DE LA RECHERCHE À L'INDUSTRIE

SIMULATION FOR FAST TRANSIENT DYNAMICS WITH FLUID STRUCTURE INTERACTION

SOFTWARE ARCHITECTURE FOR NEXT GENERATION SUPERCOMPUTERS

EUROPLEXUS Software

ARISTOTE Seminar
Vincent FAUCHER
CEA/DEN/DANS/DM2S/SEMT/DYN

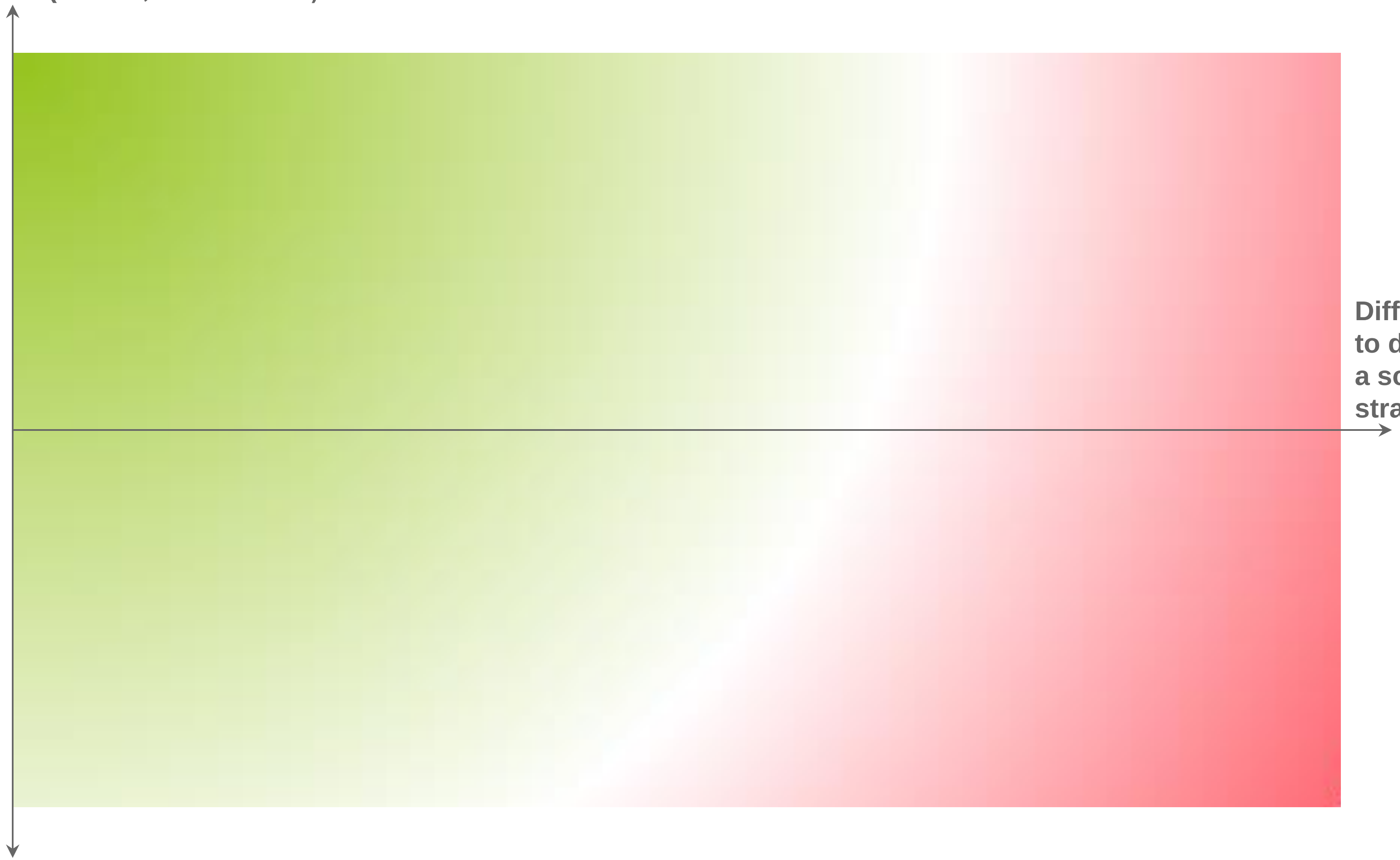
FEBRUARY, 5, 2015

www.cea.fr

OUTLINE

STRATEGIES FOR HPC DEPEND ON THE NATURE OF THE ALGORITHMS…

Access to high performance libraries (PETSc, TRILINOS…)



