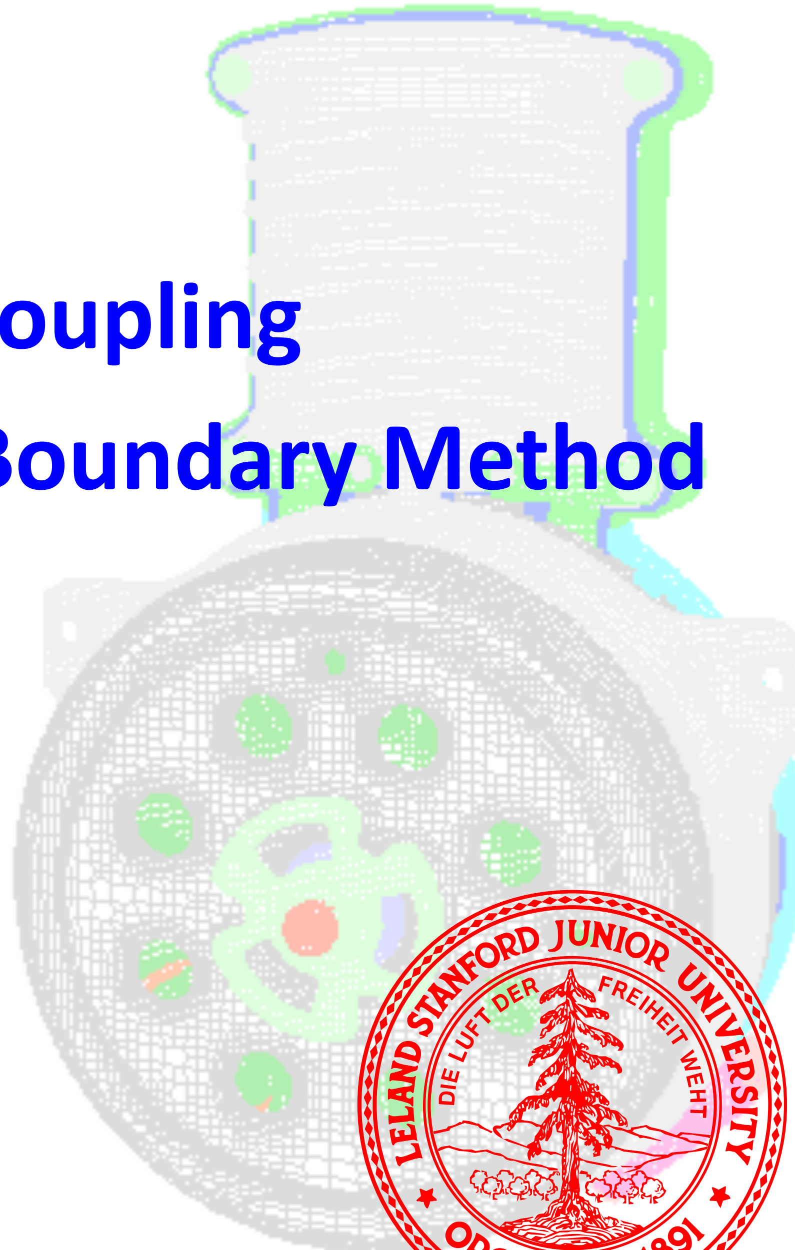
Solid/Fluid Thermal Coupling Using the Immersed Boundary Method



Gianluca Iaccarino
Mechanical Engineering
Stanford University

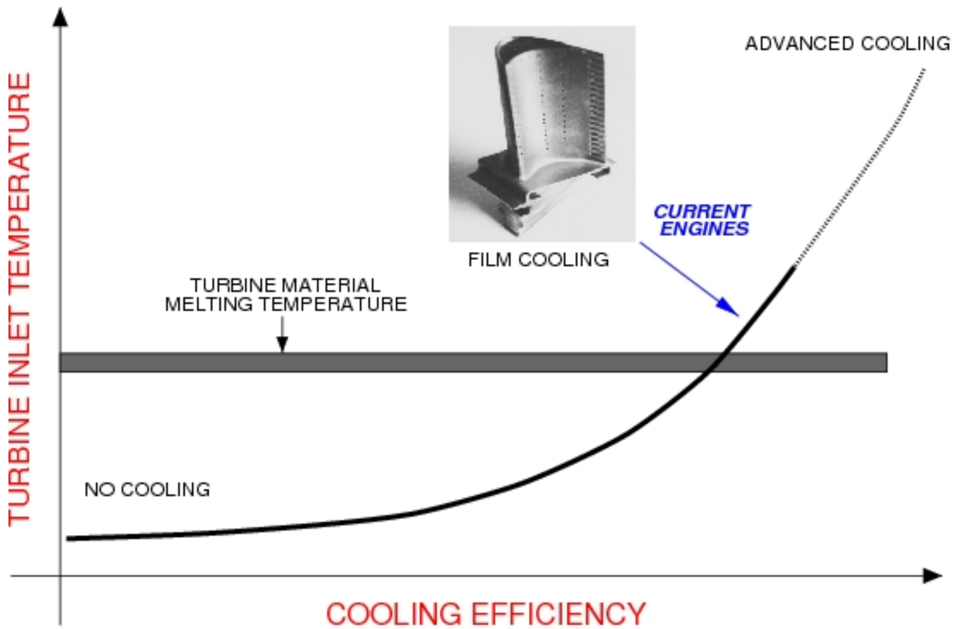
XXXV WSC Conference
October 6-10, 2010, Zeist, The Netherlands.

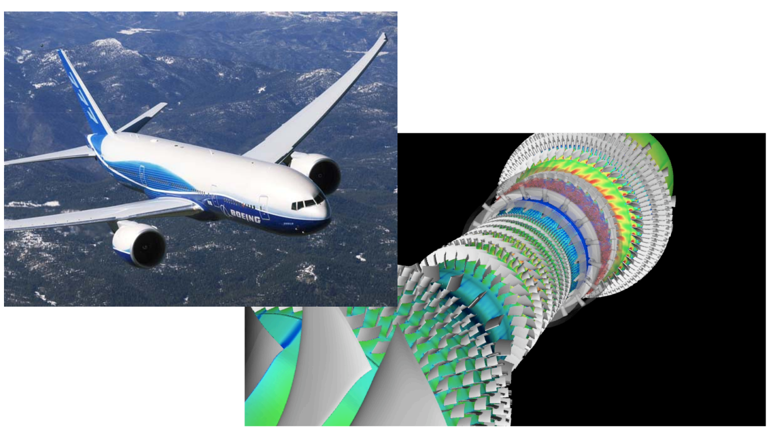
1. Motivations & Objectives

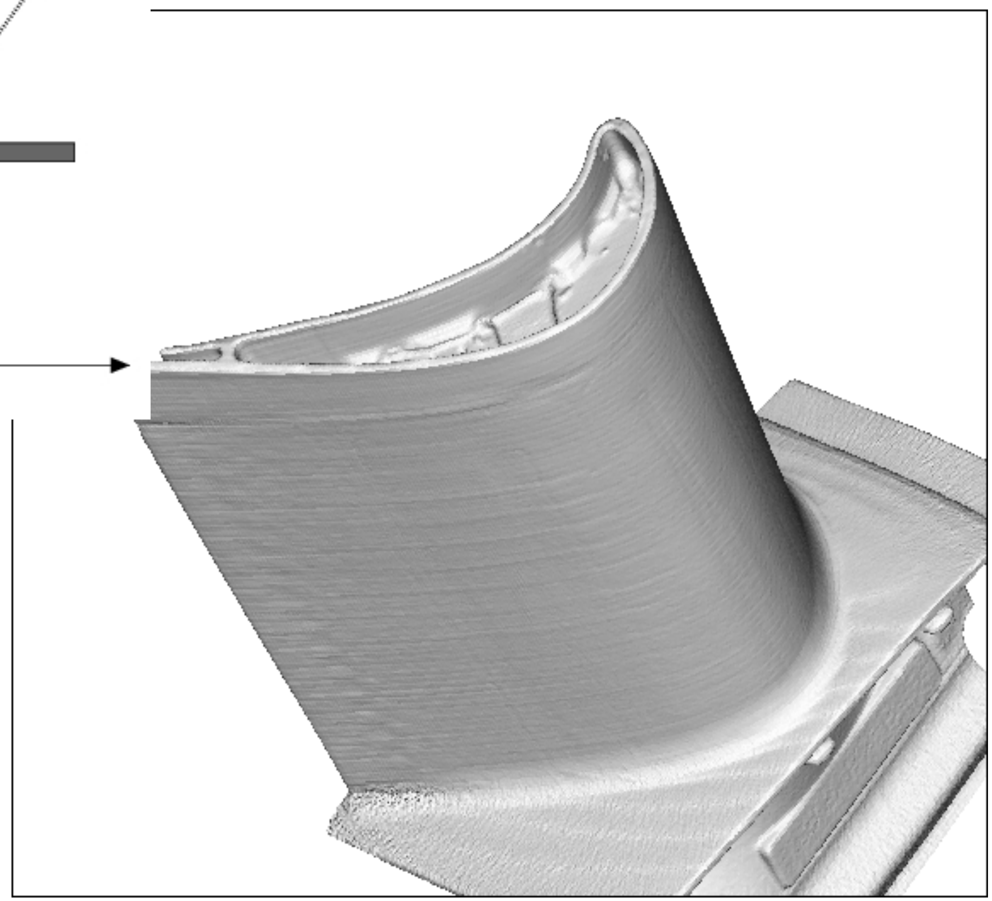
Solid/Fluid Thermal Coupling
Using the Immersed Boundary Method

Why Solid/Fluid
Thermal Coupling?

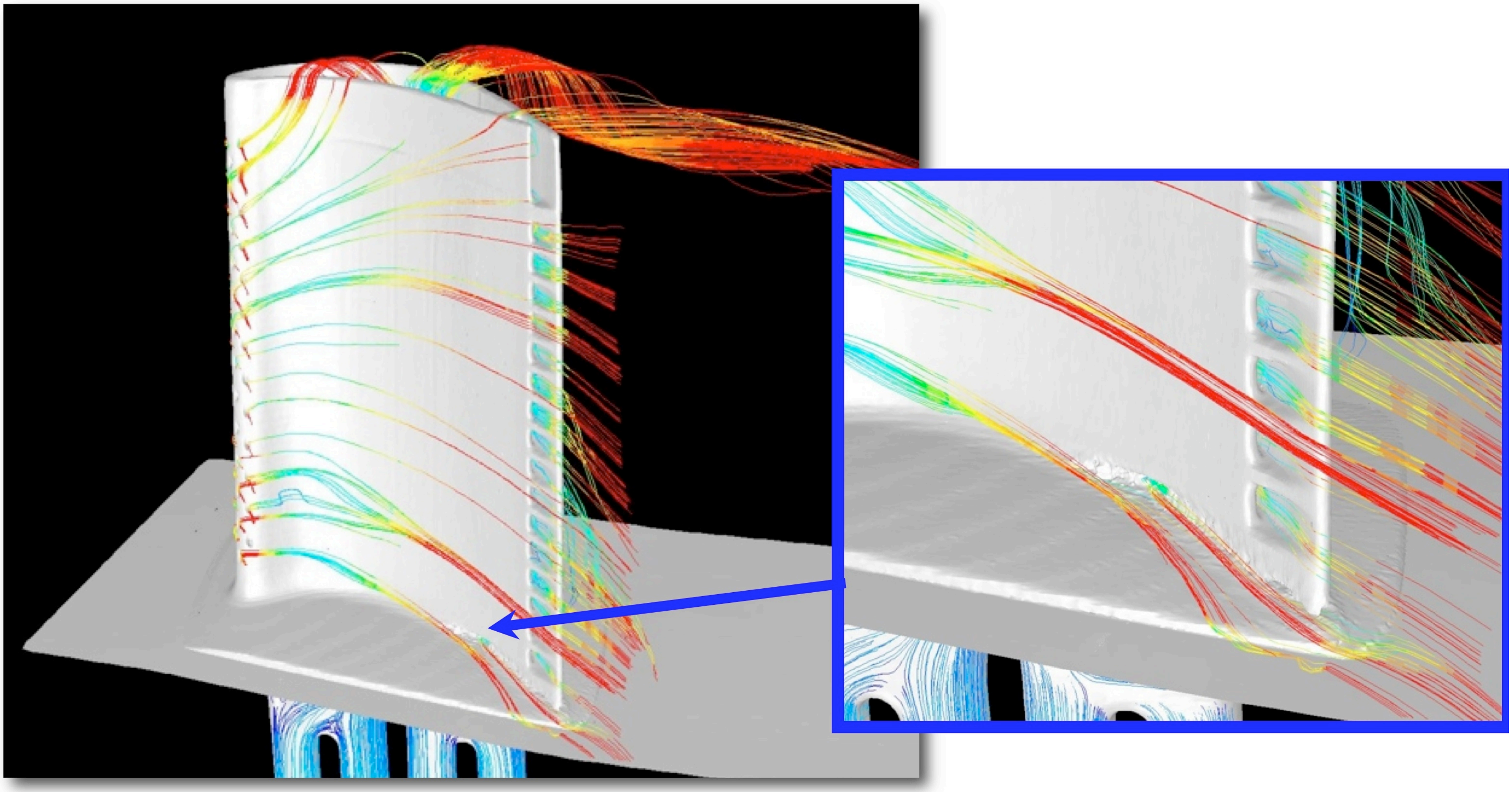
Cooling of a Turbine Blade



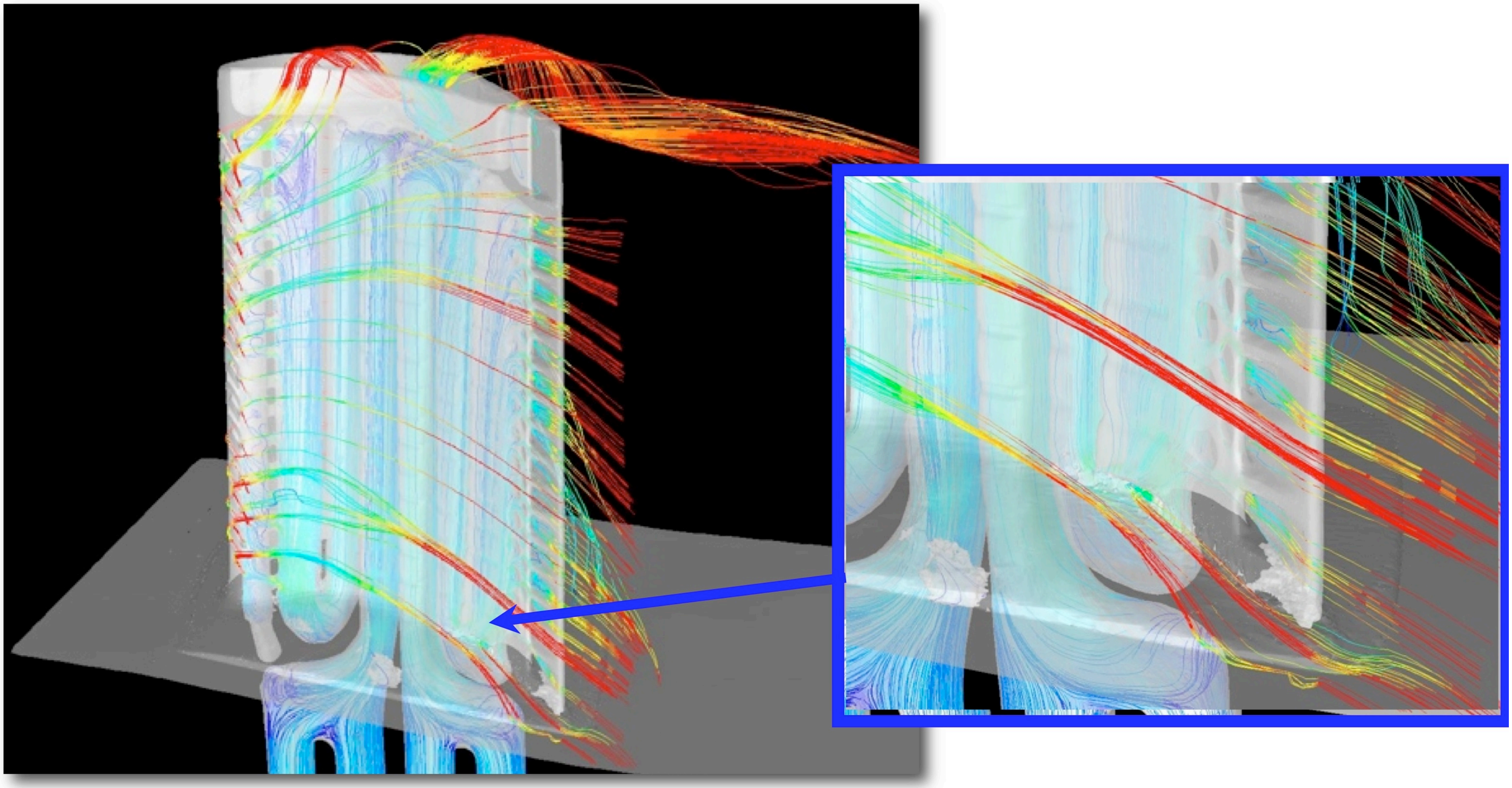




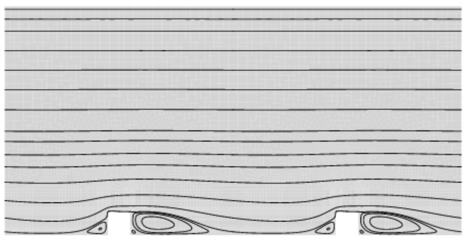
Thermally Induced Damage?



