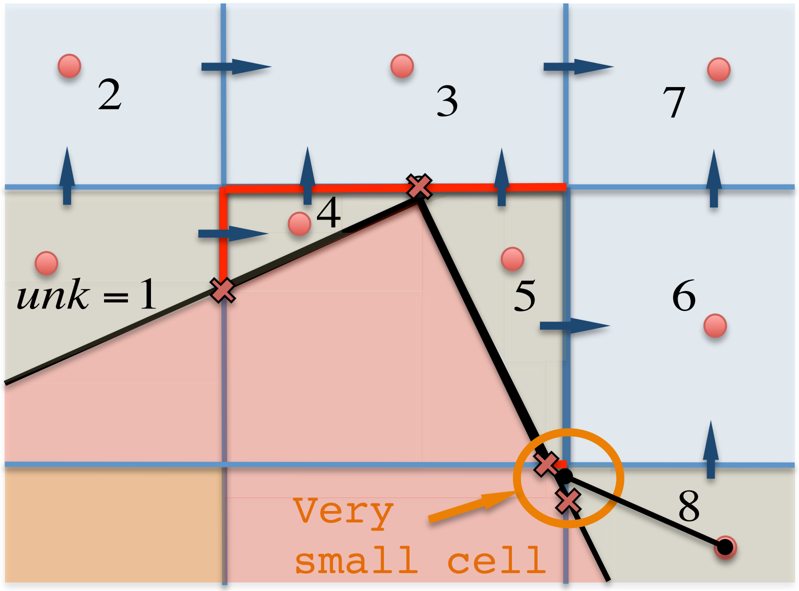
- S. May, M. Berger. Proc. Finite Vol. Cmplx App. VII, pp. 393-400 (2014).





Scalar Transport

- **Number** cell centered **unknowns** for $(\rho Y_\alpha)^{n+1}$.
- Build **face lists** on implicit region (cut-face and regular, GASPHASE or INBOUNDARY).
- Advection diffusion **matrices are built** by face. End result in CSR format.
- The corresponding discretized matrix-vector system:



Implicit (BE):

$$\left[\mathbf{M} + \Delta t\left(\mathbf{A}_{adv} + \mathbf{A}_{diff}\right)\right]\left\{\rho Y_\alpha\right\}^{n+1} = \mathbf{M}\left\{\rho Y_\alpha\right\}^{n} + \Delta t\left\{f\right\}$$

- Implicit: Solve using the **Intel MKL Pardiso**. Explicit: Trivial as **M** is diagonal.
- Very small cells cause **ill conditioned** systems. **Link** small cells to neighbors when $Vol_{CC} < C_{link} Vol_{Cart}$ and $C_{link} \approx 10^{-4}$.
- Fully explicit option (FE): $\left[\mathbf{M}\right]\left\{\rho Y_\alpha\right\}^{n+1} = \left[\mathbf{M} - \Delta t\left(\mathbf{A}_{adv} + \mathbf{A}_{diff}\right)\right]\left\{\rho Y_\alpha\right\}^{n} + \Delta t\left\{f\right\}$ $\quad C_{link} \approx 0.95$

Energy

We factor the velocity divergence from the sensible enthalpy evolution equation (FDS).

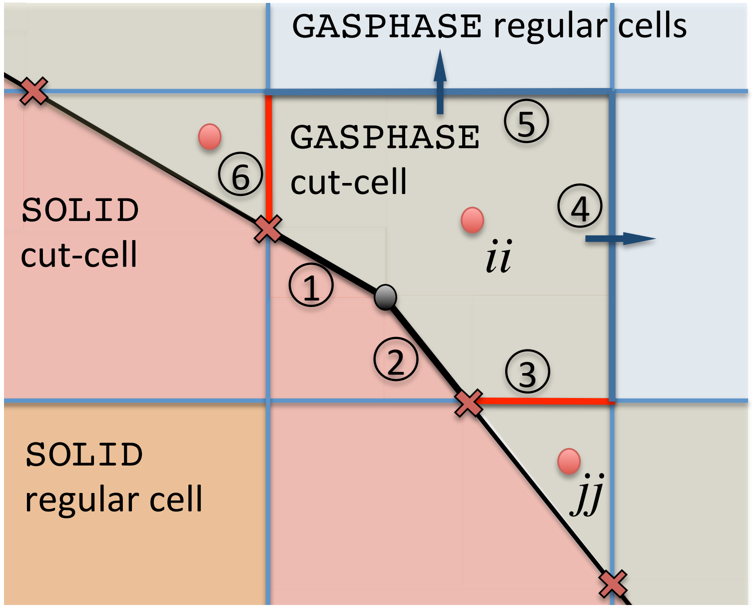
Objective:

- Discretize terms in thermodynamic divergence consistently with the scalar transport formulation for cut-cells (unstructured finite volume mesh).
- Use divergence integral equivalence to relate this divergence to the FDS Cartesian mesh.