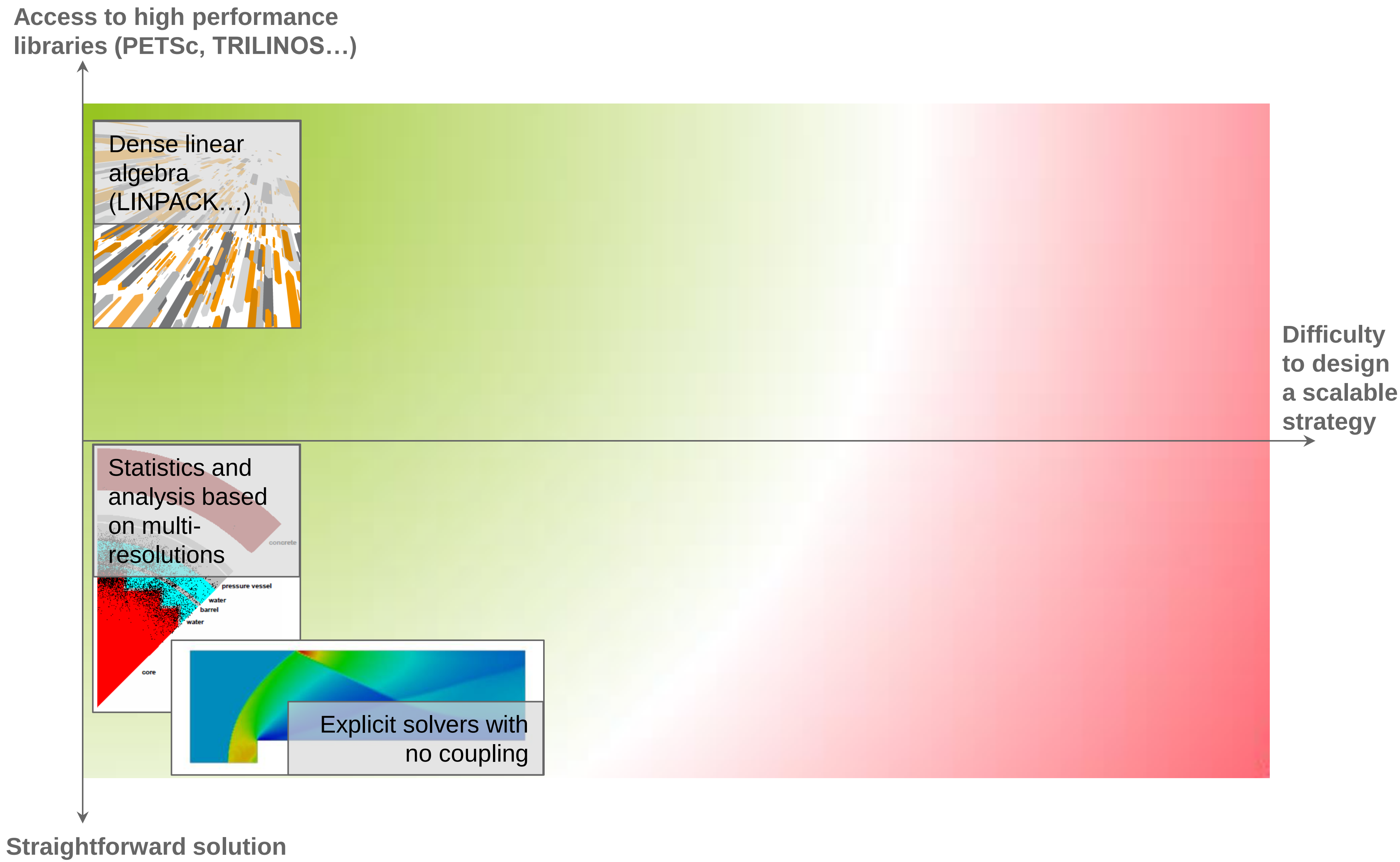
Difficulty to design a scalable strategy