Thermally Induced Damage?

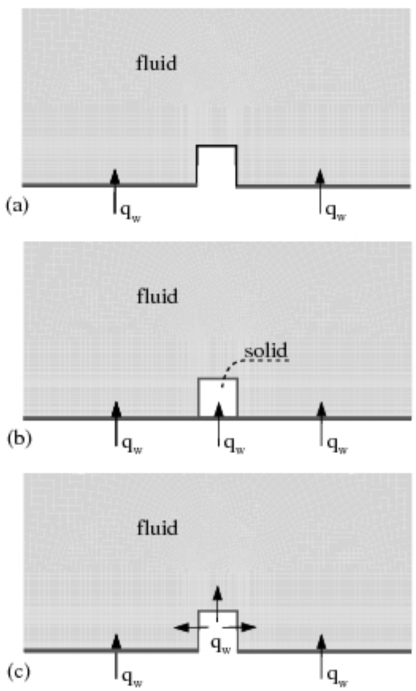


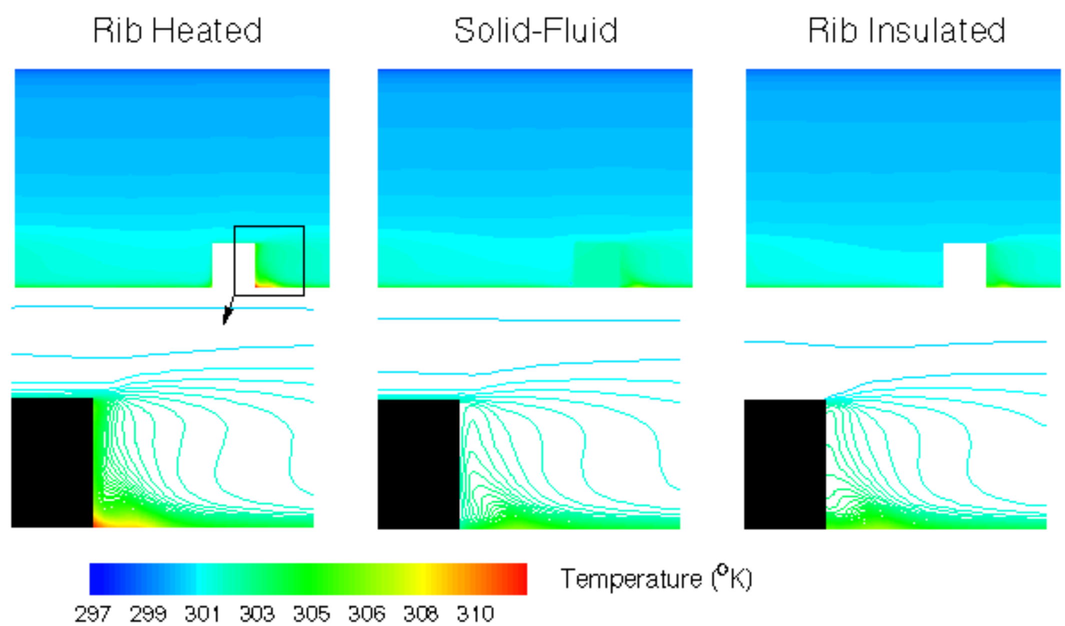
Effect of Thermal B.C.



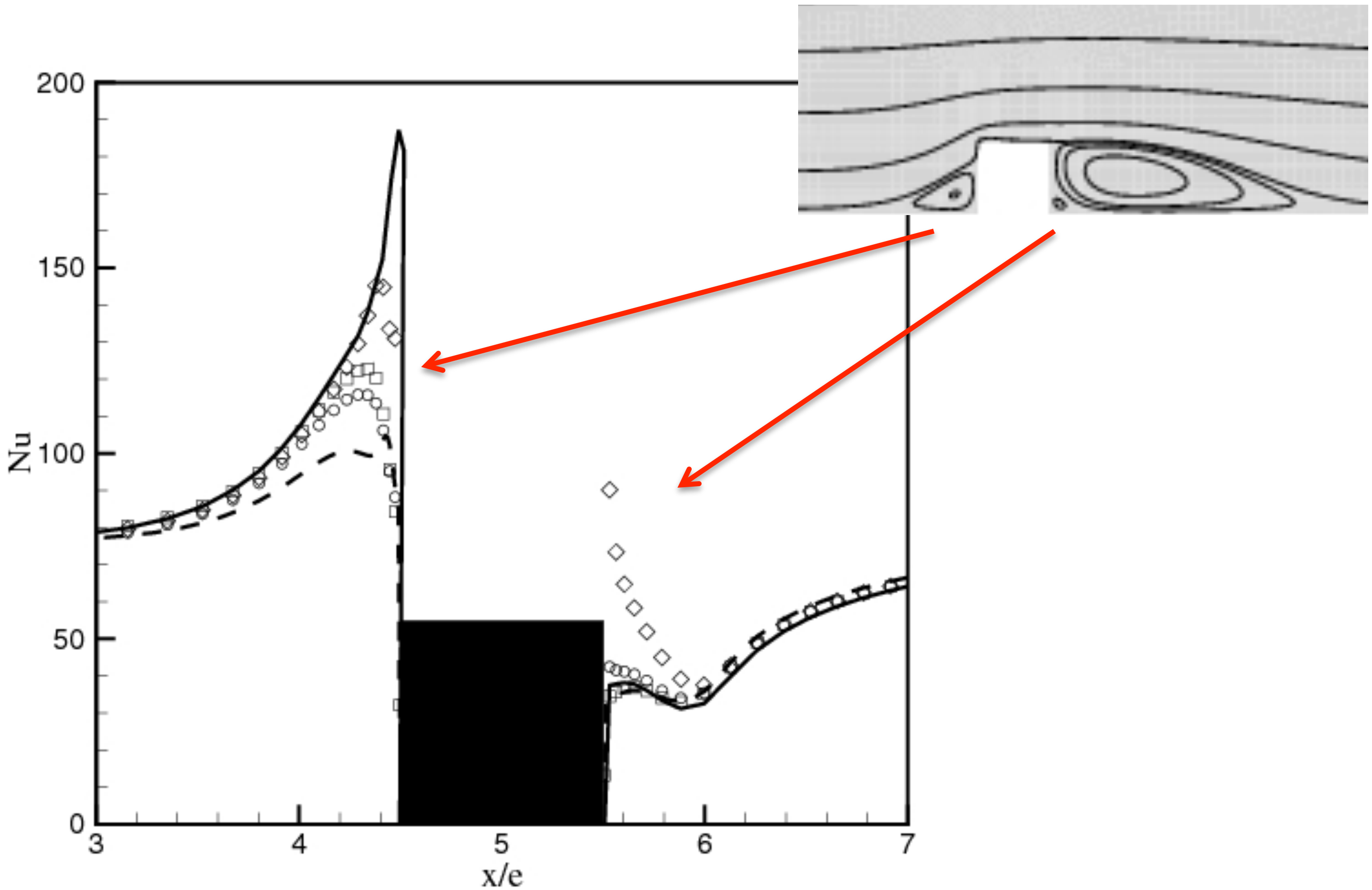
What is the appropriate thermal boundary condition?

Is it important to evaluate the wall heat transfer?





Effect of Thermal B.C.



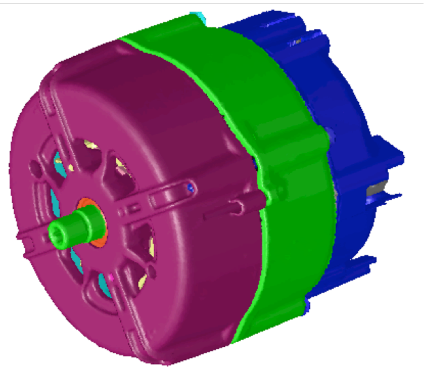
Fluid Only

Solid/Fluid Coupled

——— Insulated rib, − − − heated rib,

◇ $k_s/k_f = 100$, ○ $k_f = k_s$ and □ $k_f/k_s = 100$.

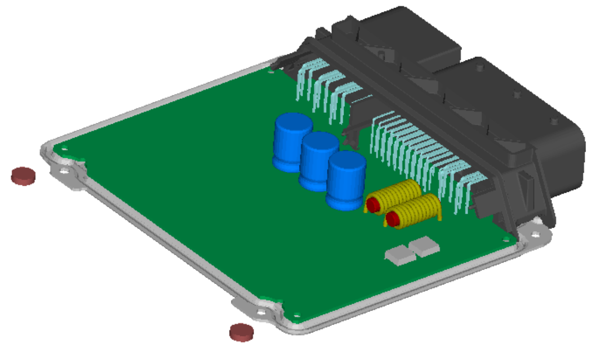
Not only Turbine Blades

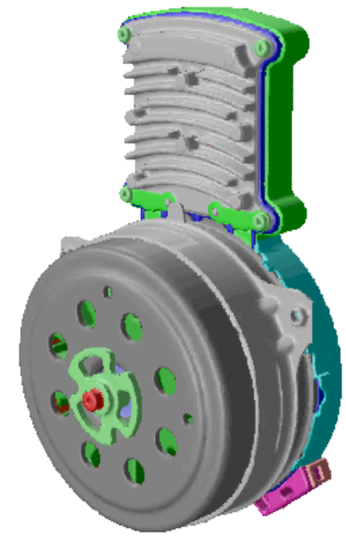


Alternator

Valeo

Electronic Component Unit (ECU) BOSCH



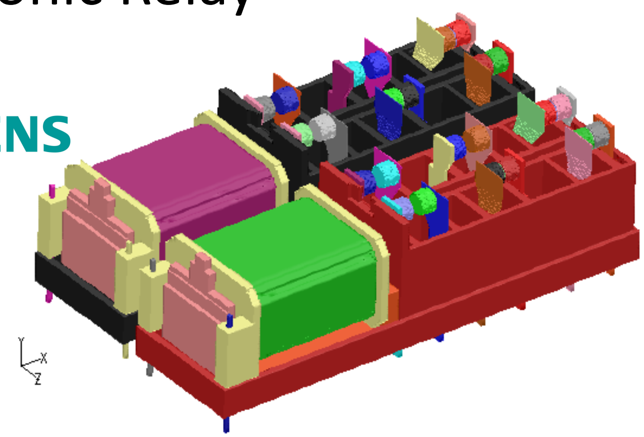


Electric Motor

Valeo

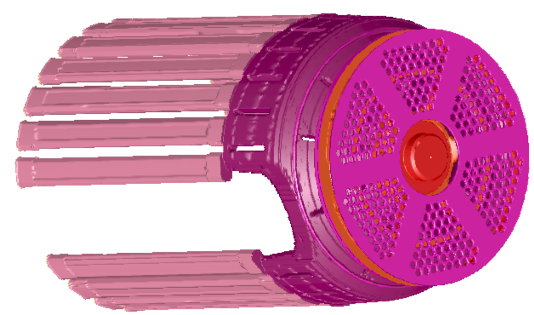
Electronic Relay Board

SIEMENS



Electronic Motor

SIEMENS



The future is GREEN

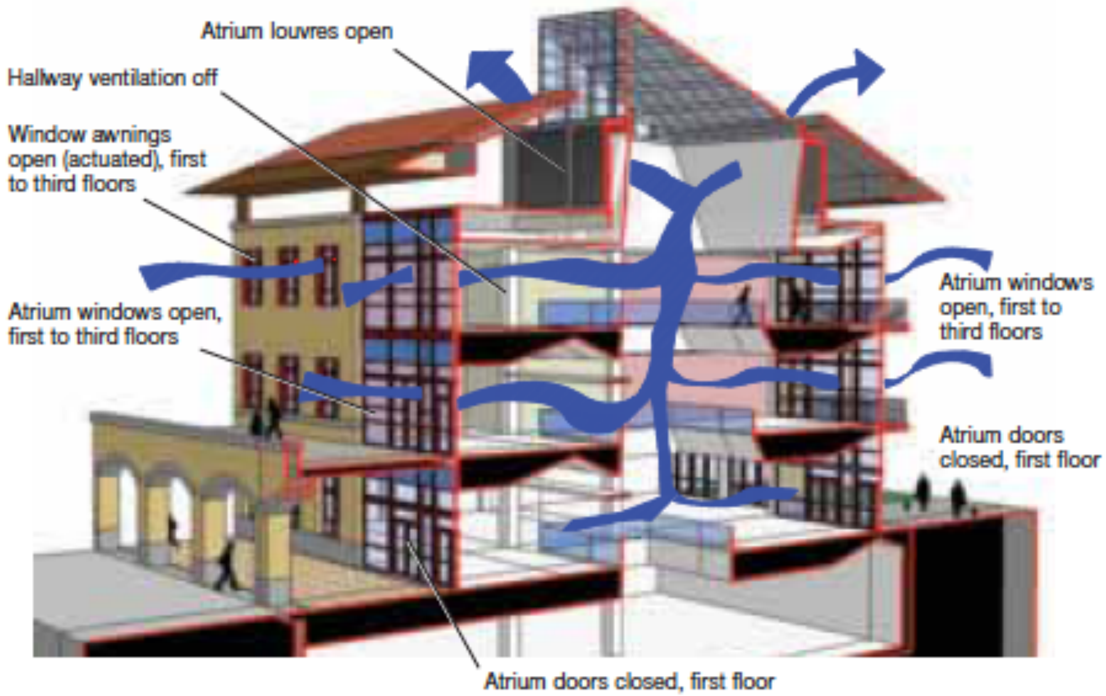
Y2E2 Building @ Stanford

Natural Ventilation is a key for energy savings

The future is GREEN

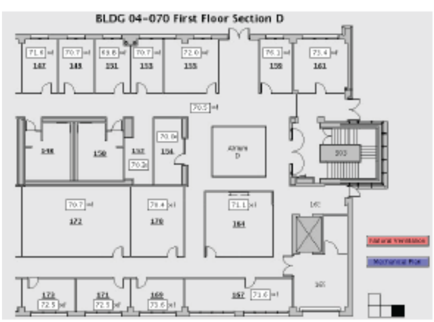
Y2E2 Building @ Stanford

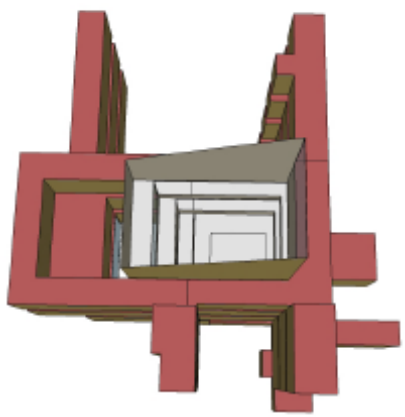
Objective: simulate night-time air-flushing