Our Scheme:

- Implemented transport terms in cut-cells.
- Added combustion in regular cells of cut-cell region, radiation next.
- Linked cells for scalar transport get volume averaged thermodynamic divergence.

$$
\begin{aligned}
(\nabla\cdot\mathbf{u})^{th}_{ii}\,V_{ii} \;=\; & \left[\frac{1}{(\rho c_p T)_{ii}} - \frac{1}{\bar{p}_{ii}}\right]\frac{\partial \bar{p}_{ii}}{\partial t}\,V_{ii} + \frac{w_{ii}\,\rho_0 g_z}{(\rho c_p T)_{ii}} \\
& + \frac{1}{(\rho c_p T)_{ii}}\left[\dot{q}'''V_{ii} - \sum_{k=1}^{nf_c}\dot{\mathbf{q}}''_{ii,k}\cdot\hat{\mathbf{n}}_{ii,k}A_k - \overline{\mathbf{u}\cdot\nabla(\rho h_s)}V_{ii}\right] \\
& + \frac{1}{\rho_{ii}}\sum_{\alpha}\left(\frac{\overline{W}}{W_\alpha} - \frac{h_{s,\alpha}}{c_p T}\right)_{ii}\left[\dot{m}'''_\alpha V_{ii} - \sum_{k=1}^{nf_c}\mathbf{J}_{\alpha,ii,k}\cdot\hat{\mathbf{n}}_{ii,k}A_k - \overline{\mathbf{u}\cdot\nabla(\rho Y_\alpha)}V_{ii}\right]
\end{aligned}
$$



Schematic of cut-cell in 2D: velocities and fluxes on faces, and scalars defined in cells.

Momentum Coupling

Scheme sequence:

1. Time advancement of scalars on cut-cells and regular gas cells.

2. IBM Interpolation to get **target velocities** in cut-faces

$$u_i^{ibm} = c_0 u_i^B + c_1 u_i^{\text{int}}$$

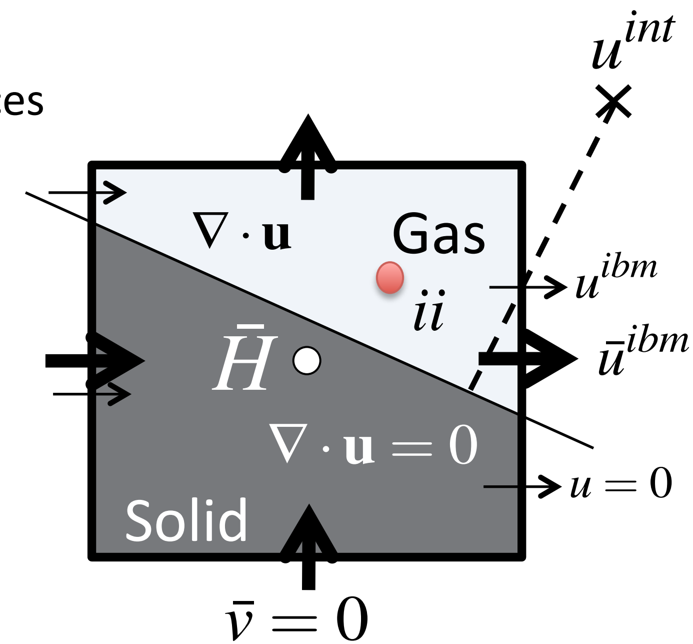
3. Flux **average** target velocities to Cartesian faces.

$$\bar{u}_i^{ibm} = \frac{1}{A_{cart}} \sum_k (u_i^{ibm} A_{cf})_k$$

4. Compute **direct forcing at Cartesian level**:

$$\bar{F}_i^n = -\left(\frac{\bar{u}_i^{ibm} - \bar{u}_i^n}{\Delta t} + \frac{\delta \bar{H}^n}{\delta x_i} \right)$$

5. Compute **thermodynamic divergence** on $(\nabla \cdot \mathbf{u})_{ii}^{th}$ cut-cells.



Momentum Coupling

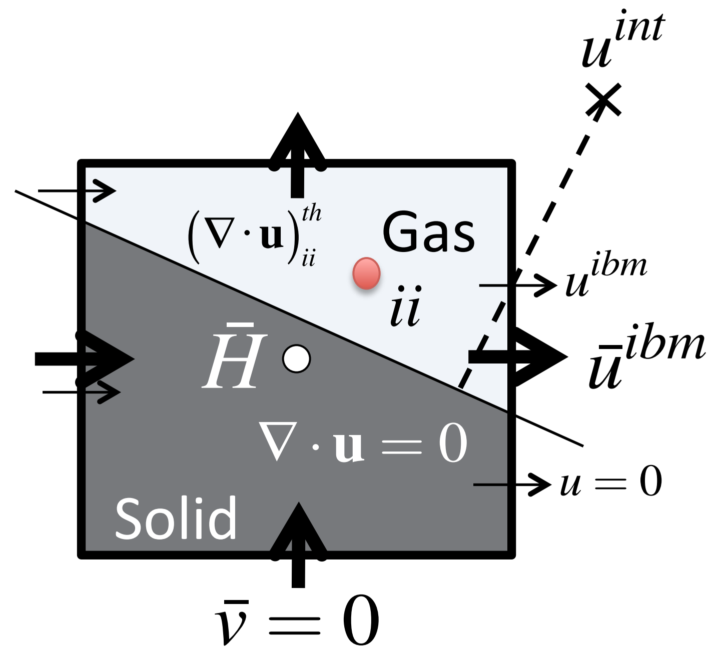
6. Use ***divergence integral equivalence***

$$\int_{\Omega_{cart}} \left(\nabla\cdot\overline{\mathbf{u}}\right)^{th} d\Omega = \sum_{ii} \int_{\Omega_{ii}} \left(\nabla\cdot\mathbf{u}\right)_{ii}^{th} d\Omega$$

to get Cartesian level target divergence $\left(\nabla\cdot\overline{\mathbf{u}}\right)^{th}$.