Straightforward solution

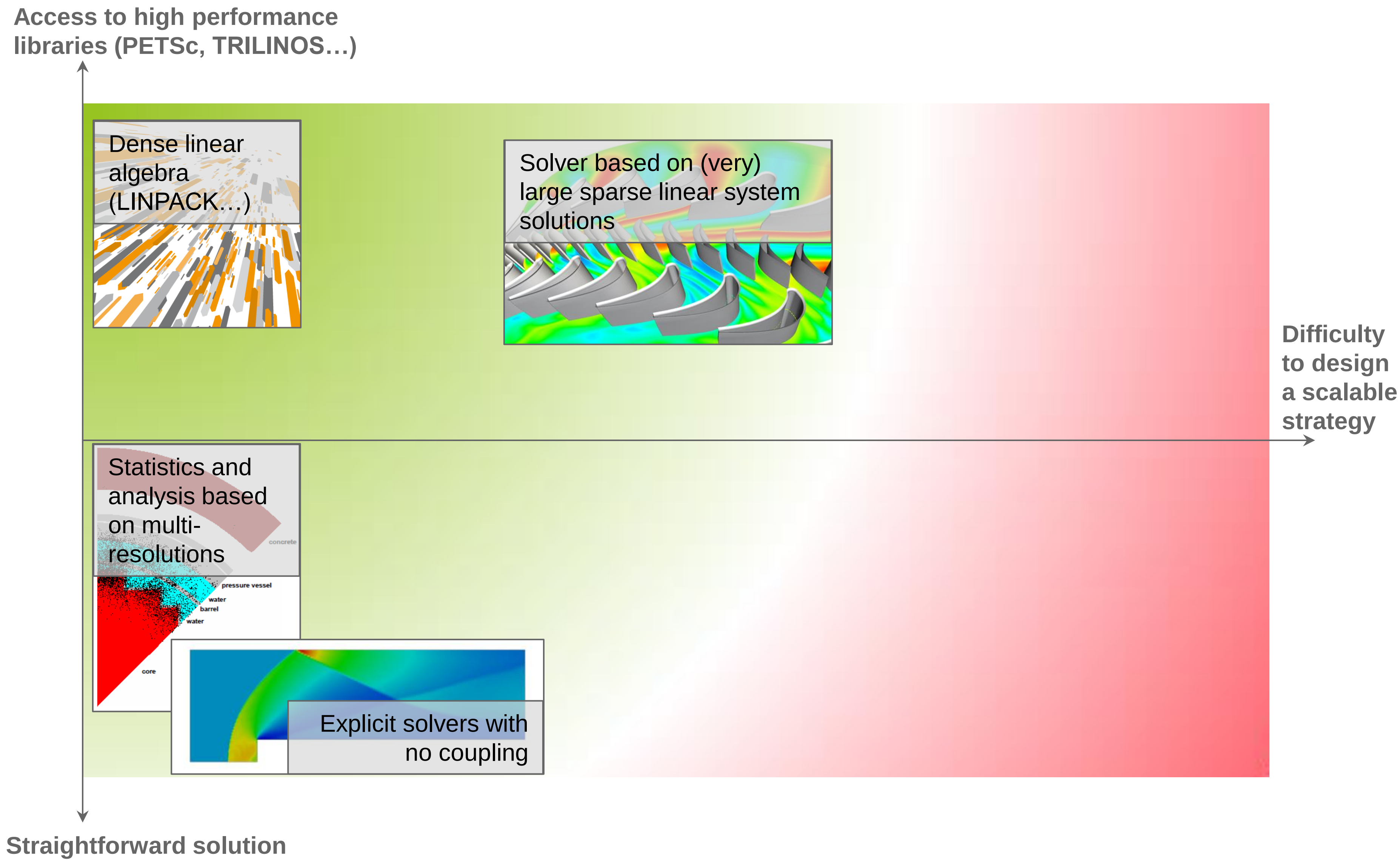
STRATEGIES FOR HPC DEPEND ON THE NATURE OF THE ALGORITHMS…

STRATEGIES FOR HPC DEPEND ON THE NATURE OF THE ALGORITHMS…

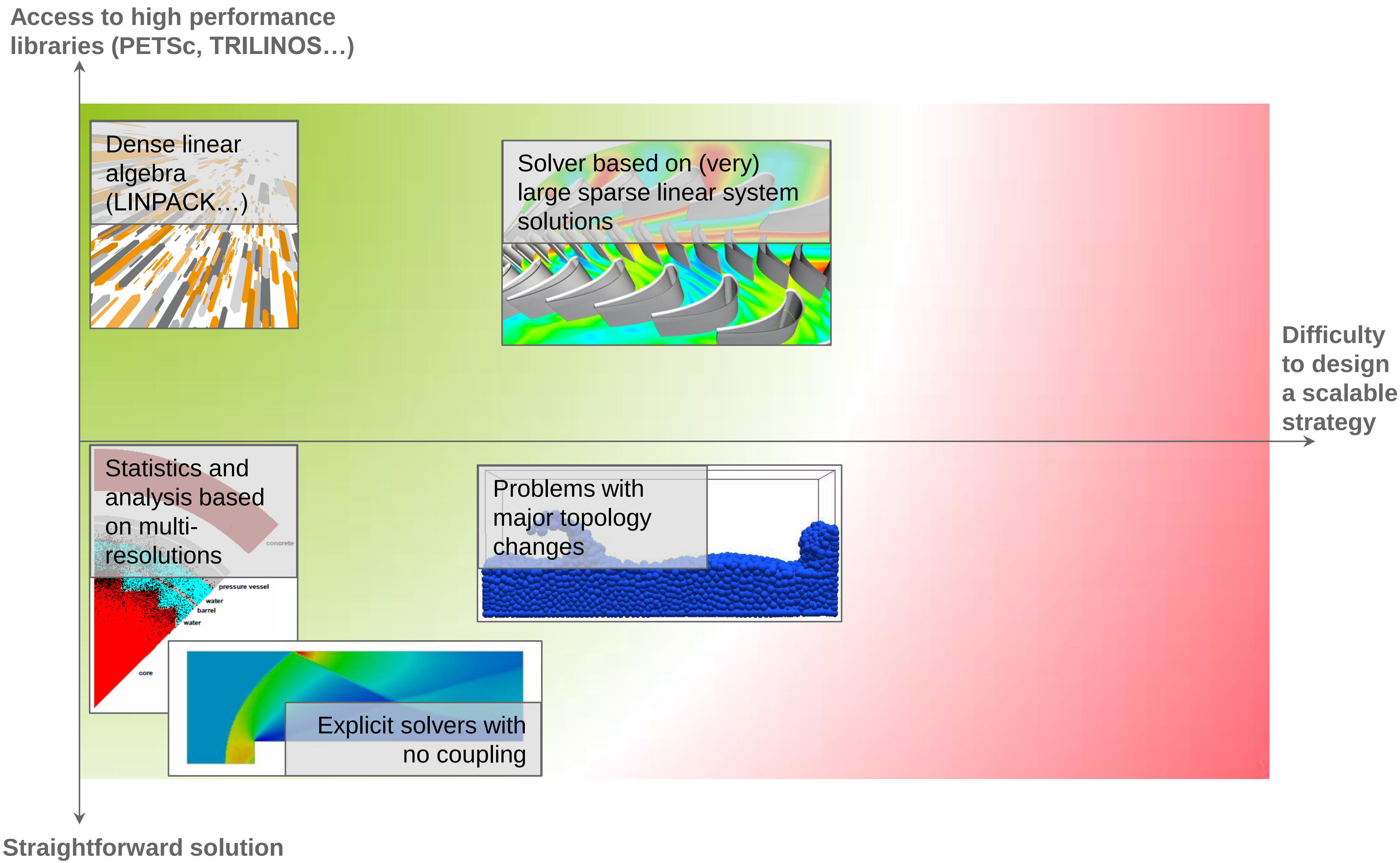
STRATEGIES FOR HPC DEPEND ON THE NATURE OF THE ALGORITHMS…



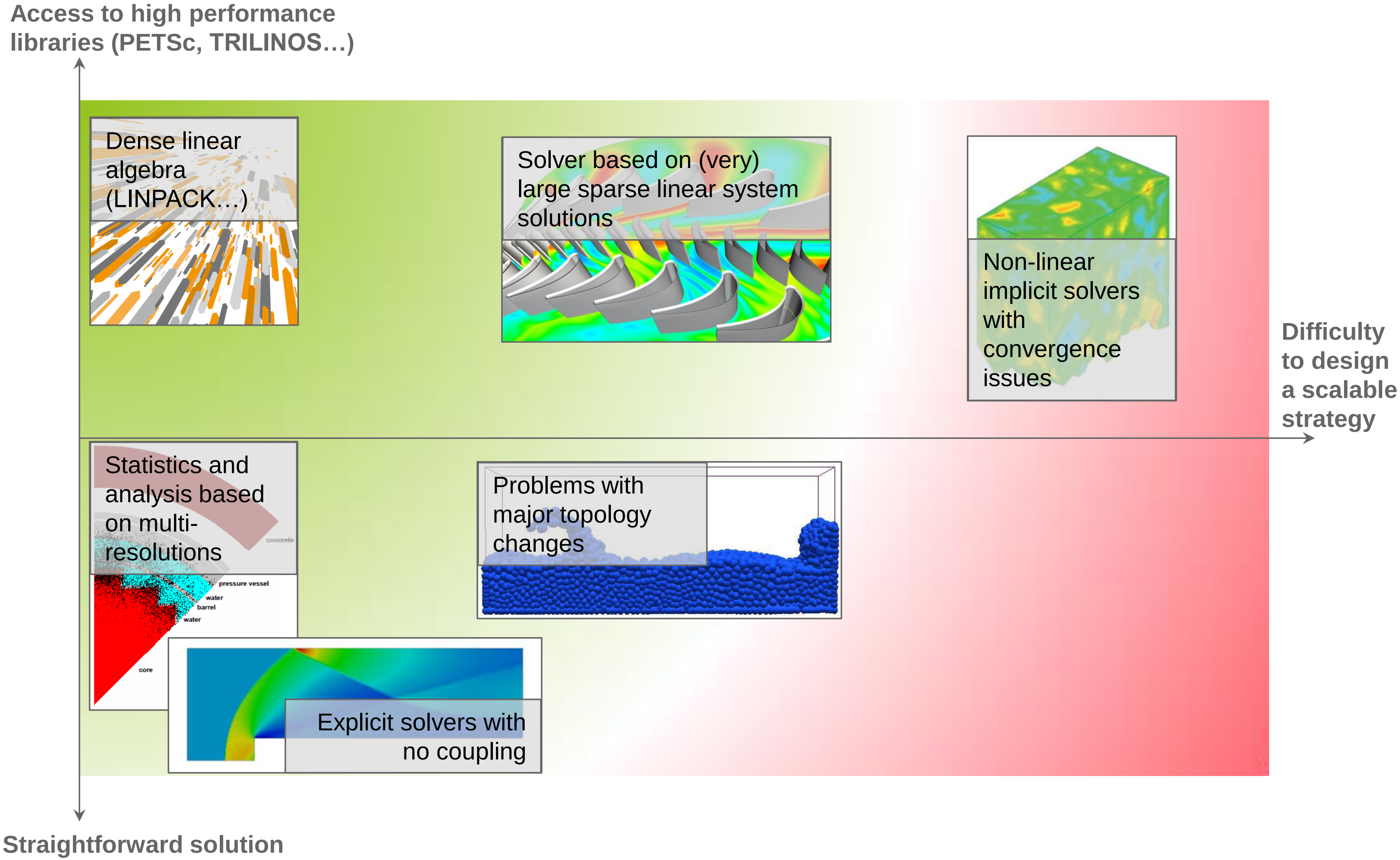