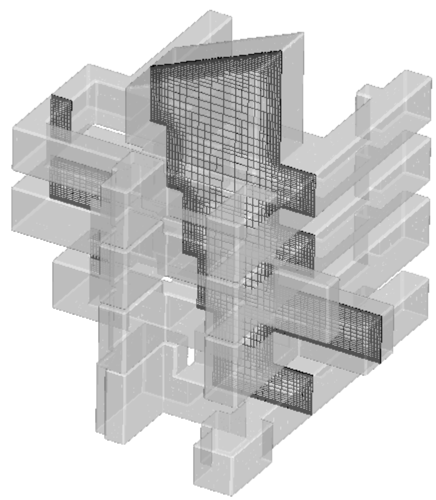


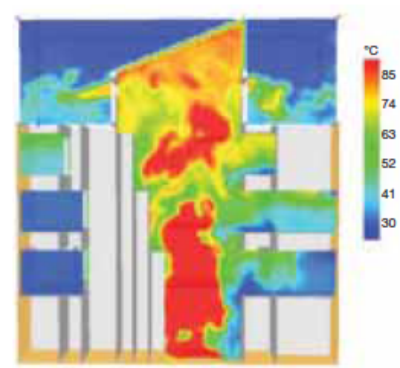
The future is GREEN

Y2E2 Building @ Stanford









Enable CFD-aided Architectural Design

Challenges

- CFD simulation of Conjugate Heat Transfer (CHT) are NOT routinely performed
  - Geometry: many parts, intricate passages, range of length scales
  - Grid generation: resolution requirement dependent on the part, time consuming
  - Physical models: treatment of multi-modal thermal transfer: natural/forced laminar/turbulent convection, conduction & radiation
  - Coupling: Enforce proper conservation at the solid/fluid interface

Objective

- Introduce an Immersed Boundary method to perform coupled solid/fluid simulations in extremely complex geometries
- Provide a technical description and discuss implementation details of the approach
- Demonstrate the accuracy in simple applications and selected industrial problems

2. Immersed Boundary Methods

Solid/Fluid Thermal Coupling

Using the Immersed Boundary Method

Immersed Boundary

Science is a differential equation, religion is the boundary condition

– A. Turing

The differential equation is the science, the boundary condition is a religion

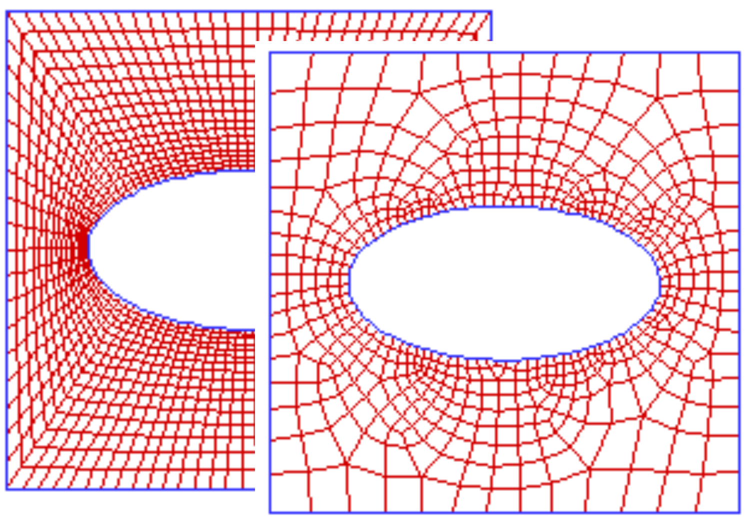
– G. Iaccarino

- A variety of research areas:
  - Develop numerical methods
  - Study fluid mechanics in complex domain (perhaps with moving boundaries)
  - Interested in interactions between fluid and structures
  - Analyze multiphase flows
  - ...

Alternative Approaches

Body-Fitted

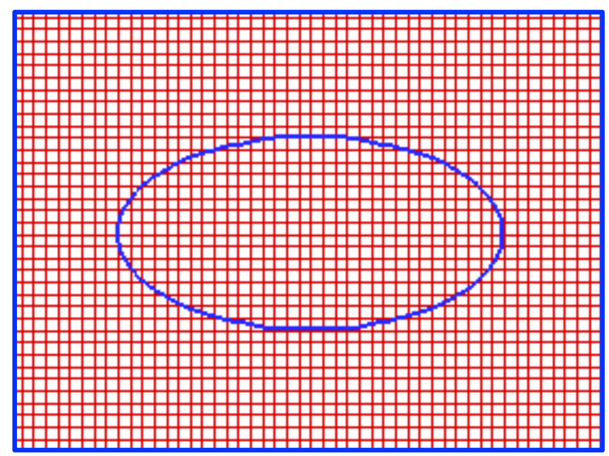
- The computational domain is contained within physical boundaries
- A body-fitted mesh is generated in the domain
- Solution algorithms handle structured or unstructured grids



Structured Grid or
Unstructured Grid

Immersed Boundary

- The computational domain extends beyond the physical boundaries
- A Cartesian mesh is generated in the domain
- The governing equations are modified in the cells cut by the interface



Cartesian Grid

Scaling Argument

Desired resolution is Δn, Δt at the boundary

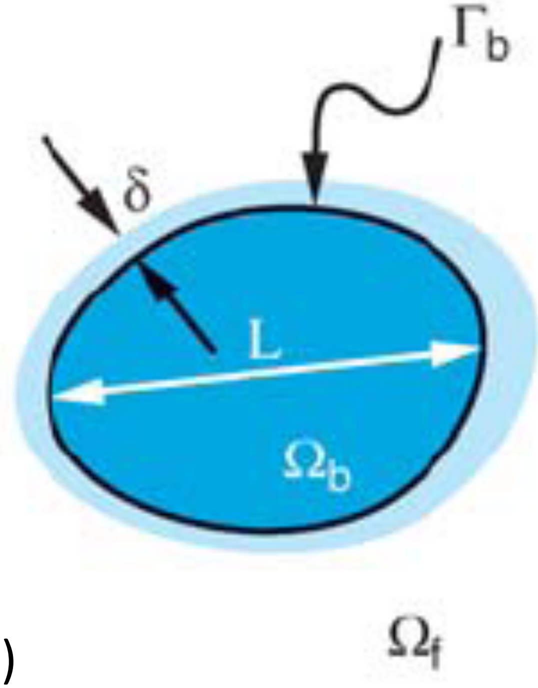
Assume $\delta << L$

Body Conformal Grid ~ $(L/\Delta t)(\delta/\Delta n)$

Cartesian Grid ~ $(L/\Delta n)^2$

Assume $\Delta n \sim \delta$ and $\Delta t \sim L$ (and with $L/\delta = Re^{0.5}$)

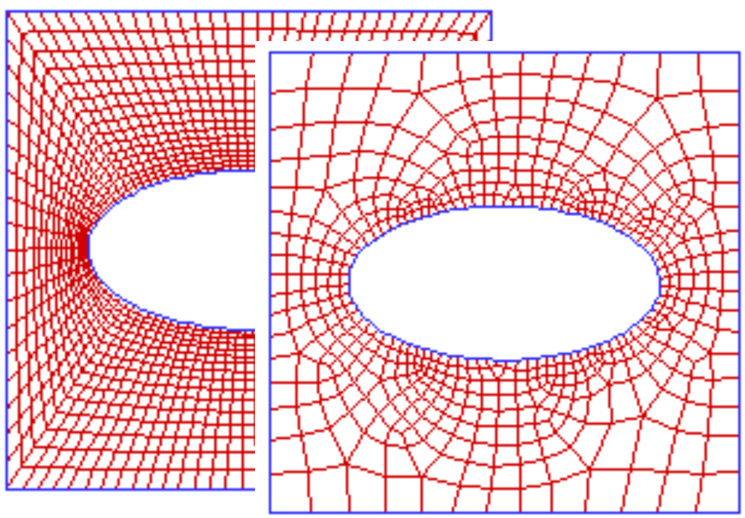
The grid-size ratio (in 2D) scales as the Re number: the IB is progressively more expensive….



Alternative Approaches

Body-Fitted

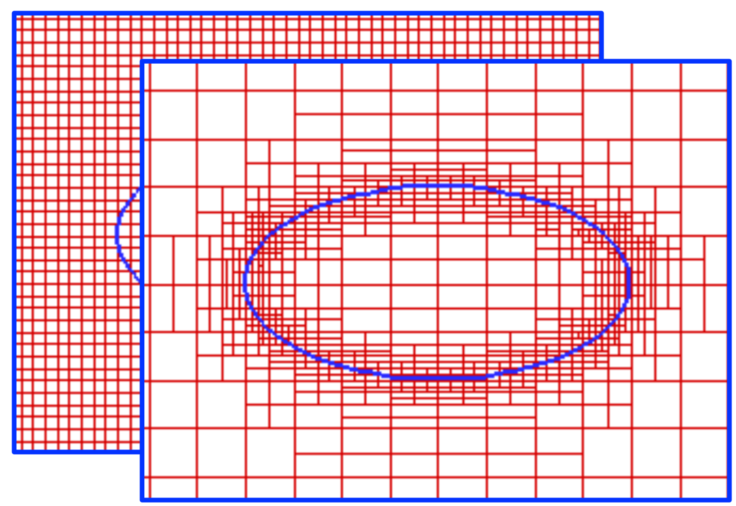
- The computational domain is contained within physical boundaries
- A body-fitted mesh is generated in the domain
- Solution algorithms handle structured or unstructured grids



Structured Grid or
Unstructured Grid

Immersed Boundary

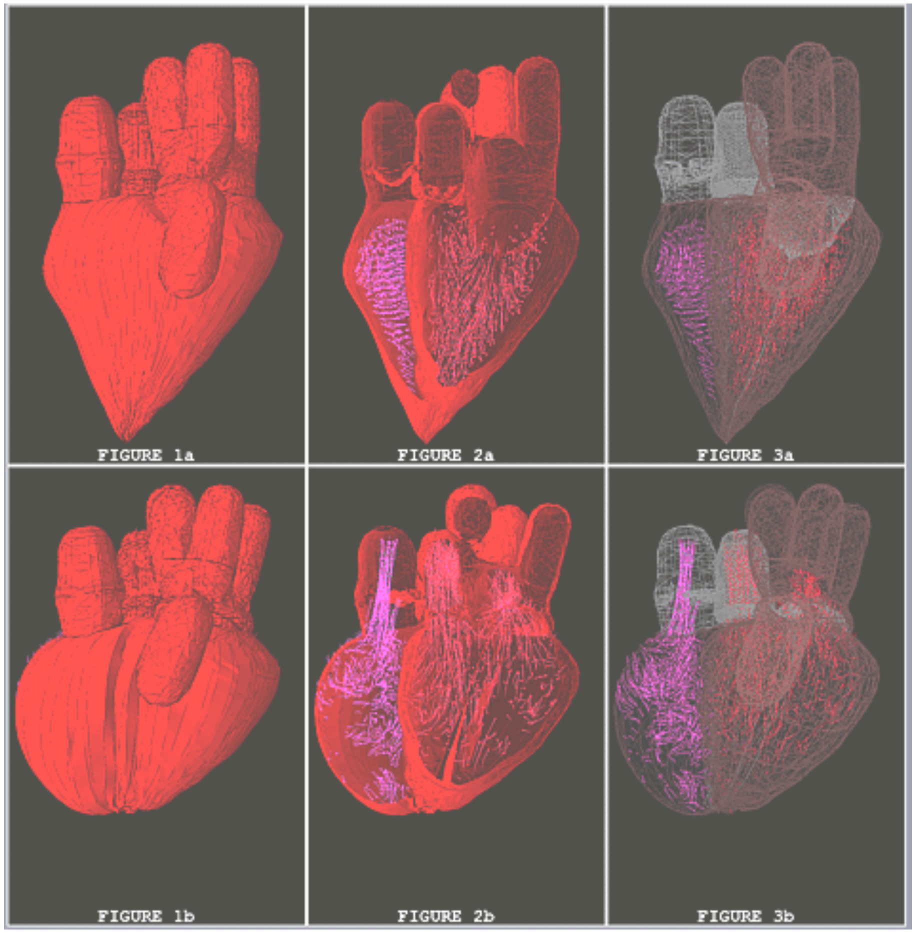
- The computational domain extends beyond the physical boundaries
- A Cartesian mesh is generated in the domain
- The governing equations are modified in the cells cut by the interface



Cartesian Grid
Unstructured Grid

IB Methods: the origin

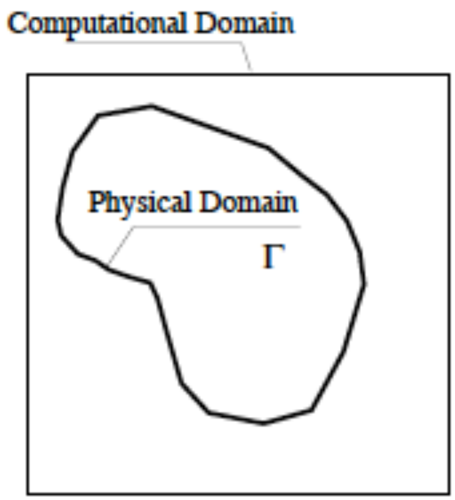
One father (many grandparents): Charles Peskin, 1972



1st ever simulation of a human heart-beat.

IB Methods: the origin

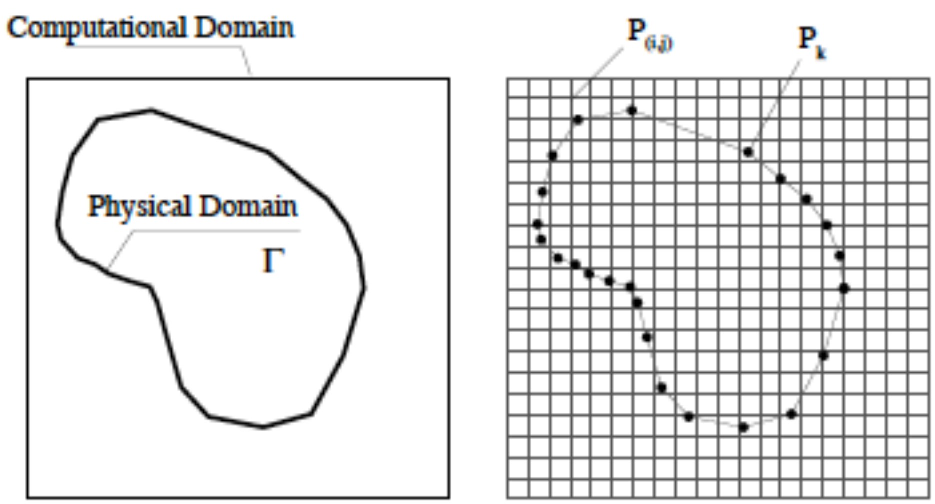
1. The physical domain is the heart: Γ
   The computational domain is a box.