7. Solve Cartesian level Poisson equation

$$\nabla^2 \overline{H} = -\left(\nabla\cdot\overline{\mathbf{F}}^n + \frac{\left(\nabla\cdot\overline{\mathbf{u}}\right)^{th} - \left(\nabla\cdot\overline{\mathbf{u}}\right)^{n}}{\Delta t}\right)$$



(in order to avoid mass penetration into body, solve on gas phase and cut-cell underlying Cartesian cells).

8. Project Cartesian velocities into target divergence field $\overline{\mathbf{u}}^{n+1} = \overline{\mathbf{u}}^{n} - \Delta t\left(\overline{\mathbf{F}}^n + \nabla\overline{H}\right)$

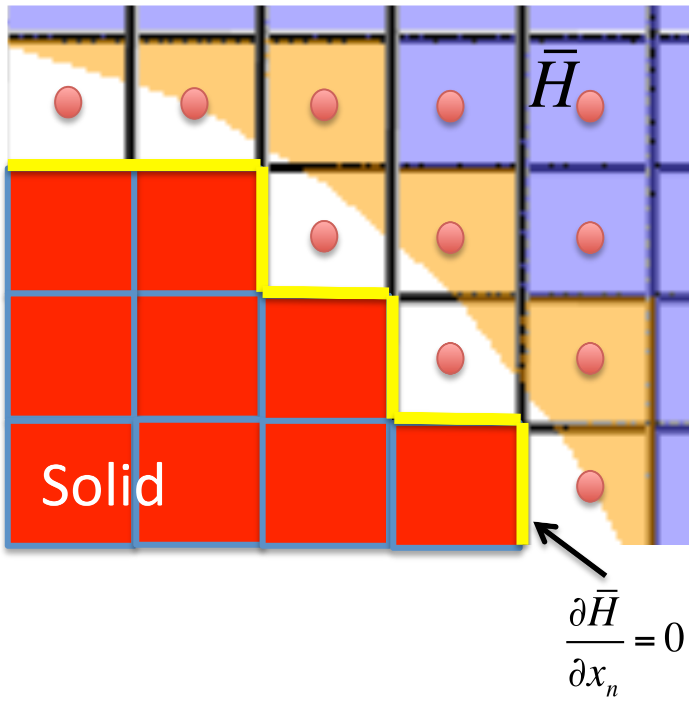
9. Reconstruct cut-face velocities.

Poisson Equation

- IBM: solve Poisson equation on the whole Cartesian domain, including cells within the immersed solids.
- Introduces mass penetration into the solid on the projection step. Undesirable for conservation, combustion.
- Our Momentum Coupling scheme: use this type of Pressure solver, or an **unstructured solution on Cartesian gas cells and cells underlying cut cells**.

Global linear system solver:

- Building a global Laplacian matrix in parallel.
- Building the global RHS.
- Calling Parallel Matrix-Vector solver, currently MKL **cluster sparse direct solver**.
- Capability to define correct H boundary condition in FDS &OBSTS and complex geometry bodies &GEOM.



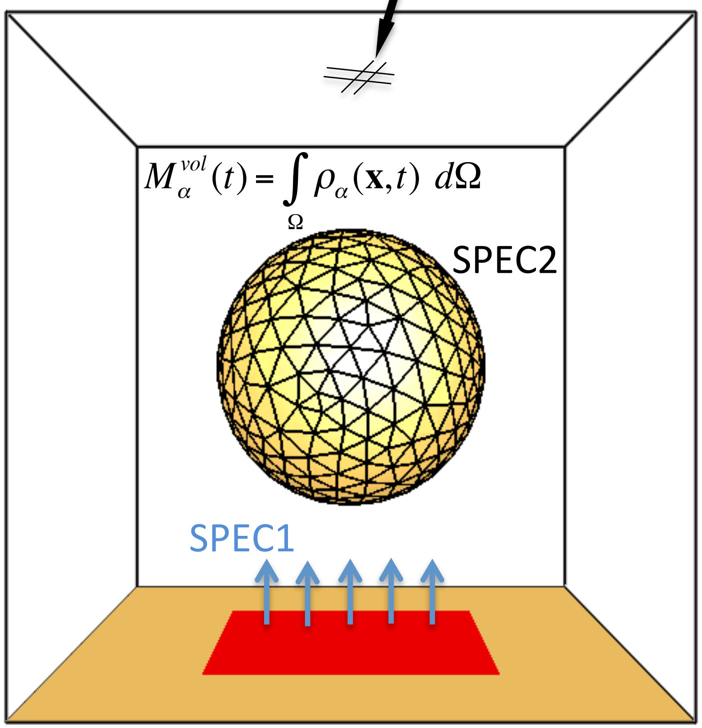
Example

- **Isothermal Gas Plume around immersed sphere:**

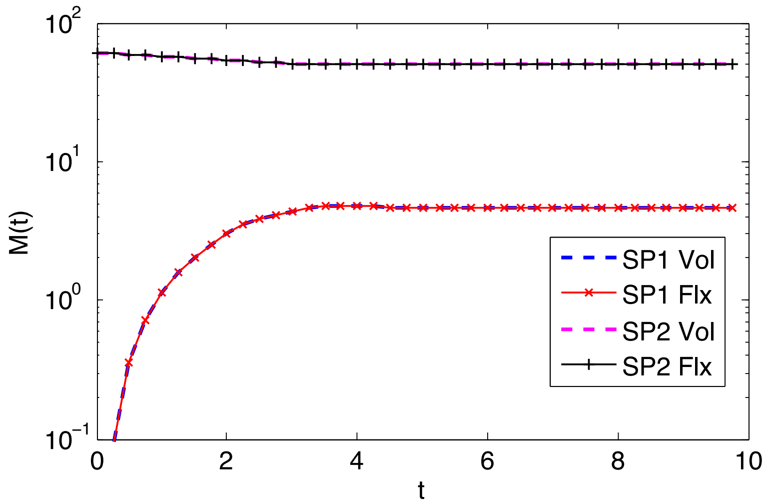
Test conservation of EXIM scalar transport and transport terms in divergence expression for cut-cells.