STRATEGIES FOR HPC DEPEND ON THE NATURE OF THE ALGORITHMS…



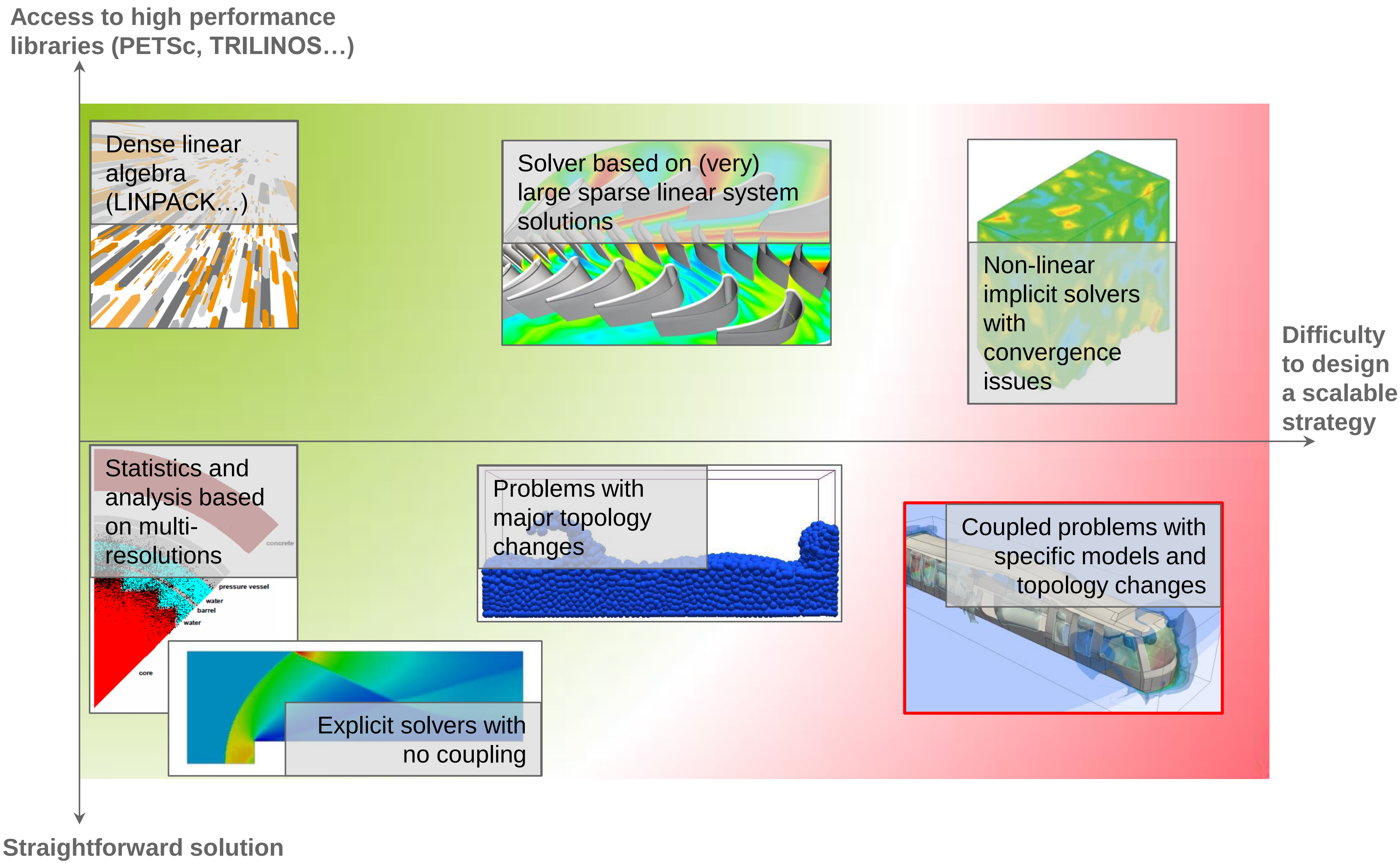
STRATEGIES FOR HPC DEPEND ON THE NATURE OF THE ALGORITHMS…



STRATEGIES FOR HPC DEPEND ON THE NATURE OF THE ALGORITHMS…



STRATEGIES FOR HPC DEPEND ON THE NATURE OF THE ALGORITHMS…



IT IS NOT ALWAYS POSSIBLE TO RESTART FROM SCRATCH…

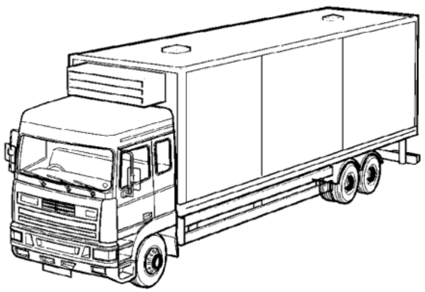
Why long life applications

- **Repositories of experience and knowledge for research teams**
  - Rise of computational methods since late 70’s
  - Continuous improvement with both new models and methods, guaranty of the sustainability and the implementation of the scientific skills of the teams
  - Integration into external industrial processes