IB Methods: the origin

2. A Cartesian, uniform grid covers the physical domain
   The Navier-Stokes Equations in Eulerian form are solved using a finite difference techniques

$$
\begin{aligned}
\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{1}{\rho} \nabla p - \frac{\mu}{\rho} \nabla^2 \vec{u} &= \boxed{\vec{f}_m(\vec{x}, t)} \\
\nabla \cdot \vec{u} &= 0
\end{aligned}
$$

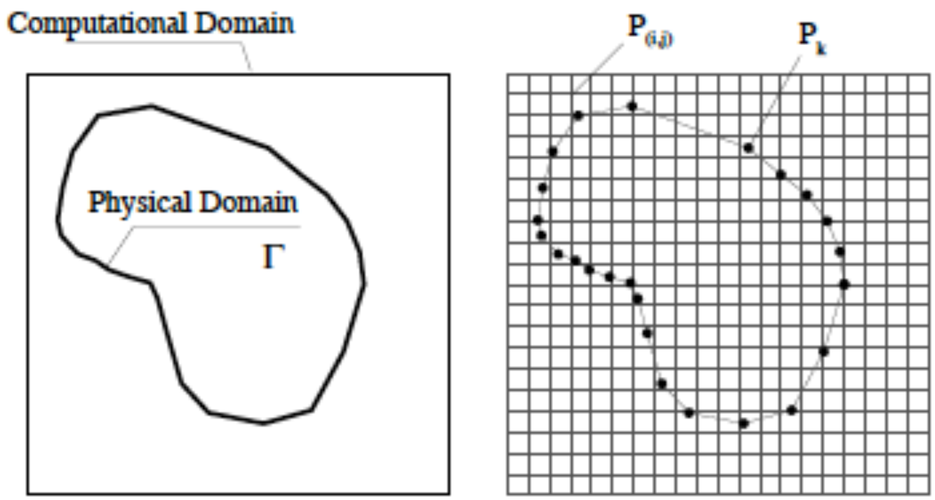


$\boxed{\vec{f}_m(\vec{x}, t)}$ is the force exerted by the heart on the fluid (at every location in space)

IB Methods: the origin

3. A set of points describing the heart walls are advected using a Lagrangian method and the local fluid velocity

$$\frac{\partial \vec{X}_k}{\partial t} = \boxed{\vec{u}(\vec{X}_k, t)}$$

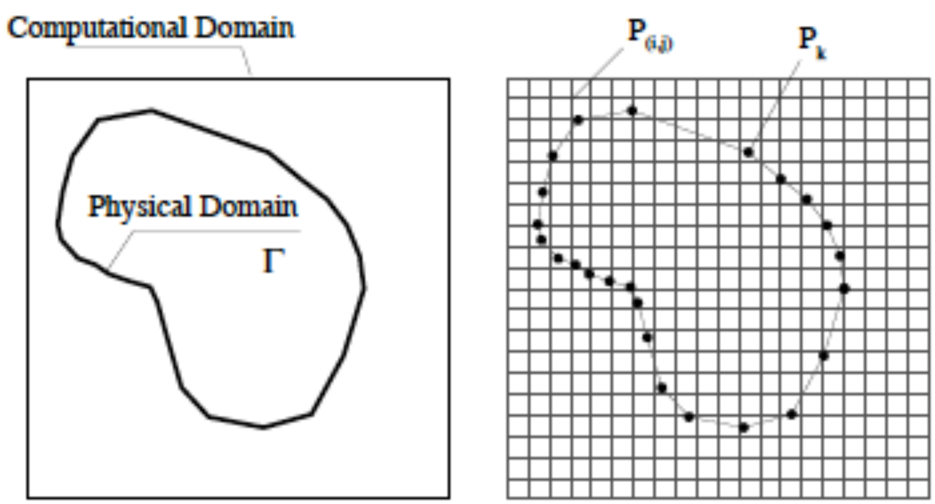


$\boxed{\vec{u}(\vec{X}_k, t)}$ is the velocity of the fluid at the location of the point k

IB Methods: the origin

4. Define the coupling between the heart and the fluid

$$\vec{f}_m(\vec{x},t) = \sum_k \vec{F}_k(t)\delta(|\vec{x} - \vec{X}_k|)$$



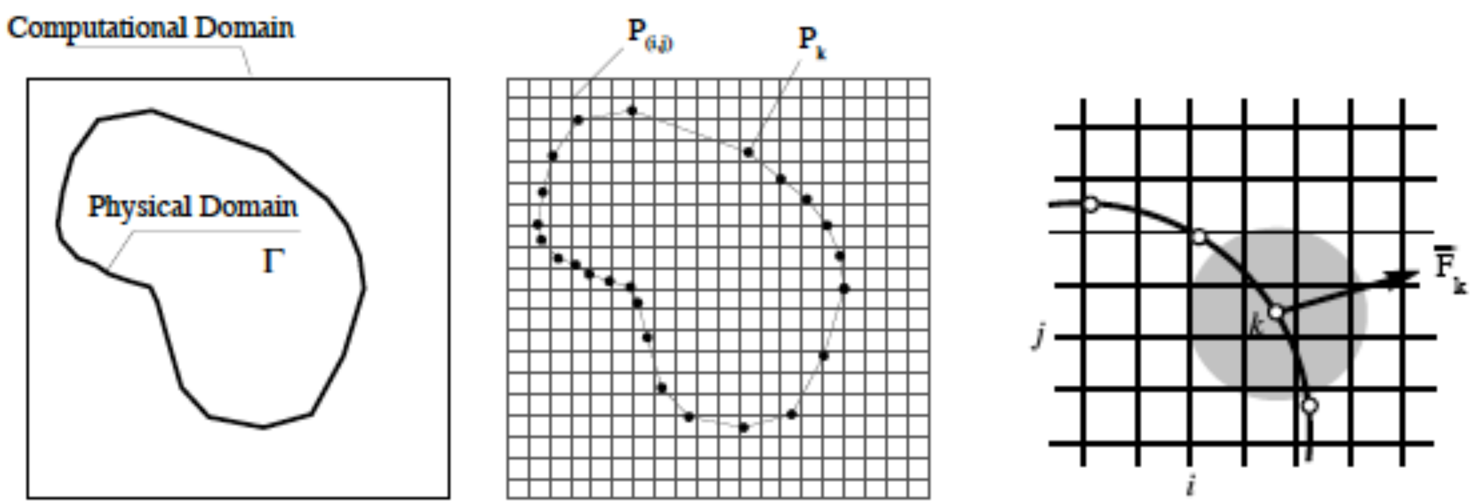
The coupling force depends on the model assumed for the hear wall:

- Points
- Fibers (Peskin)
- Membranes

IB Methods: the origin

5. Transfer the force from the heart to the fluid

$$\vec{f}_m(\vec{x},t) = \sum_k \vec{F}_k(t)\delta(|\vec{x} - \vec{X}_k|)$$



IB Methods: the forcing function

$$\vec{f}_m(\vec{x},t) = \sum_k \boxed{\vec{F}_k(t)} \delta(|\vec{x} - \vec{X}_k|)$$

In Peskin's approach it is derived directly from Hooke's law

For rigid boundaries various choices have been proposed:

$$\vec{F}_k(t) = -\kappa(\vec{X}_k - \vec{X}_k^e(t))$$ Peskin & Lai (2000)

$$\vec{F}_k(t) = (\mu/K)\vec{u}.$$ Angot *et al.* (1998)

$$\vec{F}_k(t) = \alpha \int_0^t \vec{u}(\tau)d\tau + \beta\vec{u}(t)$$ Goldstein *et al.* (1993)

IB Methods: the forcing function

Peskin's IB approach is well suited for elastic boundaries

Standard forcing terms become ill-behaved in the rigid limit

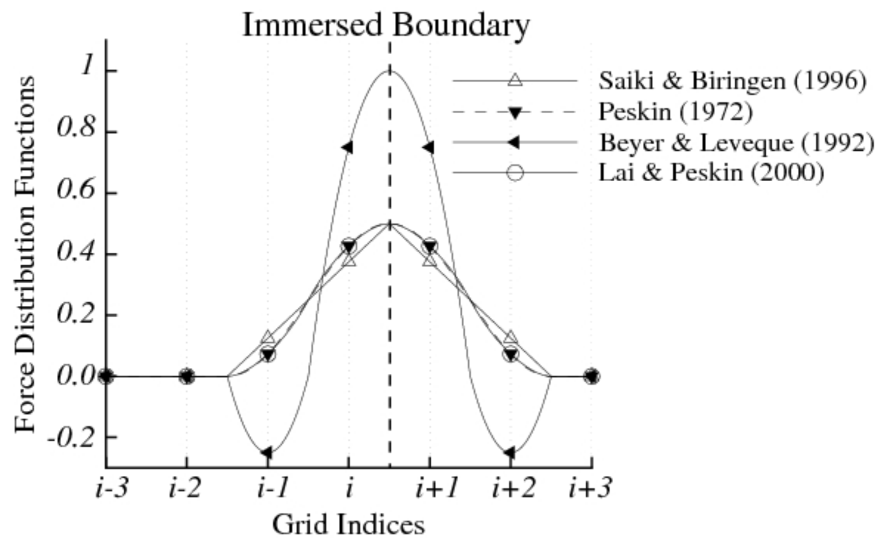
Ad-hoc forcing terms (i.e. porosity) tend to be inaccurate and unstable

Definition of the forcing terms for turbulent quantities (as in RANS) is extremely challenging

Solution is required inside solid bodies

IB Methods: the transfer function

$$\vec{f}_m(\vec{x},t) = \sum_k \vec{F}_k(t)\,\delta(|\vec{x} - \vec{X}_k|)$$



Again various choices have been proposed:

$$\delta(r) \approx d_h(r) = \frac{1}{h}\begin{cases} (cos(\pi r/2h)+1)/4h, & \text{if } r \le 2h, \\ 0 & \text{otherwise.} \end{cases}$$

Peskin (1972)

$$\delta(r) \approx d_h(r) = \frac{1}{h}\begin{cases} (2h+r)/4h^2, & \text{if } r \le 2h, \\ 0 & \text{otherwise.} \end{cases}$$

Peskin & Lai (2000)

$$\delta(r) \approx d_h(r) = \frac{1}{h}\begin{cases} 1-(r/h)^2, & \text{if } r \le h, \\ 2-3r/h+(r/h)^2, & h \le r \le 2h, \\ 0 & \text{otherwise.} \end{cases}$$

Beyer & Leveque (1992)

IB Methods: the forcing function

The transfer of the forcing functions in Peskin's approach implies a non-sharp representation of the boundary which is not appropriate for boundary layers and heat transfer problems

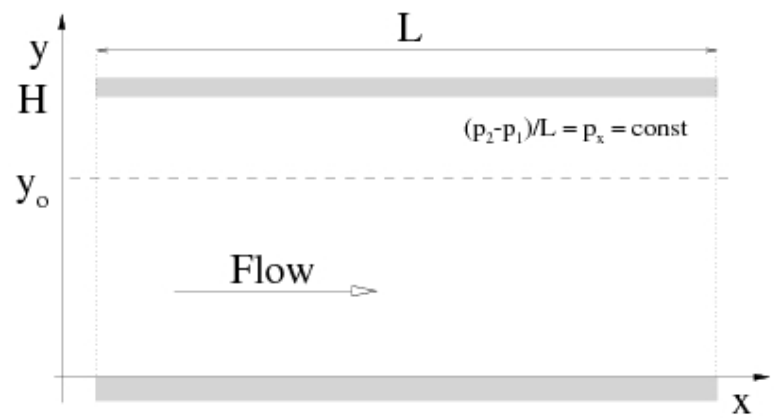
Direct reference to the grid size h make the force not well suited for general locally refined grids

Some formulations result in forces that do NOT remain positive

…but does this matter?

A simple example

Channel flow with a fixed horizontal membrane



Navier-Stokes Equations

$$\frac{\mu}{\rho}\frac{d^2U}{dy^2} = \frac{1}{\rho}p_x + F\delta(y - y_o)$$

Boundary Conditions

$$\begin{aligned} U(y=0) &= 0 \\ U(y=H) &= 0 \\ U(y=y_o) &= 0 \end{aligned}$$

A simple example

Channel flow with a fixed horizontal membrane

Exact solution (2 parabolic flows)

$$U(y) = \begin{cases} (y/2\mu)(y-H)p_x - Fy(1-y_o/H)(\rho/\mu) & \text{if } y \le y_o, \\ (y/2\mu)(y-H)p_x - Fy_o(1-y/H)(\rho/\mu) & \text{otherwise.} \end{cases}$$

A simple example

Channel flow with a fixed horizontal membrane

Exact solution (2 parabolic flows)

$$U(y) = \begin{cases} (y/2\mu)(y-H)p_x - Fy(1-y_o/H)(\rho/\mu) & \text{if } y \leq y_o, \\ (y/2\mu)(y-H)p_x - Fy_o(1-y/H)(\rho/\mu) & \text{otherwise.} \end{cases}$$

From the boundary conditions we can obtain: $F = -Hp_x/2\rho$

A simple example

Channel flow with a fixed horizontal membrane

Exact solution (2 parabolic flows)

$$U(y) = \begin{cases} (y/2\mu)(y-H)p_x - Fy(1-y_o/H)(\rho/\mu) & \text{if } y \le y_o, \\ (y/2\mu)(y-H)p_x - Fy_o(1-y/H)(\rho/\mu) & \text{otherwise.} \end{cases}$$

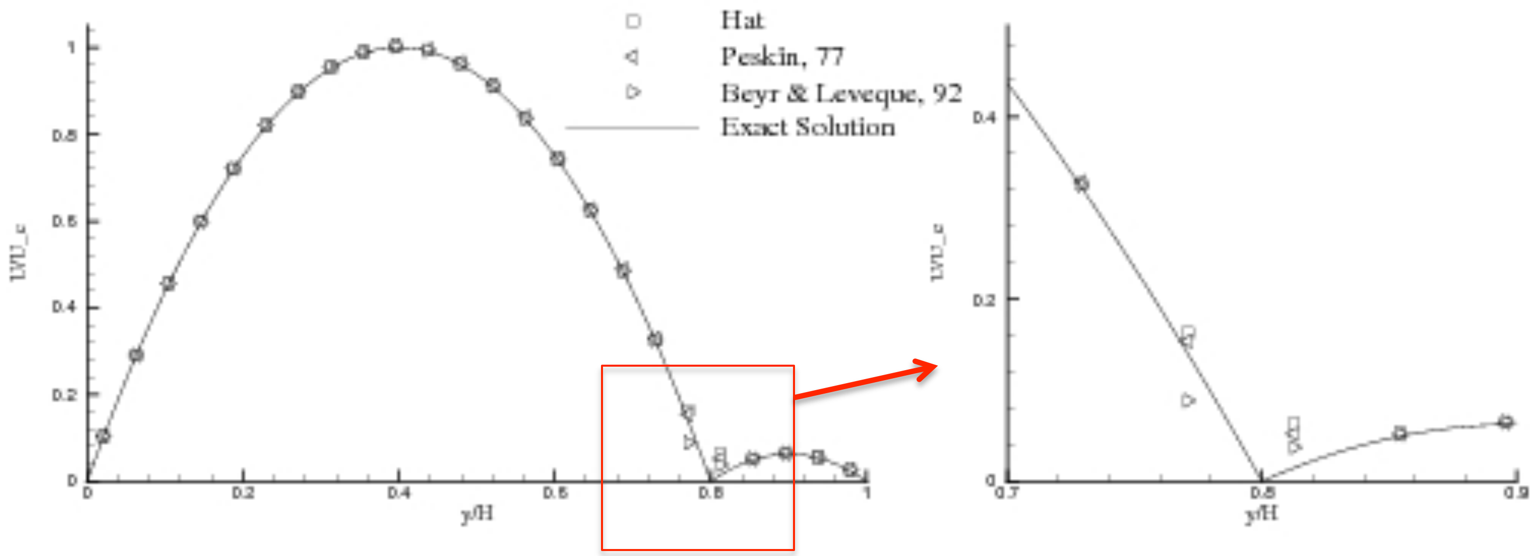
From the boundary conditions we can obtain: $F = -Hp_x/2\rho$

Note: velocity is continuous but not its derivative $\mu/\rho[dU/dy] = F.$

The effect of the force transfer

Channel flow with a fixed horizontal membrane

Use the exact force and various form of the transfer function (discrete delta function)