- **Two species**: SPEC1: MW ~ 12, SPEC2: MW ~ 24 kg/mol.
- $\mu = 0.0005; \quad D = 0.0002$ SI units.
- Inflow on bottom VENT, open boundary on others.
- Re=4000 based on unit velocity (inflow) and SPEC2 density.
- SPEC2 taken as background, DNS mode.
- Run for 10s, dt=0.0025, 40^3 Cartesian cells.
- Transport for **scalars** in cut-cell region using **BE Predictor + Trapezoidal Corrector**, solved with MKL **Pardiso**.
- **Poisson equation** defined in regular gas and cut-cell underlying Cartesian cells, solved with MKL **Pardiso**.
- Scalar calculation takes about twice the time of FDS using a square block OBST.
- Check total mass deficit of species as volume integral vs. domain boundary mass flux time integrals.

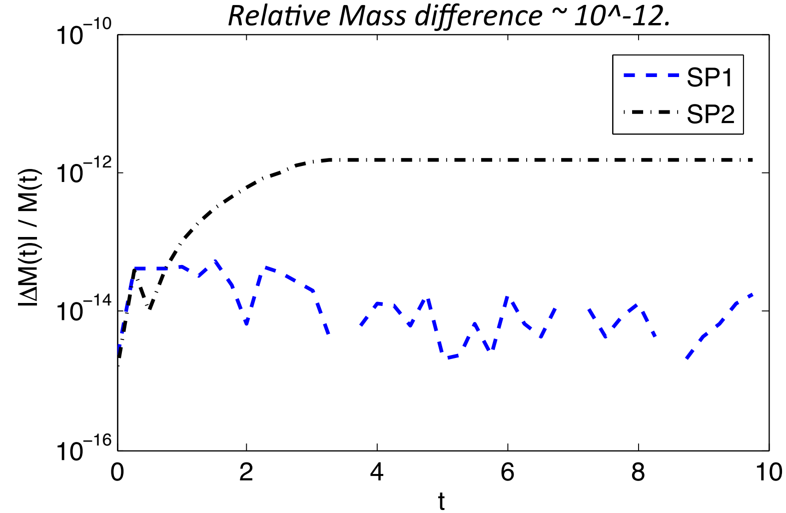
$$M_\alpha^{flx}(t) = M_\alpha^{vol}(t_0) + \int_{t_0}^{t} \int_{\Gamma} \dot{m}_\alpha''(\mathbf{x},t') \; d\Gamma dt'$$