IT IS NOT ALWAYS POSSIBLE TO RESTART FROM SCRATCH…

Why long life applications

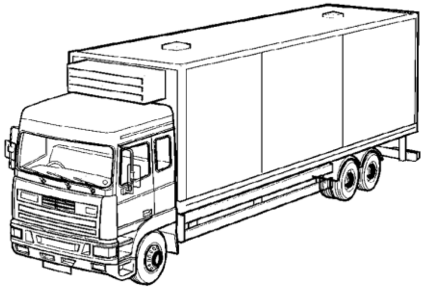
- **Repositories of experience and knowledge for research teams**
  - Rise of computational methods since late 70's
  - Continuous improvement with both new models and methods, guaranty of the sustainability and the implementation of the scientific skills of the teams
  - Integration into external industrial processes
- **Inertia**
  - Complete rewriting process rapidly tedious and out of reach (> 100 000 lines)
  - Development support based on the usage of the application
  - No initial thinking about the life time of applications (otherwise, no Year 2000 bug…)



IT IS NOT ALWAYS POSSIBLE TO RESTART FROM SCRATCH…

Why long life applications

- **Repositories of experience and knowledge for research teams**
  - Rise of computational methods since late 70’s
  - Continuous improvement with both new models and methods, guaranty of the sustainability and the implementation of the scientific skills of the teams
  - Integration into external industrial processes
- **Inertia**
  - Complete rewriting process rapidly tedious and out of reach (> 100 000 lines)
  - Development support based on the usage of the application
  - No initial thinking about the life time of applications (otherwise, no Year 2000 bug…)



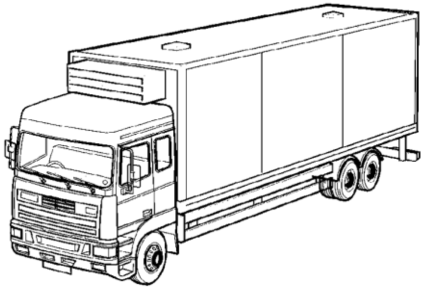
Software identity

- **Main data structure depending on the progamming standards at the beginning of the development**
  - FORTRAN = Formula Translation
  - Convergence on methods emulating dynamic allocation in the early 80’s

IT IS NOT ALWAYS POSSIBLE TO RESTART FROM SCRATCH…

Why long life applications

- **Repositories of experience and knowledge for research teams**
  - Rise of computational methods since late 70's
  - Continuous improvement with both new models and methods, guaranty of the sustainability and the implementation of the scientific skills of the teams
  - Integration into external industrial processes
- **Inertia**
  - Complete rewriting process rapidly tedious and out of reach (> 100 000 lines)
  - Development support based on the usage of the application
  - No initial thinking about the life time of applications (otherwise, no Year 2000 bug…)



Software identity

- **Main data structure depending on the progamming standards at the beginning of the development**
  - FORTRAN = Formula Translation
  - Convergence on methods emulating dynamic allocation in the early 80's
- **Hardware influence on programming techniques for scientific applications**
  - Vectorial programming for CRAY (80)
  - Share memory OpenMP (90)
  - Distributed memory MPI (fin 90, mi-2000)
  - Everything alltogether, vectorial is back (SIMD), asynchronicity…
  - Hard conversion between programming models
    - Large rewriting and debugging
    - Risk for long immobilization of the application

EPX: PARALLEL ALGORITHMS FOR FAST TRANSIENT FLUID STRUCTURE DYNAMICS

EUROPLEXUS SOFTWARE

EURDYN (1973-1988)
Fluid-structure fast transient dynamics

PLEXUS (1977-1999)
Fast transient dynamics for accidental situations in the field of civil nuclear energy

PLEXIS-3C (1985-1999)
Data structure derived from PLEXUS
Integration of functionalities from EURDYN (fluid-structure interaction in particular)

H. Bung, P. Galon, F. Bliard, D. Guilbaud, O. Jamond, A. Beccantini

M. Larcher, F. Casadei, G. Valsamos

Co-ownership by CEA/JRC
Development open to a **limited number of Major Partners** (EDF, ONERA)
Distribution by CEA since 2013 : **Production Version** under licence, **Education & Research Version** free for academics

EPX – A QUICK DESCRIPTION (1/2)

- **Local equations**

$$\rho\ddot{\mathbf{q}} + \nabla\cdot\left\{\sigma\left[\varepsilon\left(\mathbf{q}\right)\right]\right\} = \mathbf{f}_{vol}^{str}$$

$$\rho\dot{\mathbf{u}} + \nabla P + \mathbf{f}_{trans}\left(\mathbf{u}\right) = \mathbf{f}_{vol}^{flu}$$

$$\dot{\rho} + \nabla\cdot\left(\rho\mathbf{u}\right) = 0$$

$$\dot{E} + \nabla\cdot\left[\mathbf{u}\left(E+P\right)\right] = 0$$

- **Kinematics constraints**

$$\mathbf{C}\left(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}},\mathbf{u},\dot{\mathbf{u}}\right) = \mathbf{S}$$

- **Explicit time integration scheme**

$$\dot{\mathbf{q}}^{n+1/2} = \dot{\mathbf{q}}^{n} + \frac{\Delta t}{2}\ddot{\mathbf{q}}^{n}$$

$$\mathbf{q}^{n+1} = \mathbf{q}^{n} + \Delta t\,\dot{\mathbf{q}}^{n+1/2}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^{n} + \Delta t\,\dot{\mathbf{u}}^{n}$$