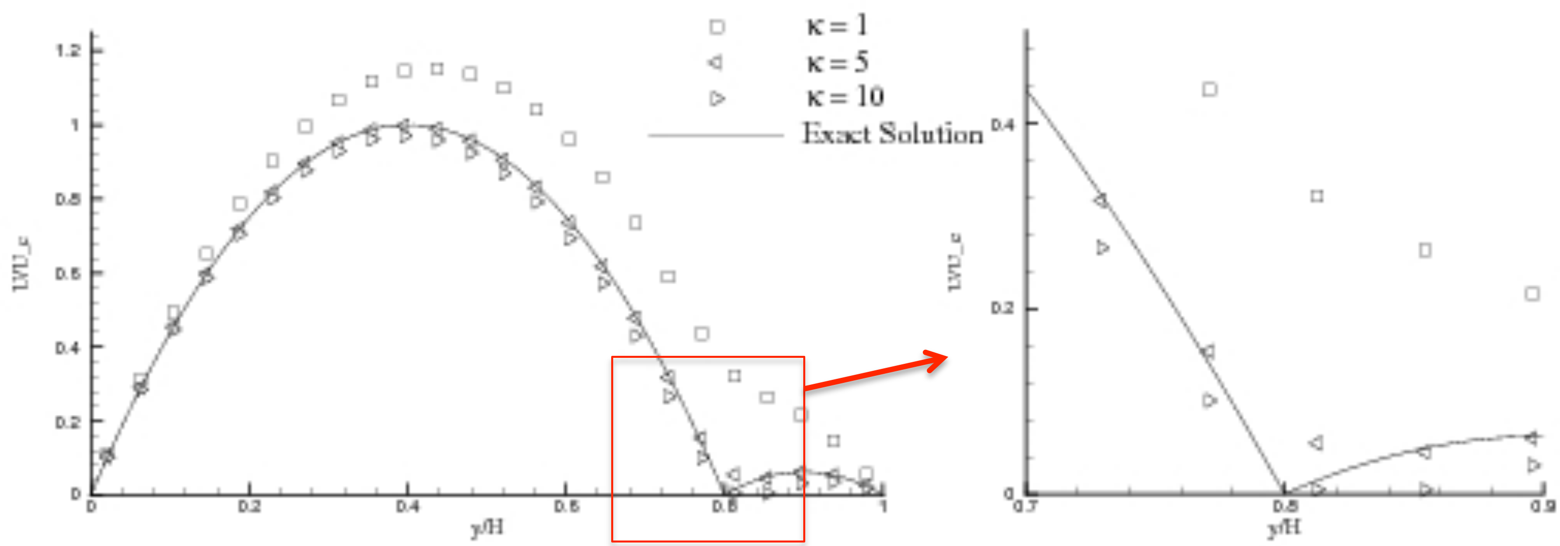
The effect of the forcing

Channel flow with a fixed horizontal membrane

Approximate the force and use the BL transfer function

$$\vec{F}_k(t) = (\mu/K)\vec{u}.$$

Angot *et al.* (1998)



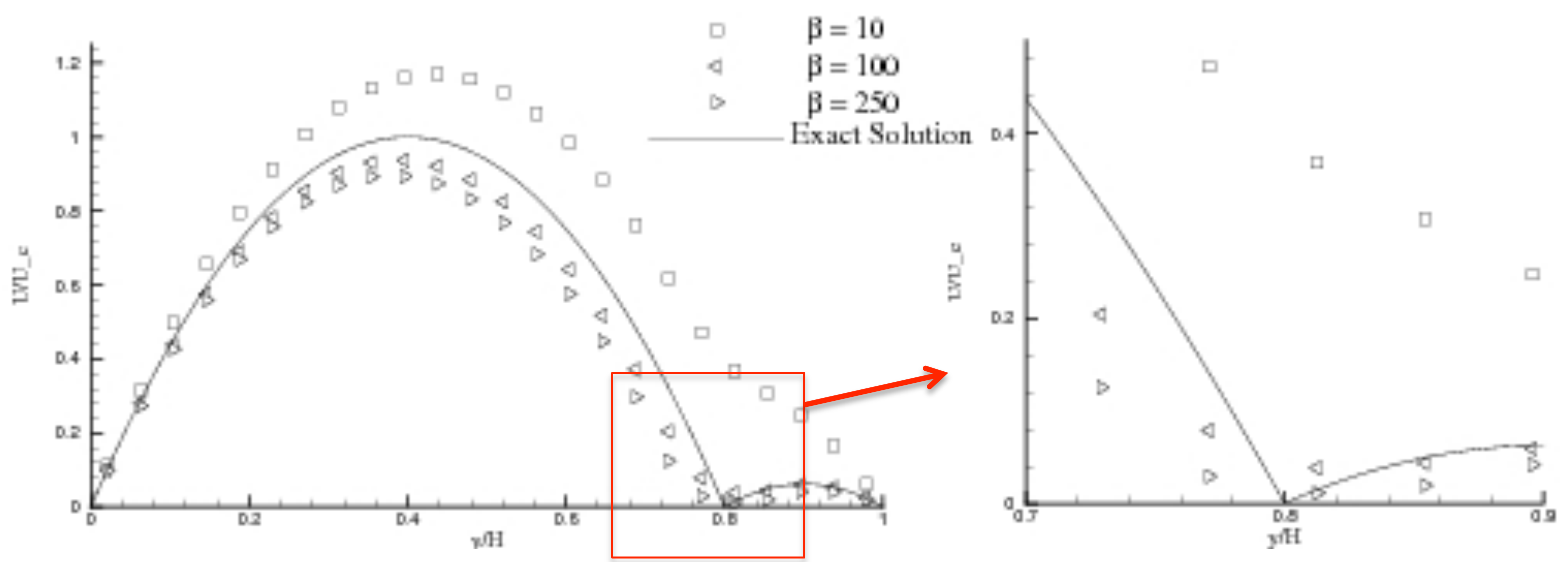
The effect of the forcing

Channel flow with a fixed horizontal membrane

Approximate the force and use the BL transfer function

$$\vec{F}_k(t) = \alpha \int_0^t \vec{u}(\tau) d\tau + \beta \vec{u}(t)$$

Goldstein *et al.* (1993)



Summary

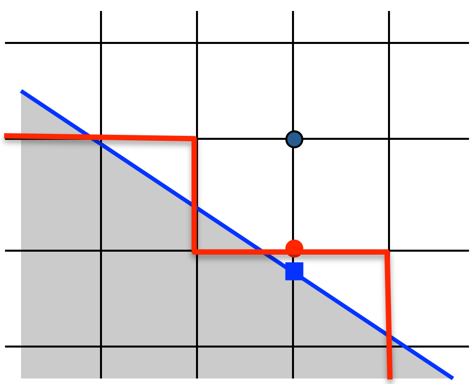
Both the choice of the forcing function and the solid/fluid transfer are important

In general it is difficult to distinguish between the errors introduced by each step

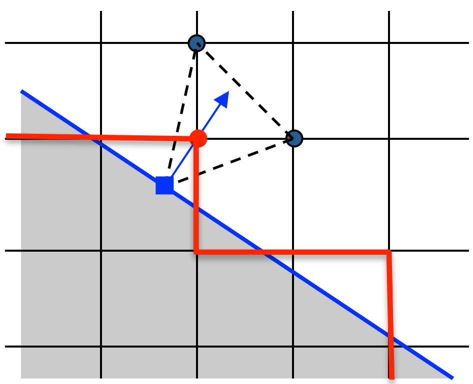
An additional difficulty is to ensure that conservation properties are preserved at the IB (mass conservation, etc.)

An alternative IB method

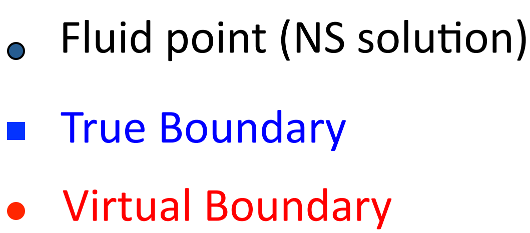
Instead of representing the fluid/solid coupling as a continuous force, we reconstruct a "new" virtual boundary condition on a modified domain



1d reconstruction



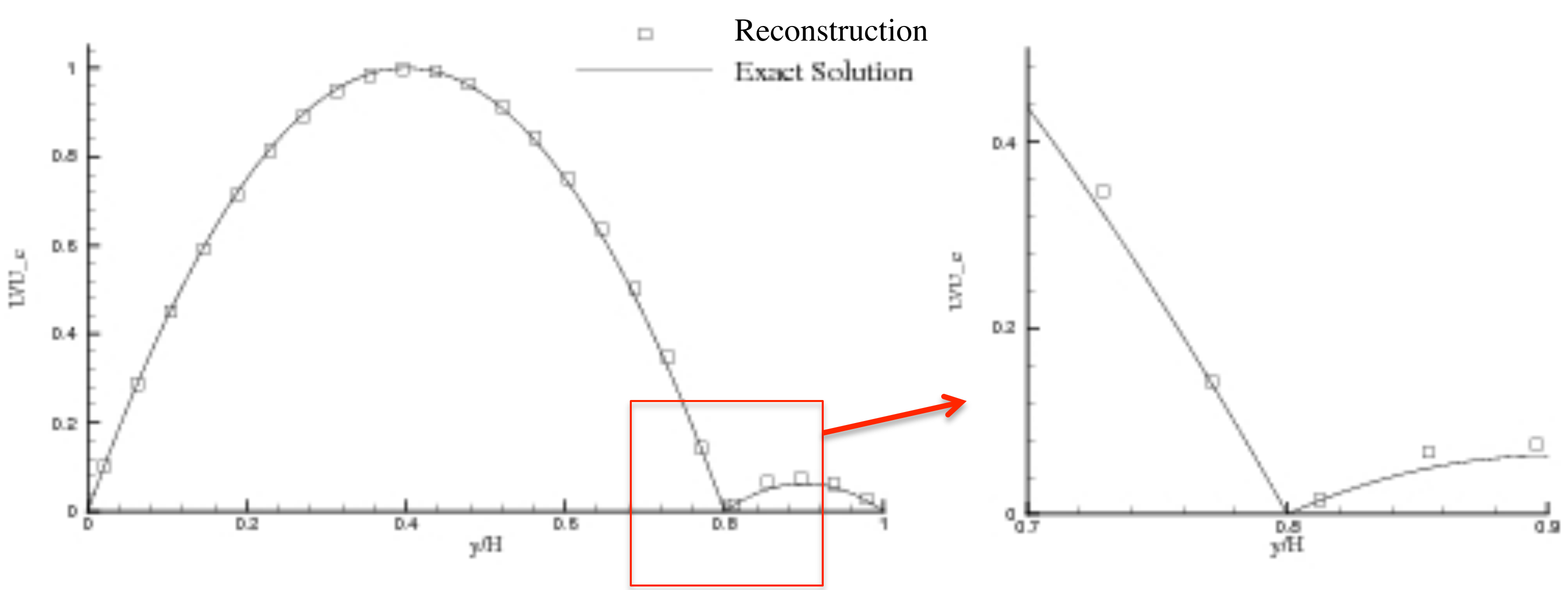
Multi-d reconstruction



The effect of the reconstruction

Channel flow with a fixed horizontal membrane

1D Linear Reconstruction Fadlun *et al.* (2000)



Reconstruction vs. Forcing

Reconstruction is applied discretely, therefore can be "synchronized" with the discretization scheme (need to prove)

Physical constraints can be added to the reconstruction, e.g. mass conservation, turbulent wall functions, etc. (need to prove)

Reconstructions are local and do not require uniform meshes and solid walls (or moving walls) do not require any special treatment (need to prove)

3. Geometry and Grid Generation

Solid/Fluid Thermal Coupling

Using the Immersed Boundary Method

Components

Description of the True Boundary

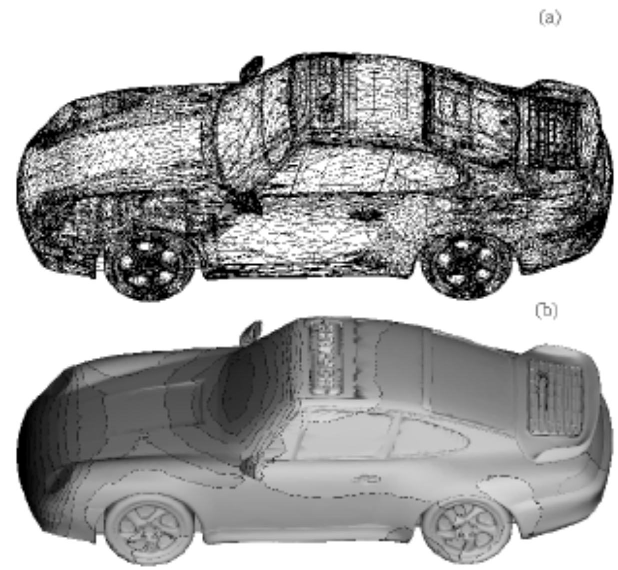
Definition of the Virtual Boundary

Grid Refinement Criteria

Handling Imperfect CAD Parts (digression – if time permits)

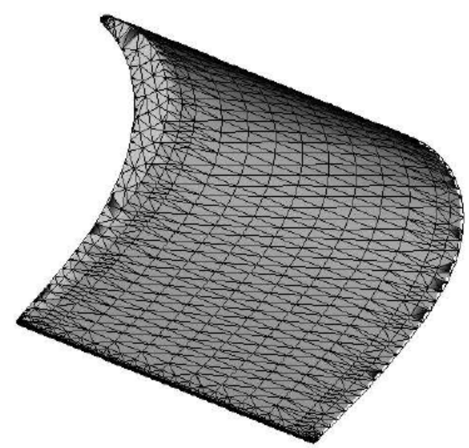
Description of the True Boundary

Stereolithography Surfaces (STL)

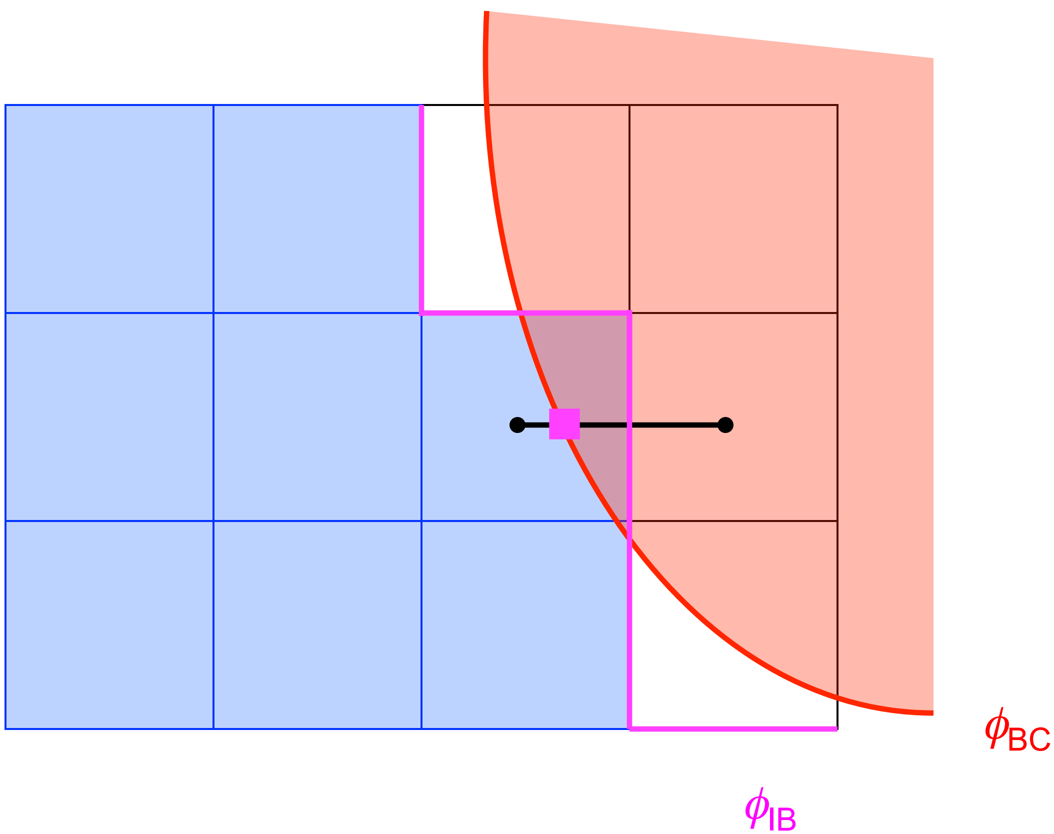


Advantages

- A set of triangles
- No high-order topological info
- Imperfections are tolerated (intersections, overlaps, etc.)