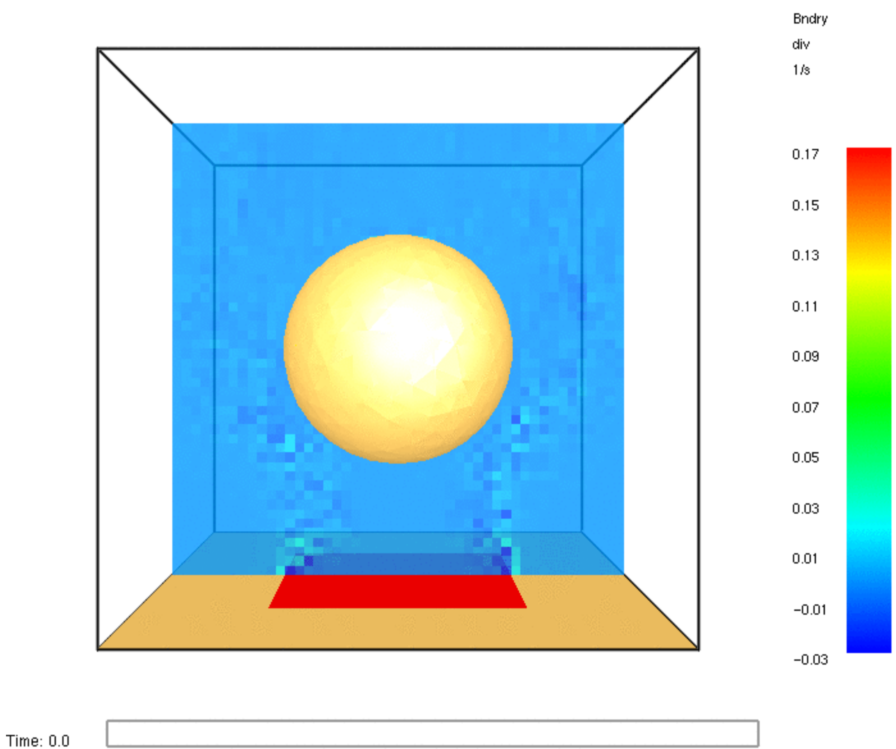
Example



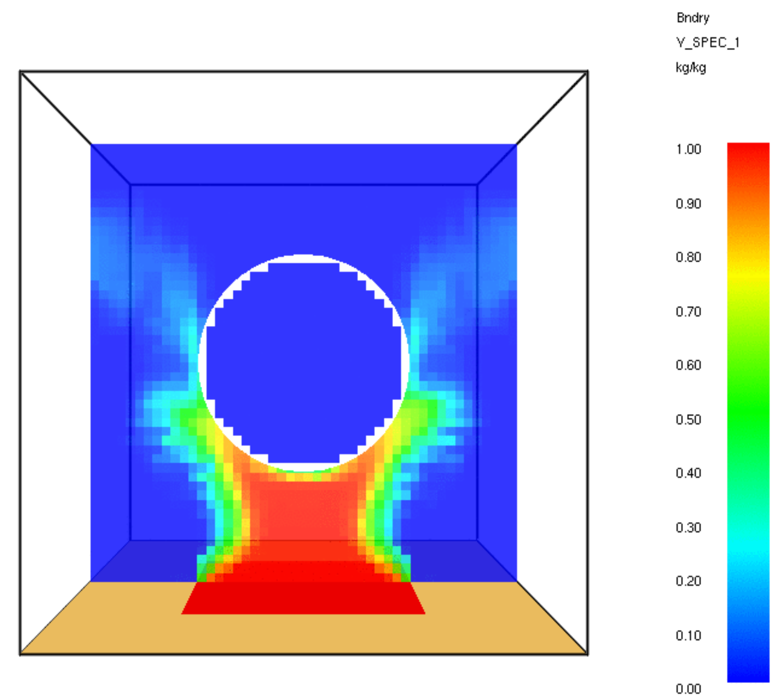
M(t)
t
SP1 Vol
SP1 Flx
SP2 Vol
SP2 Flx



Relative Mass difference ~ 10^-12.
|ΔM(t)| / M(t)
t
SP1
SP2



Bndry
div
1/s
Time: 0.0



Bndry
Y_SPEC_1