- **Main characteristics**

Geometrically non-linear

$$\varepsilon\left(\mathbf{q}\right) = \frac{1}{2}\left(\nabla\mathbf{q} + {}^{t}\nabla\mathbf{q} - \nabla\mathbf{q}^{t}\nabla\mathbf{q}\right)$$

Conditional stability

$$\Delta t \le \frac{2}{\omega_{max}} \Leftarrow \Delta t \le \frac{l_c}{c}$$

EPX – A QUICK DESCRIPTION (1/2)

Local equations

$$\rho\ddot{\mathbf{q}} + \nabla\cdot\left\{\sigma\left[\varepsilon(\mathbf{q})\right]\right\} = \mathbf{f}_{vol}^{str}$$

$$\rho\dot{\mathbf{u}} + \nabla P + \mathbf{f}_{trans}(\mathbf{u}) = \mathbf{f}_{vol}^{flu}$$

$$\dot{\rho} + \nabla\cdot(\rho\mathbf{u}) = 0$$

$$\dot{E} + \nabla\cdot\left[\mathbf{u}(E+P)\right] = 0$$

Kinematics constraints

$$\mathbf{C}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}},\mathbf{u},\dot{\mathbf{u}}) = \mathbf{S}$$

Explicit time integration scheme

$$\dot{\mathbf{q}}^{n+1/2} = \dot{\mathbf{q}}^{n} + \frac{\Delta t}{2}\ddot{\mathbf{q}}^{n}$$

$$\mathbf{q}^{n+1} = \mathbf{q}^{n} + \Delta t\,\dot{\mathbf{q}}^{n+1/2}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^{n} + \Delta t\,\dot{\mathbf{u}}^{n}$$

Main characteristics

Geometrically non-linear

$$\varepsilon(\mathbf{q}) = \frac{1}{2}\left(\nabla\mathbf{q} + {}^{t}\nabla\mathbf{q} - \nabla\mathbf{q}^{t}\nabla\mathbf{q}\right)$$

Conditional stability

$$\Delta t \le \frac{2}{\omega_{max}} \Leftarrow \Delta t \le \frac{l_c}{c}$$

Discrete system

$$\begin{bmatrix} \mathbf{M}_S & \\ & \mathbf{M}_F^{n+1} \end{bmatrix}\begin{bmatrix} \ddot{\mathbf{Q}}^{n+1} \\ \dot{\mathbf{U}}^{n+1} \end{bmatrix} + \mathbf{F}_{link}^{n+1} = \begin{bmatrix} \mathbf{F}_{vol}^{str} \\ \mathbf{F}_{vol}^{flu} \end{bmatrix}^{n+1} - \begin{bmatrix} \mathbf{F}_{int}\left(\mathbf{Q}^{n+1}\right) \\ \mathbf{F}_P\left(\mathbf{U}^{n+1}\right) + \mathbf{F}_{trans}\left(\mathbf{U}^{n+1}\right) \end{bmatrix}$$

$$\mathbf{C}^{n+1}\begin{bmatrix} \dot{\mathbf{Q}}^{n+1} \\ \mathbf{U}^{n+1} \end{bmatrix} = \mathbf{S}^{n+1}$$

$$[\rho]^{n+1} = [\rho]^{n} + \mathbf{F}_\rho(\mathbf{U})$$

$$[E]^{n+1} = [E]^{n} + \mathbf{F}_E(\mathbf{U})$$

EPX – A QUICK DESCRIPTION (1/2)

Local equations

$$\rho\ddot{\mathbf{q}} + \nabla\cdot\left\{\sigma\left[\varepsilon(\mathbf{q})\right]\right\} = \mathbf{f}_{vol}^{str}$$

$$\rho\dot{\mathbf{u}} + \nabla P + \mathbf{f}_{trans}(\mathbf{u}) = \mathbf{f}_{vol}^{flu}$$

$$\dot{\rho} + \nabla\cdot(\rho\mathbf{u}) = 0$$

$$\dot{E} + \nabla\cdot\left[\mathbf{u}(E+P)\right] = 0$$

Kinematics constraints

$$\mathbf{C}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}},\mathbf{u},\dot{\mathbf{u}}) = \mathbf{S}$$

Explicit time integration scheme

$$\dot{\mathbf{q}}^{n+1/2} = \dot{\mathbf{q}}^{n} + \frac{\Delta t}{2}\ddot{\mathbf{q}}^{n}$$

$$\mathbf{q}^{n+1} = \mathbf{q}^{n} + \Delta t\,\dot{\mathbf{q}}^{n+1/2}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^{n} + \Delta t\,\dot{\mathbf{u}}^{n}$$

Main characteristics

Geometrically non-linear

$$\varepsilon(\mathbf{q}) = \frac{1}{2}\left(\nabla\mathbf{q} + {}^t\nabla\mathbf{q} - \nabla\mathbf{q}\,{}^t\nabla\mathbf{q}\right)$$