Description of the Virtual Boundary

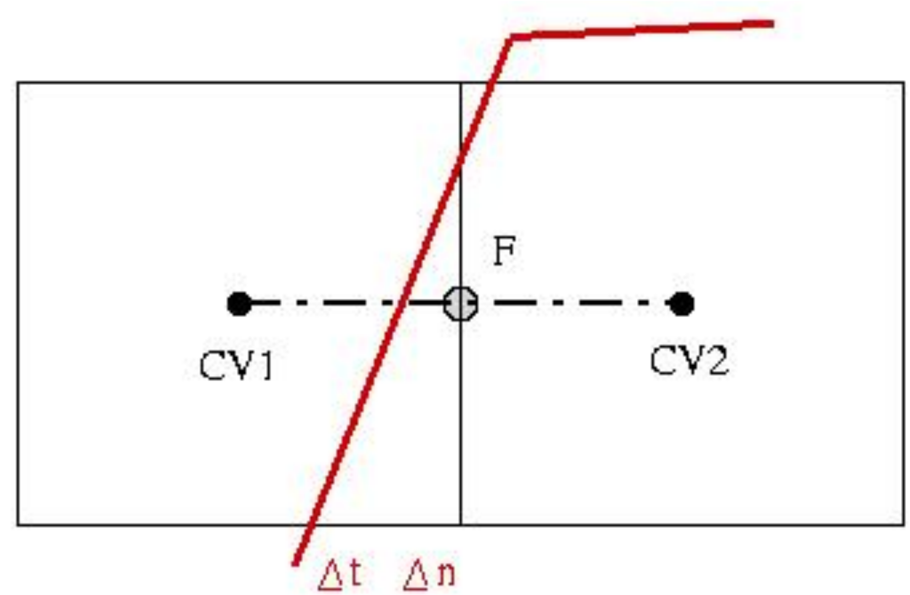


The relation between ϕ_{BC} and ϕ_{IB} involves geometrical quantities (e.g. surface normal) and physical constraints (e.g. wall model, conservation laws)

Grid Refinement Criteria

The underlying meshes are locally refined – unstructured

Given the desired resolution at the STL surface (Δn, Δt) the intersections are used to detect the CVs to refine

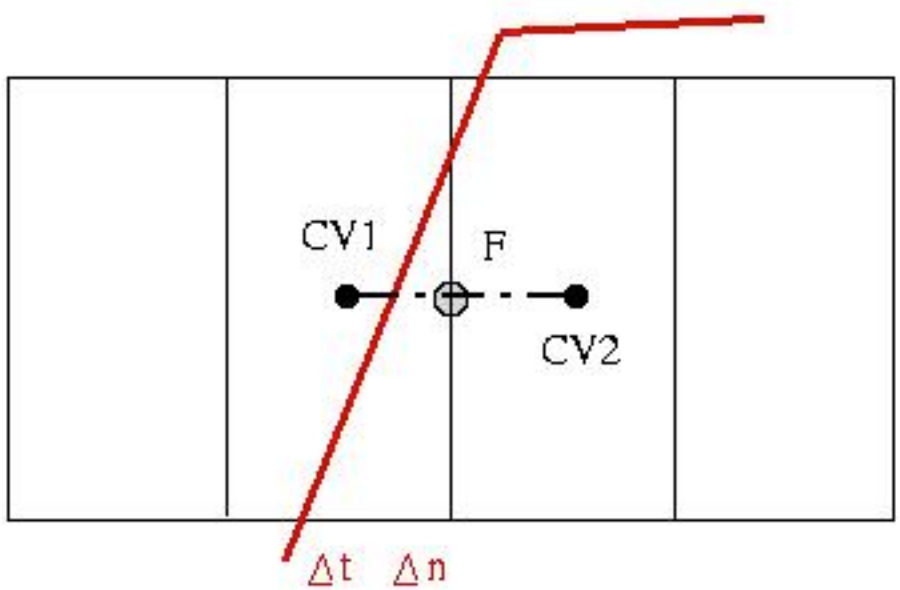


$$\Delta x_i^{CV} = MIN\left(\frac{\Delta n}{|n_i^{STL}|_i};\Delta t\right)$$

Grid Refinement Criteria

The underlying meshes are locally refined – unstructured

Given the desired resolution at the STL surface (Δn, Δt) the intersections are used to detect the CVs to refine

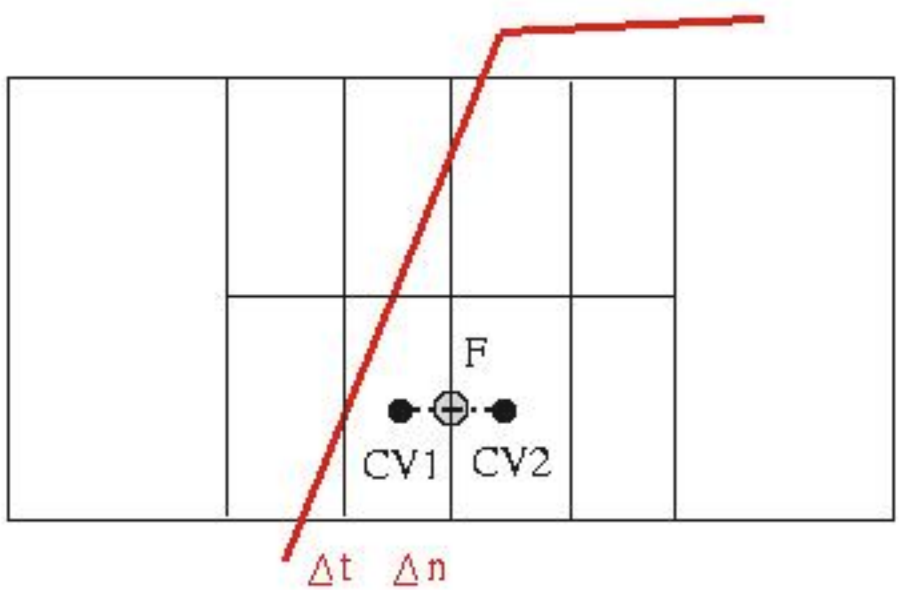


$$\Delta x_i^{CV} = MIN\left(\frac{\Delta n}{|n_i^{STL}|_i};\Delta t\right)$$

Grid Refinement Criteria

The underlying meshes are locally refined – unstructured

Given the desired resolution at the STL surface (Δn, Δt) the intersections are used to detect the CVs to refine

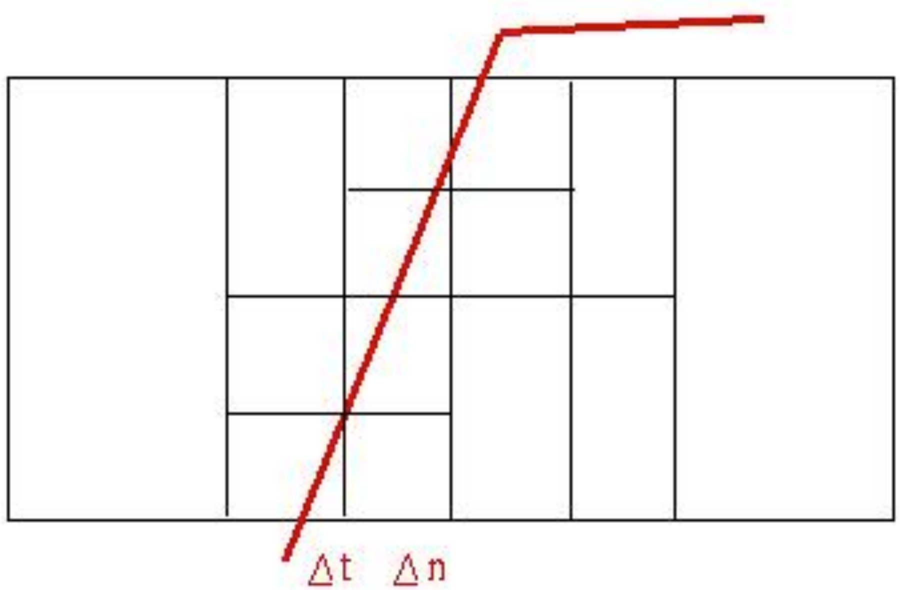


$$\Delta x_i^{CV} = MIN\left(\frac{\Delta n}{|n_i^{STL}|_i};\Delta t\right)$$

Grid Refinement Criteria

The underlying meshes are locally refined – unstructured

Given the desired resolution at the STL surface (Δn, Δt) the intersections are used to detect the CVs to refine

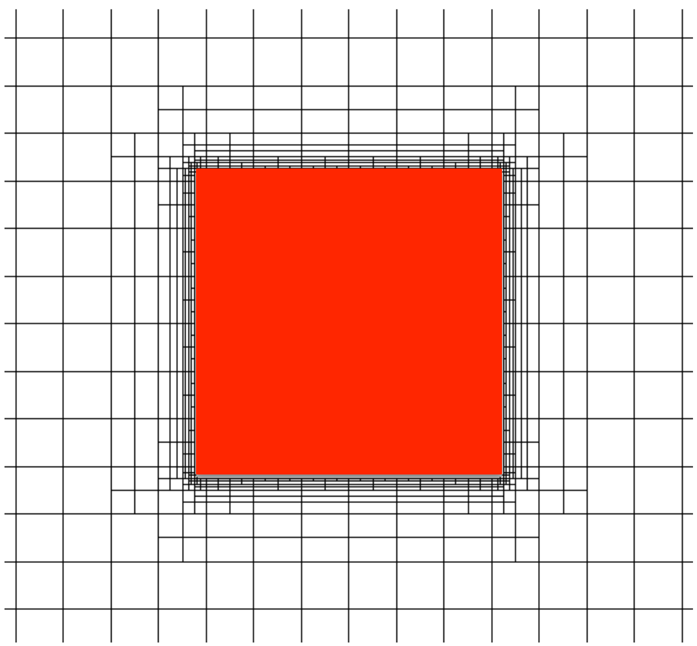


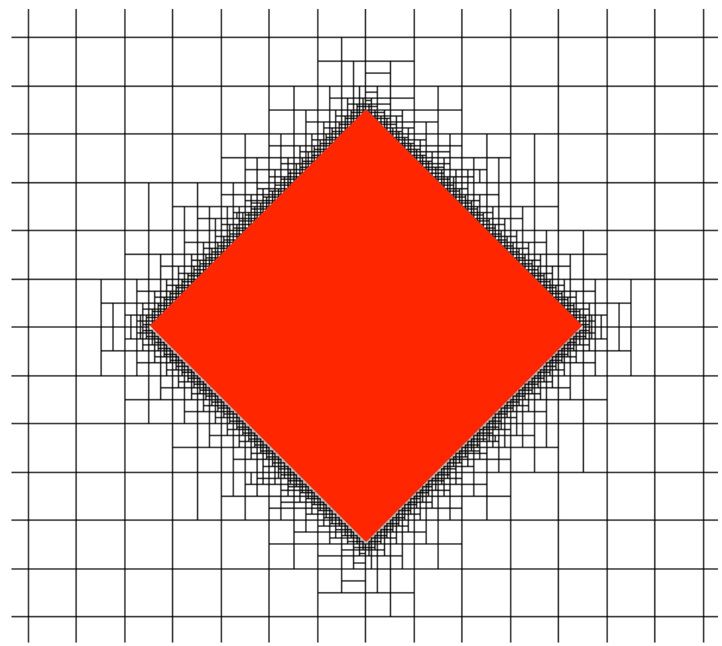
$$\Delta x_i^{CV} = MIN\left(\frac{\Delta n}{|n_i^{STL}|_i};\Delta t\right)$$