kg/kg

Example

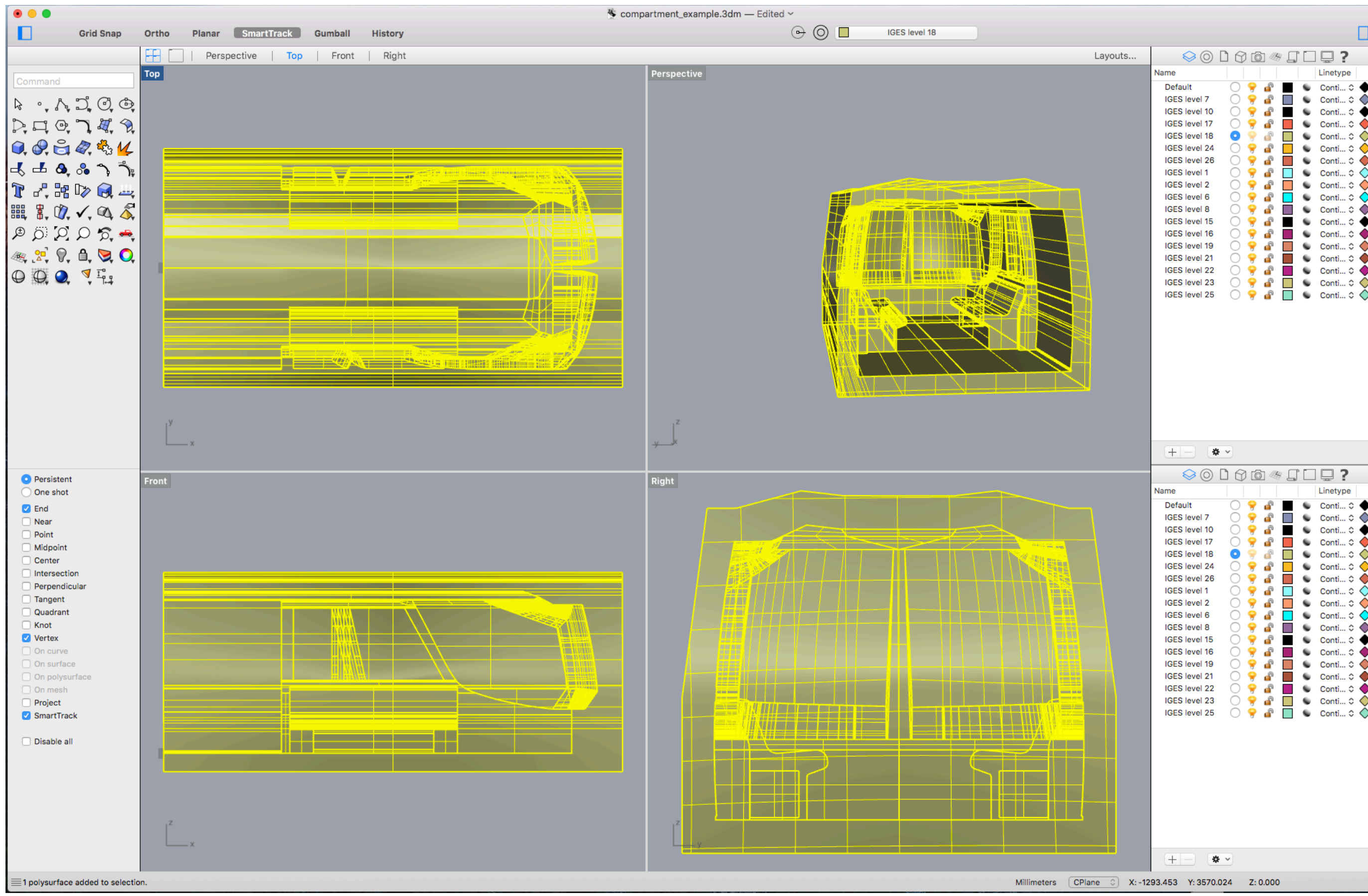
- **Propane fire in train cabin:**

FDS **&GEOM** definition Work flow

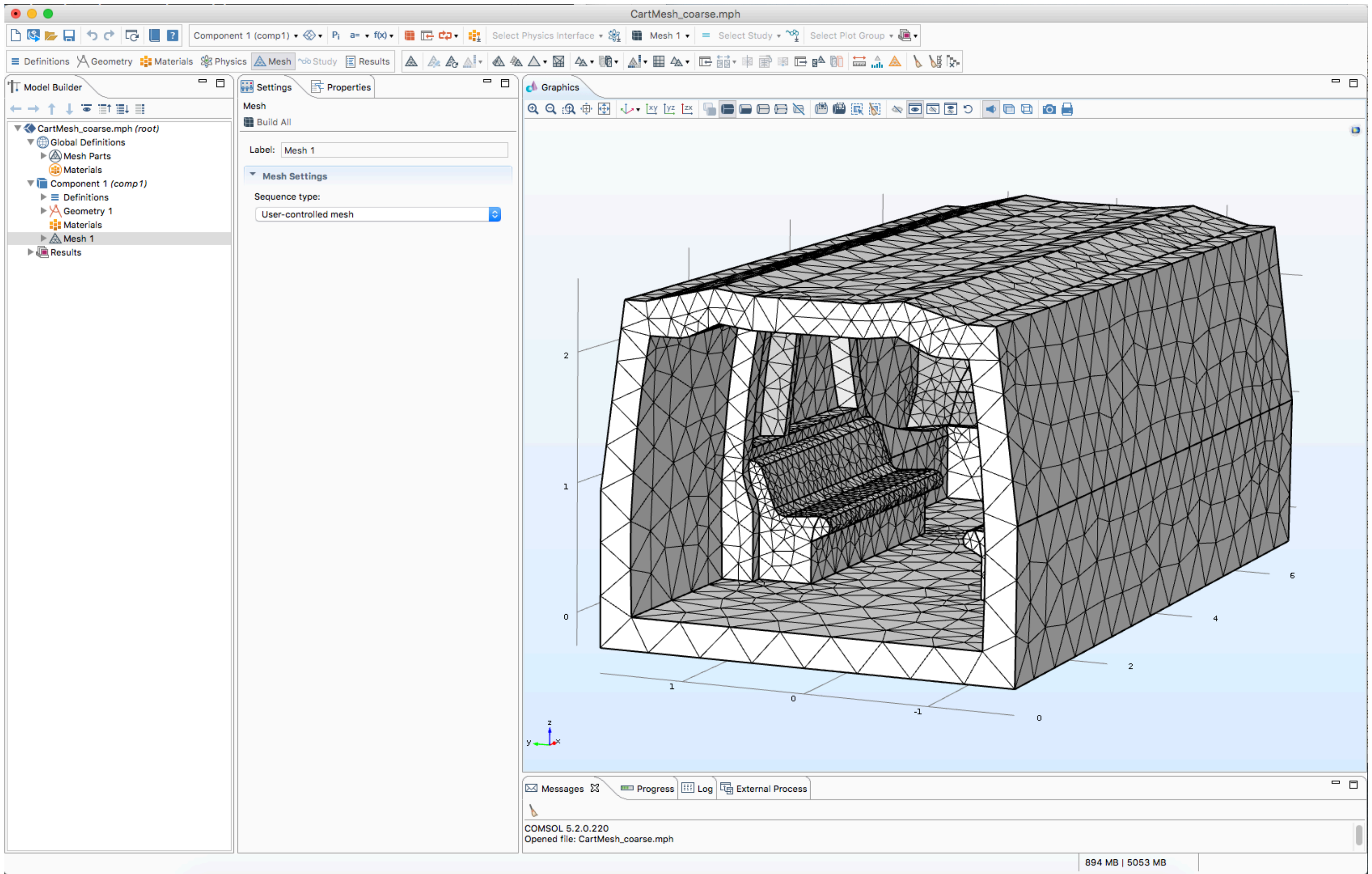
Realistic Train Cart model, courtesy of Fabian Braennstroem (Bombardier).

1. *Model defined in CAD software as a set of sanitized, disjoint volumes.*

2. *Exported in format to read on meshing software (*.igs, *.stl).*



Example



3. *Geometry meshed in meshing software (i.e. COMSOL, Hypermesh, Gambit).*

4. *Mesh exported in neutral text format.*

Example

3. *Mesh file is converted into FDS input format.*

4. *Rest of simulation data is defined.*

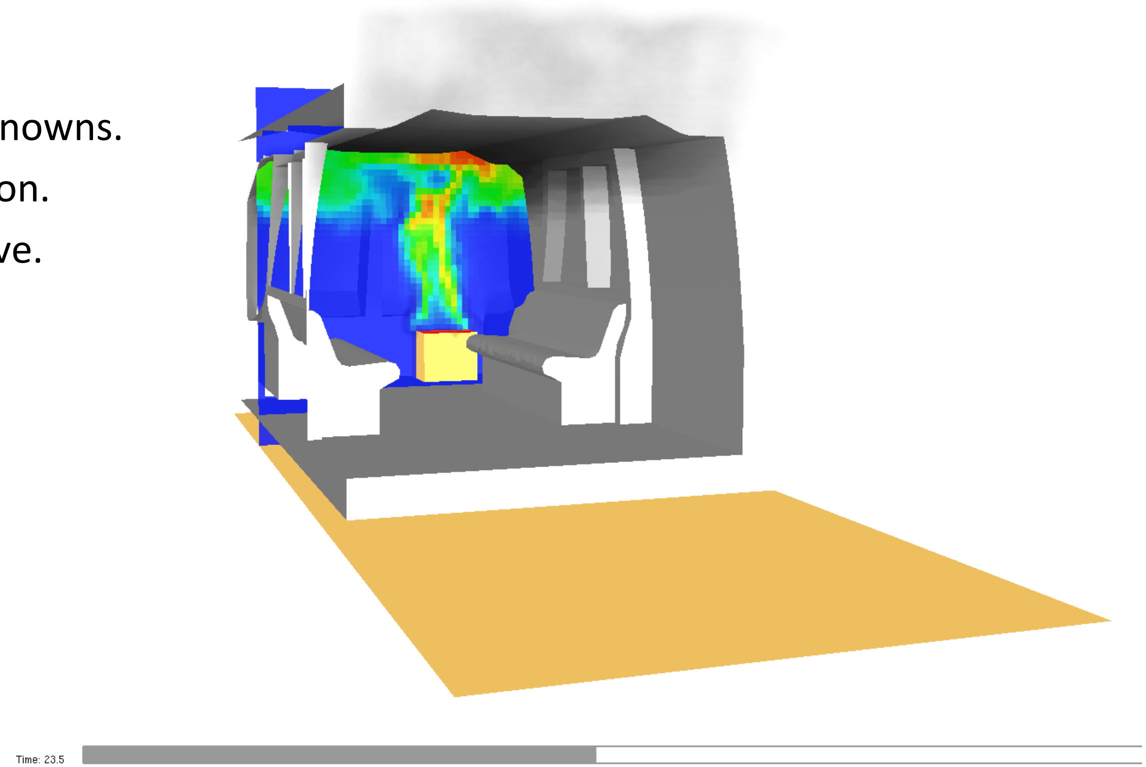
```
&HEAD CHID='cart_fire_800KW', TITLE='CC-IBM: Test propane fire on realistic train cart geometry.' /

&MESH IJK=144,78,58, XB=-2.75,7.25,-2.25,2.25,-0.5,2.85 /

&TIME T_END=50.0 /
&MISC DNS=.FALSE.,
      NOISE=.FALSE.
      STRATIFICATION=.FALSE.,
      CONSTANT_SPECIFIC_HEAT_RATIO=.FALSE.,
      BAROCLINIC=.FALSE.,
      PROJECTION=.TRUE.,
      CFL_VELOCITY_NORM=1,
      CC_IBM=.TRUE.,
      DO_IMPLICIT_CCREGION=.FALSE. /

&PRES GLMAT_SOLVER=.TRUE. /
&RADI RADIATION=.FALSE. /

# Vents:
&VENT MB='ZMAX', SURF_ID='OPEN' /
&VENT MB='YMIN', SURF_ID='OPEN' /
&VENT MB='YMAX', SURF_ID='OPEN' /
&VENT MB='XMIN', SURF_ID='OPEN' /
&VENT MB='XMAX', SURF_ID='OPEN' /

# Species:
&REAC FUEL='PROPANE', SOOT_YIELD=0.02 /

&SURF ID='BURNER', HRRPUA=3200., COLOR='RED' /
&SURF ID='cart', COLOR='GRAY', MATL_ID='cart', THICKNESS=0.1/
&MATL ID='cart', DENSITY=1, CONDUCTIVITY=1, SPECIFIC_HEAT=1/

# Geometries:
&OBST XB=4.5,5.0,-.75,-0.25,0.05,0.55, SURF_IDS='BURNER','INERT','INERT' /
&GEOM ID='FEM_MESH', MATL_ID='cart', SURF_ID='cart'
VERTS=
-0.00019300,      -1.68766011,     0.14260193,
-0.00056374,      -1.68765986,     0.55683507,
 1.78482997,      -0.59141201,     0.52607399,
```

cart_fire_800KW.fds demo FDS input file.

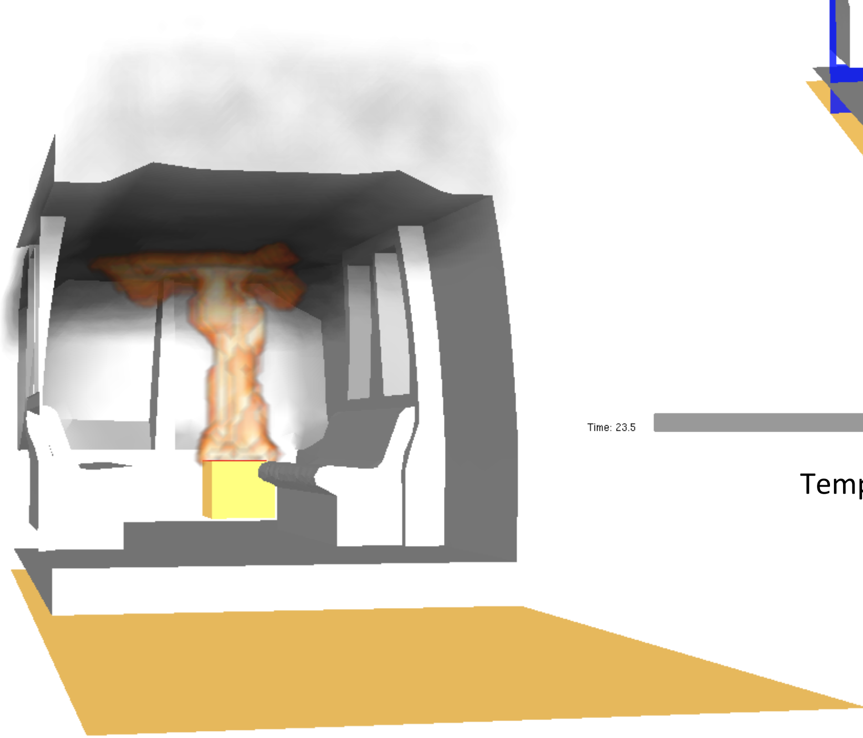
Example

- **LES of propane fire in train cabin:**

- 800 KW Propane burner.
- 144x78x58 grid, ~50K CC scalar unknowns.
- Explicit scalar integration in CC region.
- Unstructured Cartesian Poisson solve.