Grid Refinement Criteria

The underlying meshes are locally refined – unstructured

 (Δn, Δt) anisotropy results in reduced grid size for grid aligned STLs

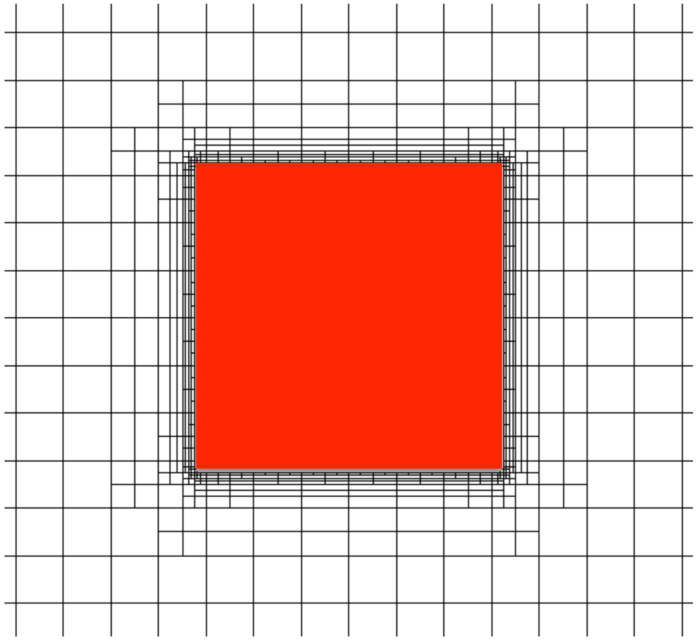


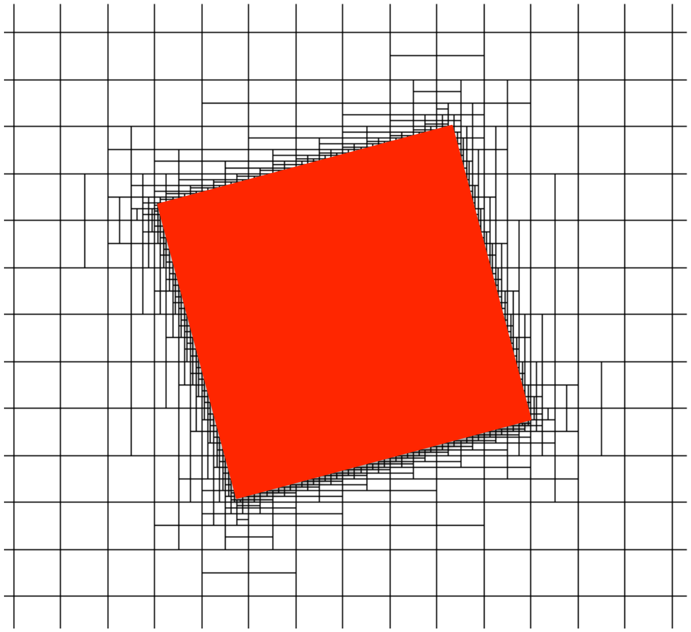


Grid Refinement Criteria

The underlying meshes are locally refined – unstructured

(Δn, Δt) anisotropy results in reduced grid size for grid aligned STLs



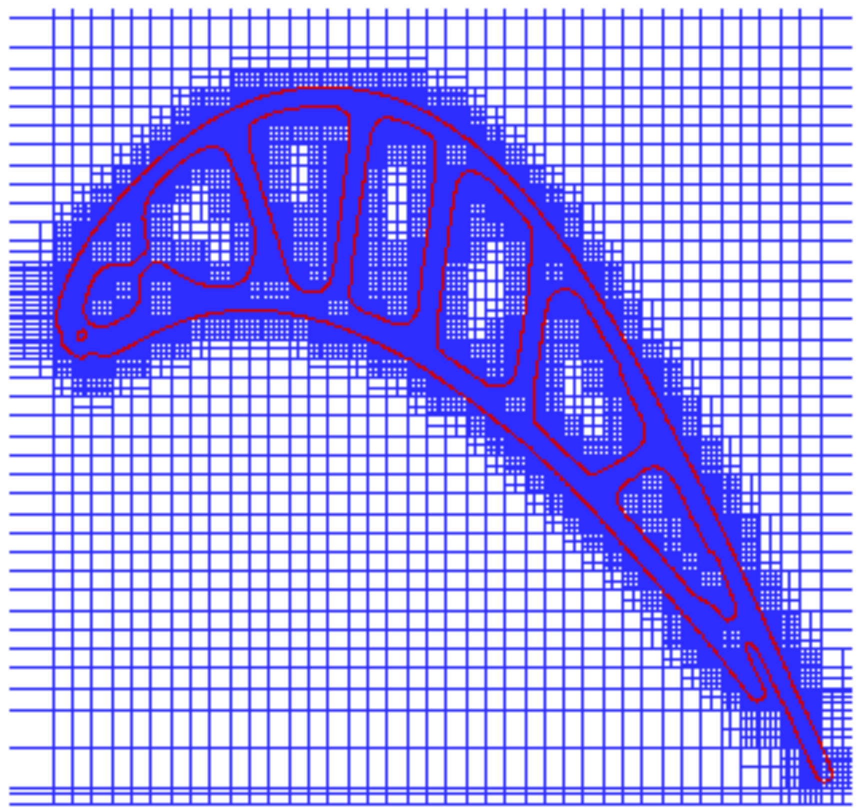


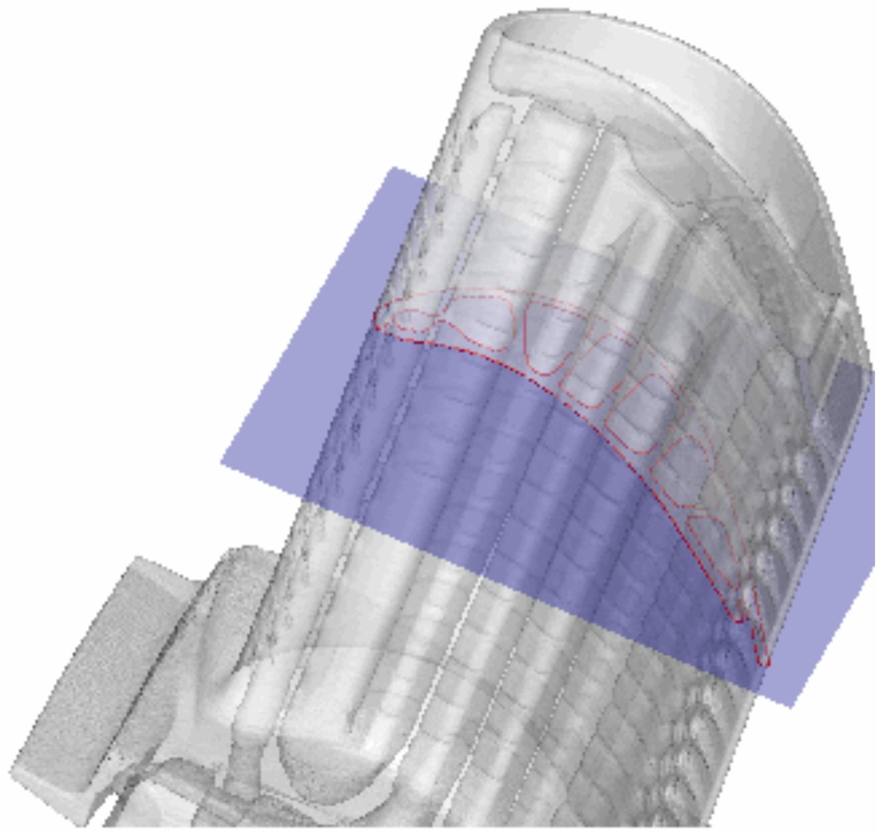
A grid generation example

Turbine Blade

Horizontal Grid Cuts

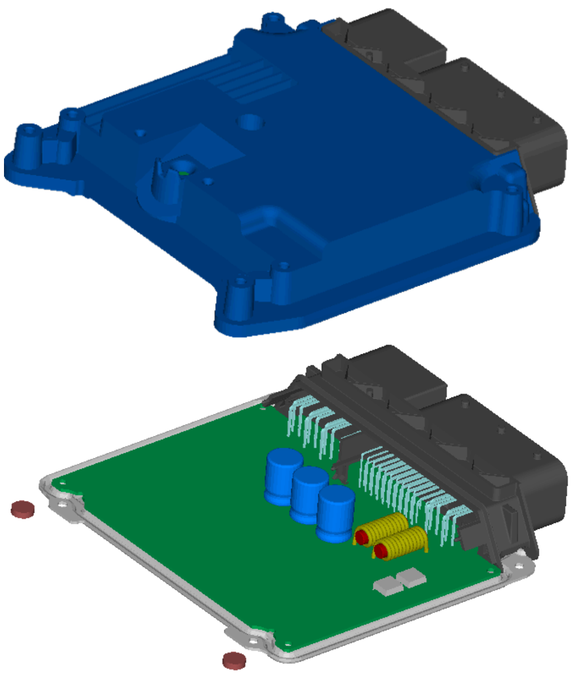
Grid Generated using 4 Iterations of Ray Tracing



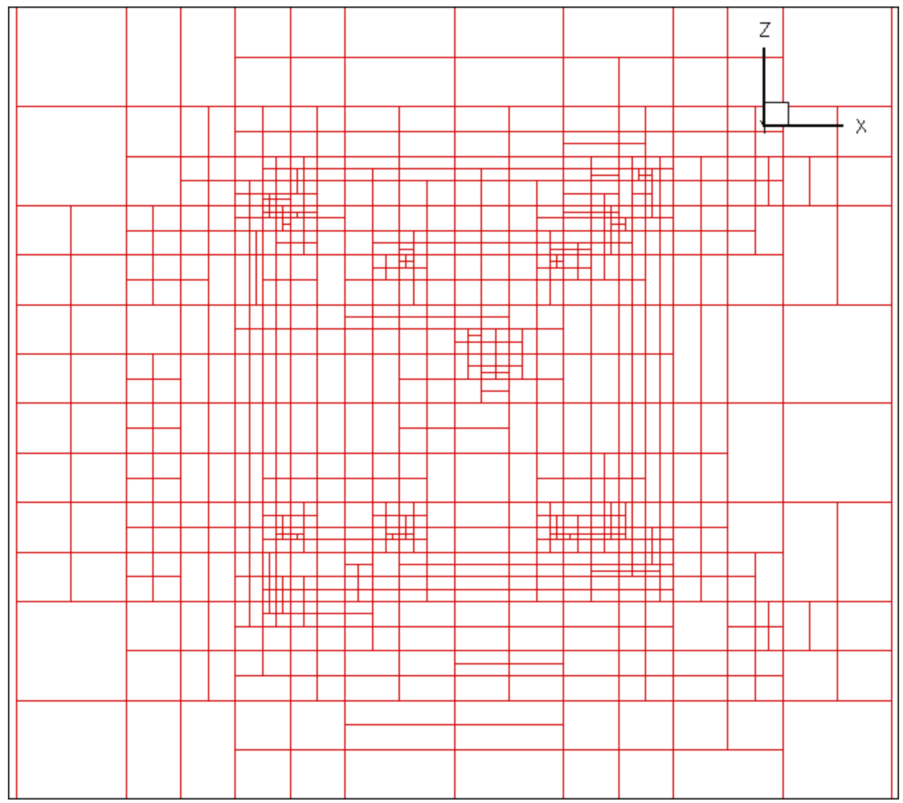


A grid generation example

Electronic Component Unit

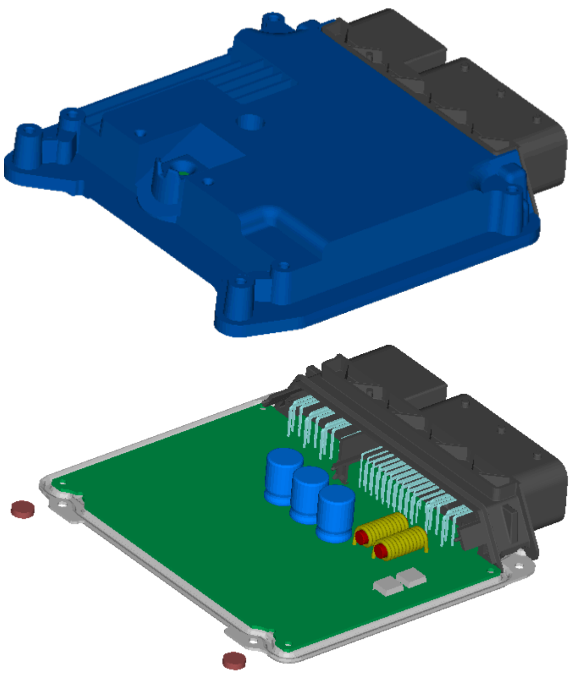


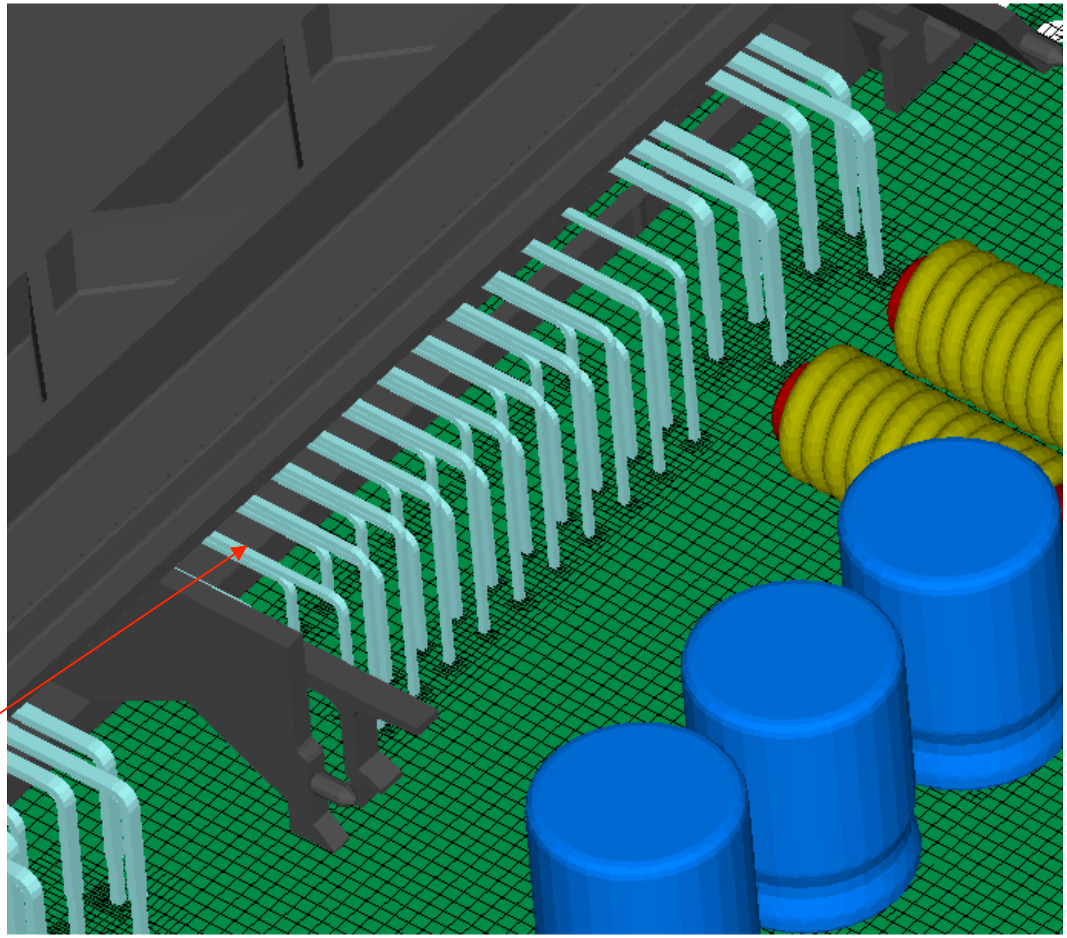
Horizontal Grid Cuts
Grid Generated using 6 Iterations of Ray Tracing



A grid generation example

Electronic Component Unit





Every single
pin is captured

Summary

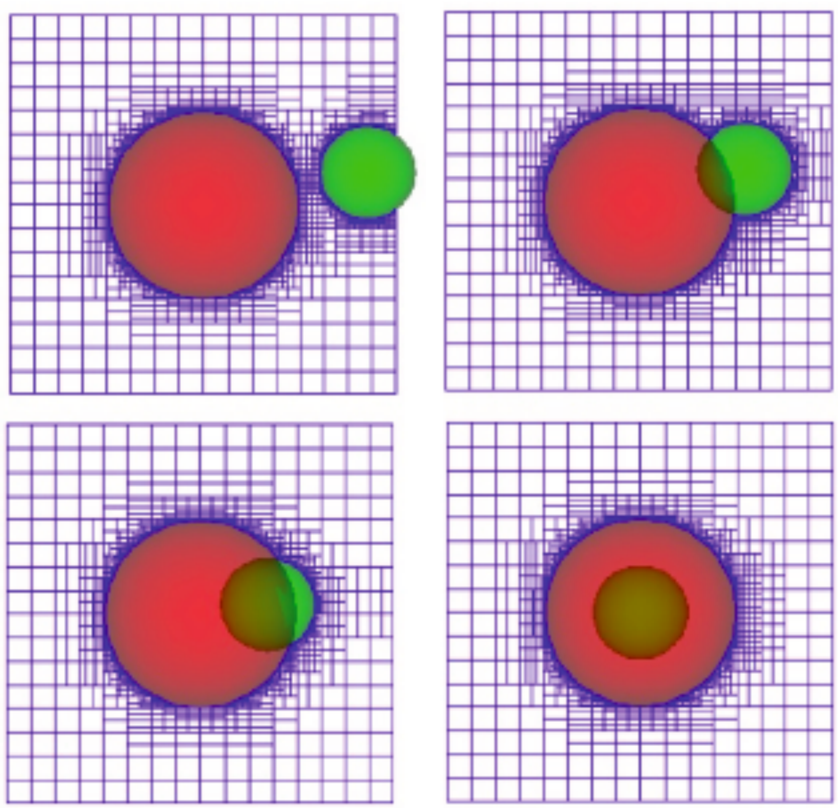
Combination of STL surfaces (triangles) and ray tracing provides a flexible infrastructure to generate the virtual boundary and the underlying grid

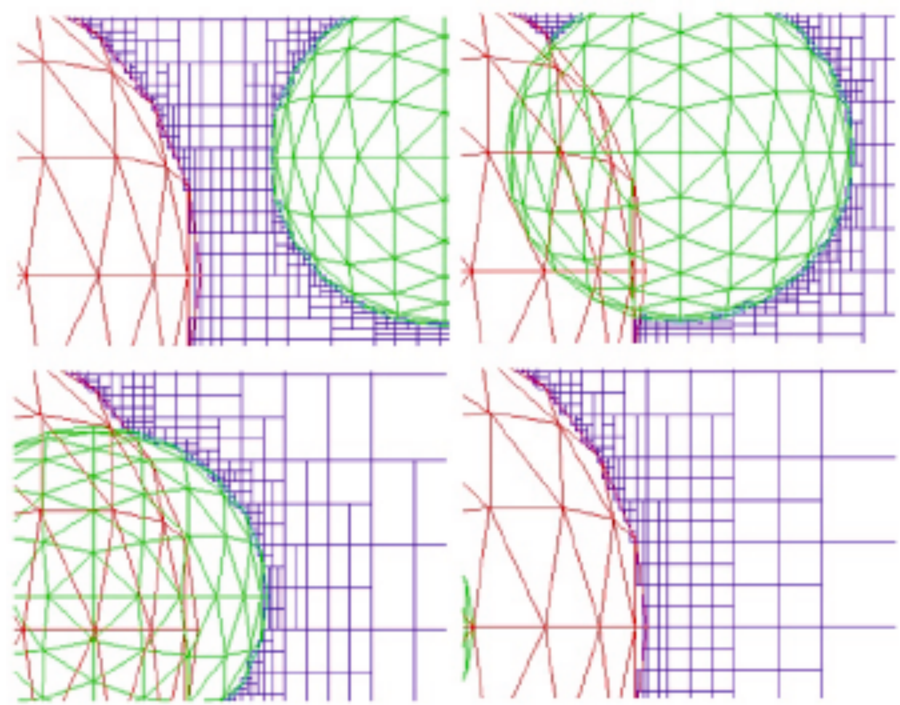
Unstructured grids with local grid refinement are efficient in achieving desired tangential and normal resolution