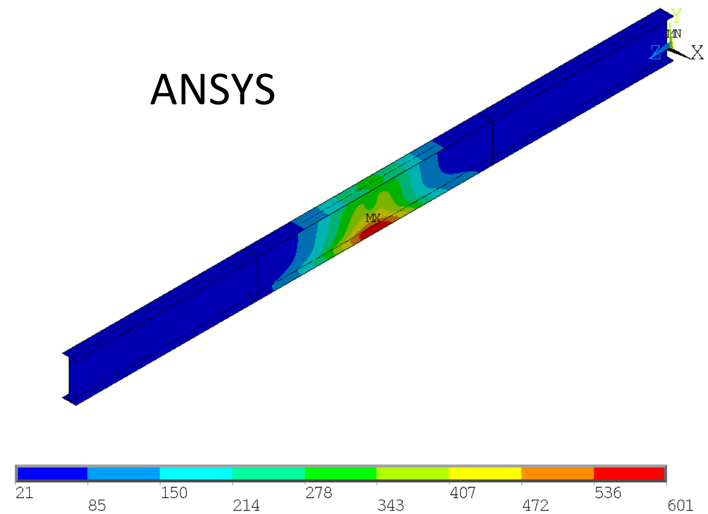
Temperature slice 20C (blue) to 1500C (red), + velocity vectors (black).

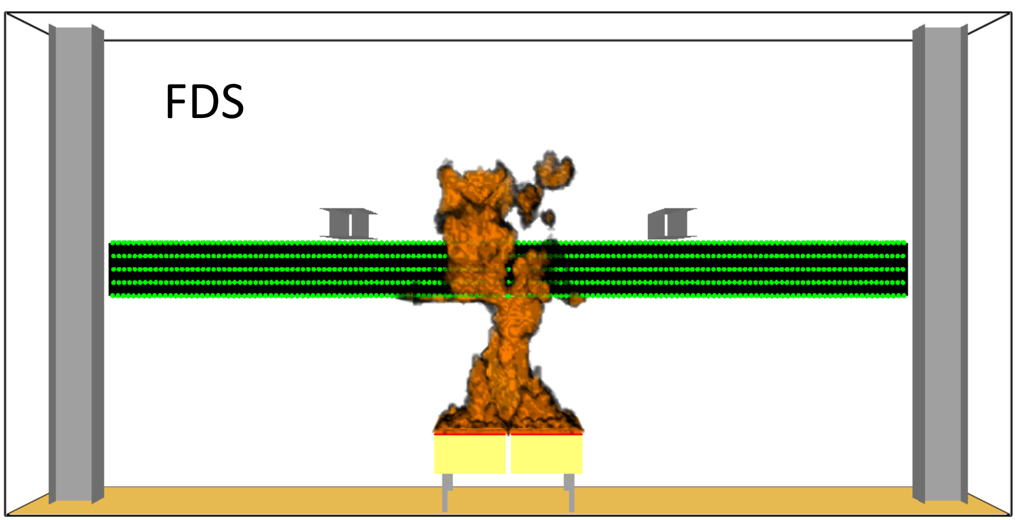


Smoke + HRRPUV contours.

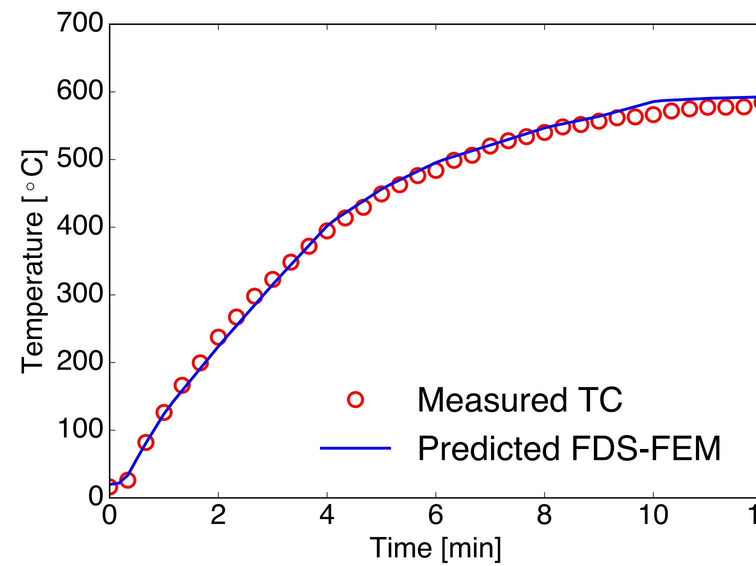
Future Work

- Verification and Validation.
- Enhance radiation solver to solve RTE on cut-cells, add radiative boundary conditions on boundary cut-faces.
- Extend the treatment of particles from Cartesian cells unstructured cut-cells.
- Develop the data transfer for two way coupling with thermo-mechanical FEM solvers + moving internal boundaries.





NFRL commissioning test, courtesy Chao Zhang.



Thank you