Unstructured environment is flexible in creating cylindrical, Cartesian or general curvilinear grids

Handling Imperfect CAD Parts

Ray tracing is applied locally at each face (CV-CV segment), therefore it provides ample flexibility



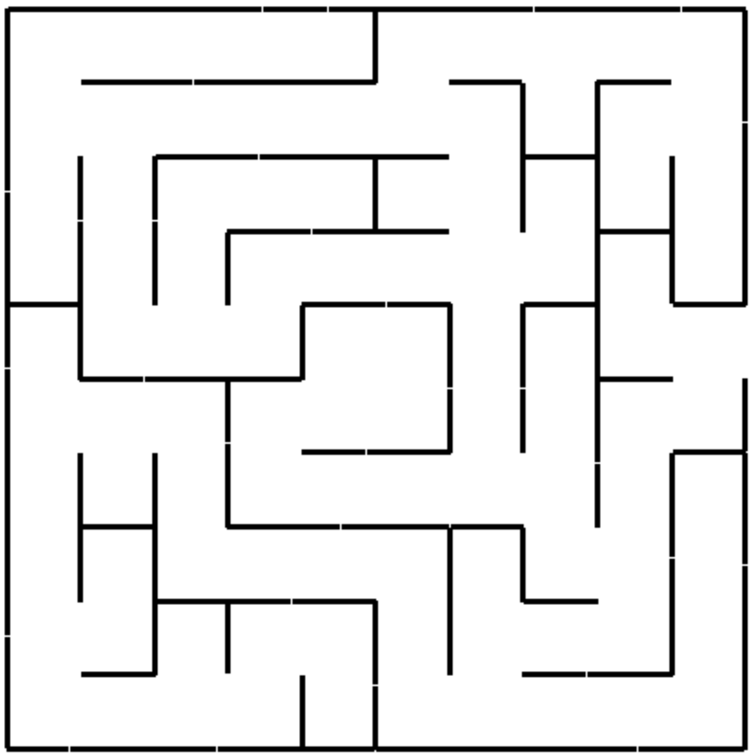


(short) digression…

Find Leakages

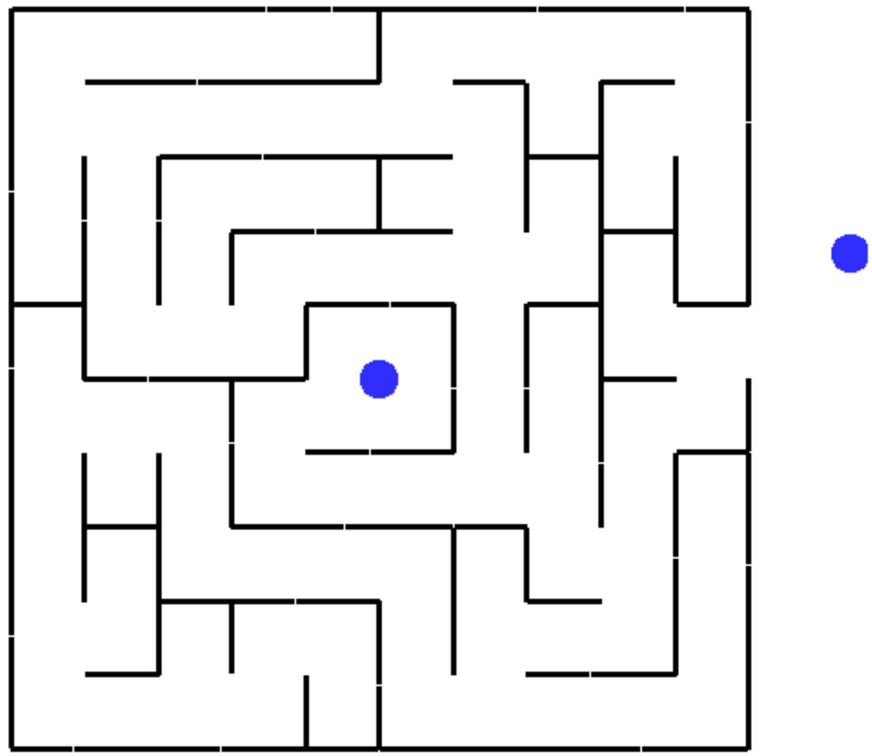
Water-tight surfaces are necessary for solid/fluid simulations
How to detect leakages?

It is like solving a 3D labyrinth



Find Leakages

Water-tight surfaces are necessary for solid/fluid simulations
How to detect leakages?



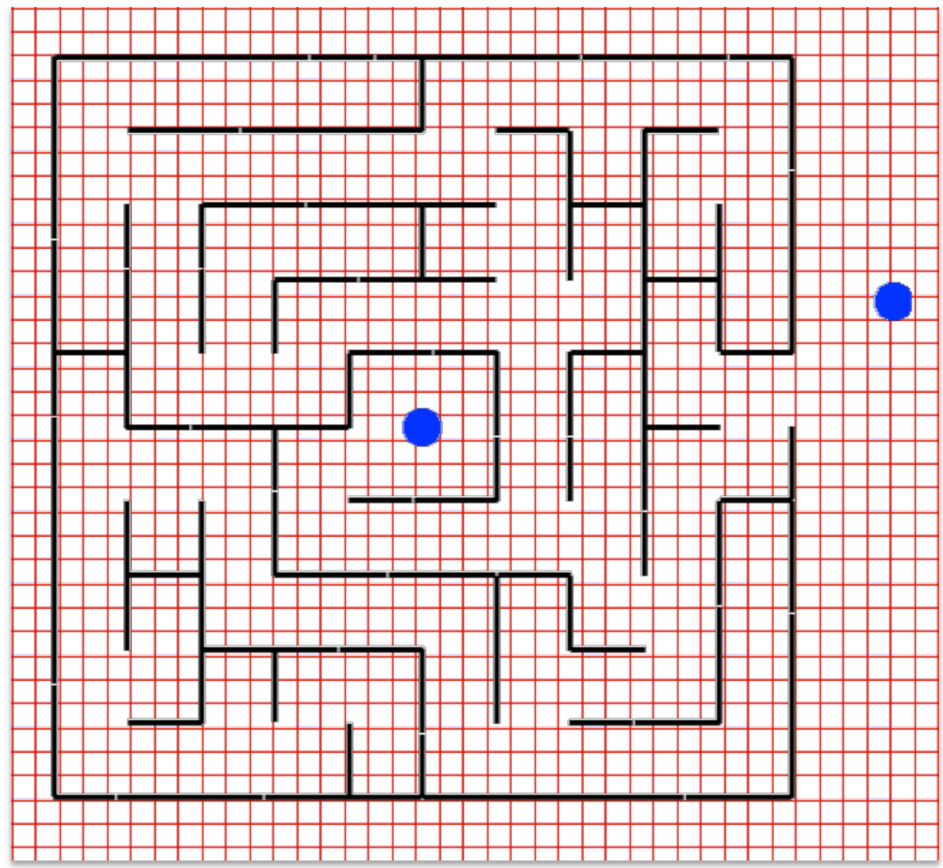
It is like solving a 3D labyrinth
Find the path…

Find Leakages

Water-tight surfaces are necessary for solid/fluid simulations

How to detect leakages?

Step 1: create a volume grid and identify all the face intersection (ray tracing)



Find Leakages

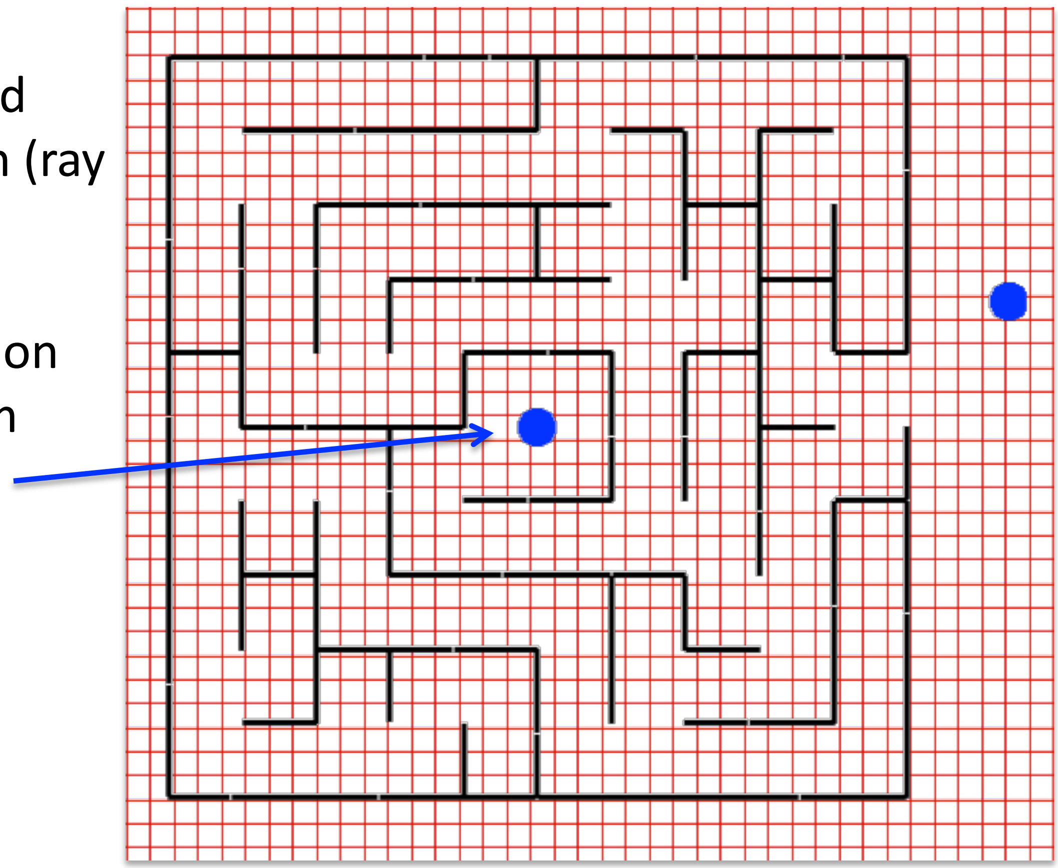
Water-tight surfaces are necessary for solid/fluid simulations

How to detect leakages?

Step 1: create a volume grid and identify all the face intersection (ray tracing)

Step 2: solve the eikonal equation with a front marching algorithm from the center

$$|\nabla u| = 1$$



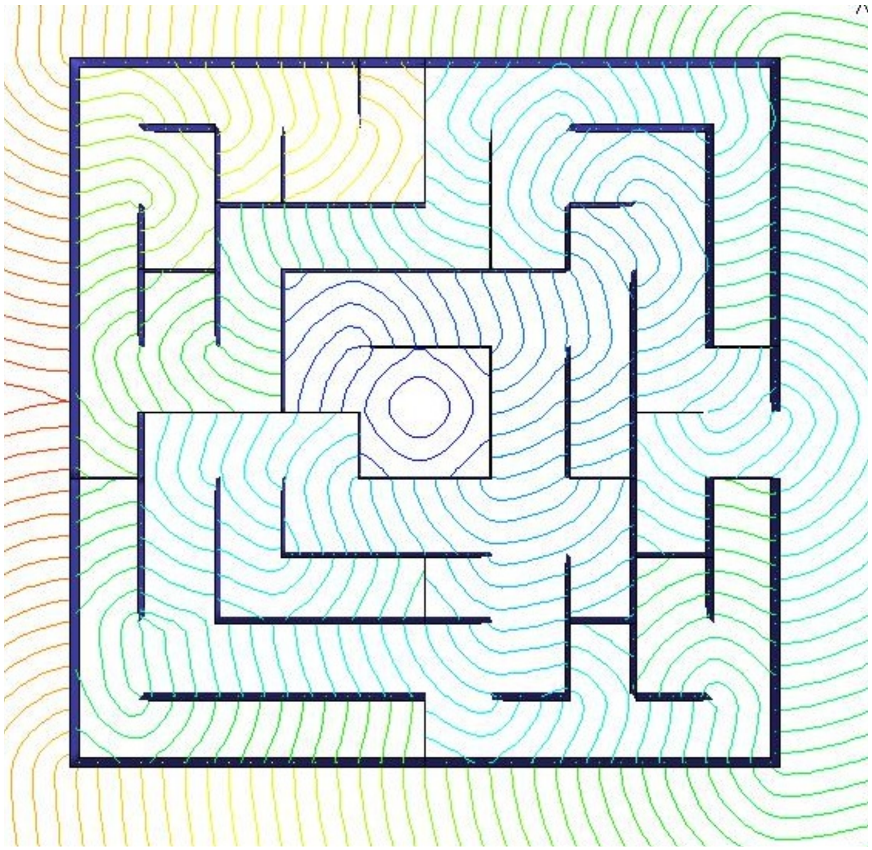
Find Leakages

Water-tight surfaces are necessary for solid/fluid simulations
How to detect leakages?

Step 1: create a volume grid and identify all the face intersection (ray tracing)

Step 2: solve the eikonal equation with a front marching algorithm from the center

$$|\nabla u| = 1$$



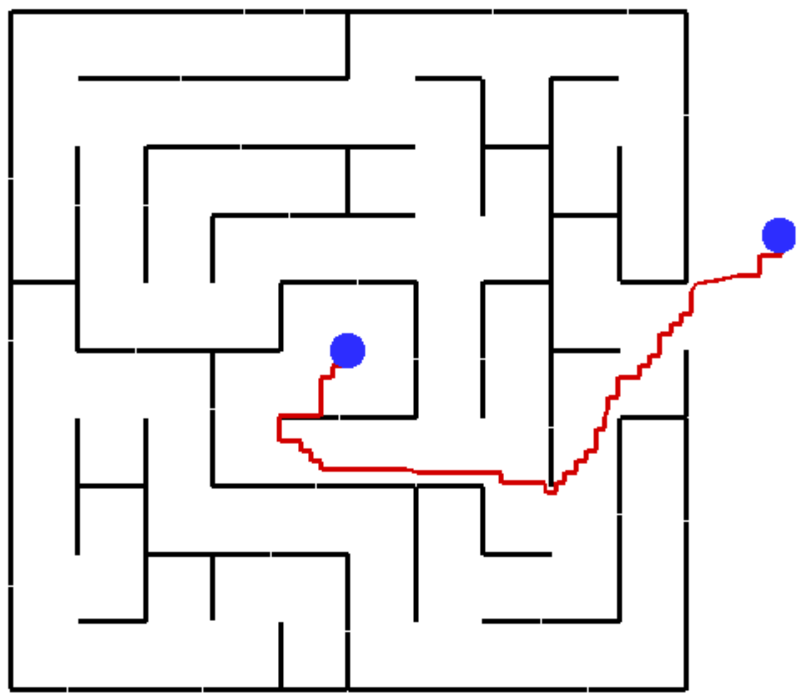
Find Leakages

Water-tight surfaces are necessary for solid/fluid simulations
How to detect leakages?

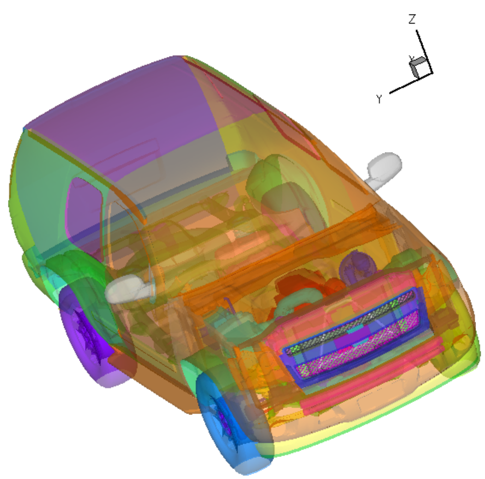
Step 1: create a volume grid and identify all the face intersection (ray tracing)

Step 2: solve the eikonal equation with a front marching algorithm from the center

Step 3: navigate the field and create a leakage path…



An example: Find Leakages



For a full car assembly the algorithm is able to automatically detect a leakage path associated to the tire and traced back to the absence of valve stems and caps from the part database!