Conditional stability

$$\Delta t \le \frac{2}{\omega_{max}} \Leftarrow \Delta t \le \frac{l_c}{c}$$

Discrete system

$$\begin{bmatrix} \mathbf{M}_S & \\ & \mathbf{M}_F^{n+1} \end{bmatrix}\begin{bmatrix} \ddot{\mathbf{Q}}^{n+1} \\ \dot{\mathbf{U}}^{n+1} \end{bmatrix} + \boxed{\mathbf{F}_{link}^{n+1}} = \begin{bmatrix} \mathbf{F}_{vol}^{str} \\ \mathbf{F}_{vol}^{flu} \end{bmatrix}^{n+1} - \begin{bmatrix} \mathbf{F}_{int}\left(\mathbf{Q}^{n+1}\right) \\ \mathbf{F}_P\left(\mathbf{U}^{n+1}\right) + \mathbf{F}_{trans}\left(\mathbf{U}^{n+1}\right) \end{bmatrix}$$

$$\mathbf{C}^{n+1}\begin{bmatrix} \dot{\mathbf{Q}}^{n+1} \\ \mathbf{U}^{n+1} \end{bmatrix} = \mathbf{S}^{n+1}$$

$$[\rho]^{n+1} = [\rho]^{n} + \mathbf{F}_\rho(\mathbf{U})$$

$$[E]^{n+1} = [E]^{n} + \mathbf{F}_E(\mathbf{U})$$

Link force computation

- Penalty approach:

$$\mathbf{F}_{link}^{n+1} = \mathbf{K}\left(\mathbf{C}^{n+1}\begin{bmatrix} \dot{\mathbf{Q}}^{n+1/2} \\ \mathbf{U}^{n+1} \end{bmatrix} - \mathbf{S}^{n+1}\right)$$

☑ *Explicit link forces*
☒ *Choice of penalty coefficients*
☒ *Impact on time integration stability*

- Dual approach:

$$\mathbf{F}_{link}^{n+1} = {}^t\tilde{\mathbf{C}}^{n+1}\mathbf{\Lambda}$$

$$\begin{bmatrix} \mathbf{M}_S & \\ & \mathbf{M}_F^{n+1} \end{bmatrix}\begin{bmatrix} \ddot{\mathbf{Q}}^{n+1} \\ \dot{\mathbf{U}}^{n+1} \end{bmatrix} + {}^t\tilde{\mathbf{C}}^{n+1}\mathbf{\Lambda} = \begin{bmatrix} \mathbf{F}_{vol}^{str} \\ \mathbf{F}_{vol}^{flu} \end{bmatrix}^{n+1} - \begin{bmatrix} \mathbf{F}_{int}\left(\mathbf{Q}^{n+1}\right) \\ \mathbf{F}_P\left(\mathbf{U}^{n+1}\right) + \mathbf{F}_{trans}\left(\mathbf{U}^{n+1}\right) \end{bmatrix}$$

$$\tilde{\mathbf{C}}^{n+1}\begin{bmatrix} \ddot{\mathbf{Q}}^{n+1} \\ \dot{\mathbf{U}}^{n+1} \end{bmatrix} = \tilde{\mathbf{S}}^{n+1}$$

☑ *Exact link verification*
☑ *No impact on time integration stability*
☒ *Matrix system to build and solve*

EPX – A QUICK DESCRIPTION (1/2)

Local equations

$$\rho\ddot{\mathbf{q}} + \nabla\cdot\left\{\sigma\left[\varepsilon(\mathbf{q})\right]\right\} = \mathbf{f}_{vol}^{str}$$

$$\rho\dot{\mathbf{u}} + \nabla P + \mathbf{f}_{trans}(\mathbf{u}) = \mathbf{f}_{vol}^{flu}$$

$$\dot{\rho} + \nabla\cdot(\rho\mathbf{u}) = 0$$

$$\dot{E} + \nabla\cdot\left[\mathbf{u}(E+P)\right] = 0$$

Kinematics constraints

$$\mathbf{C}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}},\mathbf{u},\dot{\mathbf{u}}) = \mathbf{S}$$

Explicit time integration scheme

$$\dot{\mathbf{q}}^{n+1/2} = \dot{\mathbf{q}}^{n} + \frac{\Delta t}{2}\ddot{\mathbf{q}}^{n}$$

$$\mathbf{q}^{n+1} = \mathbf{q}^{n} + \Delta t\,\dot{\mathbf{q}}^{n+1/2}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^{n} + \Delta t\,\dot{\mathbf{u}}^{n}$$

Main characteristics

Geometrically non-linear

$$\varepsilon(\mathbf{q}) = \frac{1}{2}\left(\nabla\mathbf{q} + {}^{t}\nabla\mathbf{q} - \nabla\mathbf{q}\,{}^{t}\nabla\mathbf{q}\right)$$

Conditional stability

$$\Delta t \le \frac{2}{\omega_{max}} \Leftarrow \Delta t \le \frac{l_c}{c}$$

Discrete system

$$\begin{bmatrix}\mathbf{M}_S & \\ & \mathbf{M}_F^{n+1}\end{bmatrix}\begin{bmatrix}\ddot{\mathbf{Q}}^{n+1}\\ \dot{\mathbf{U}}^{n+1}\end{bmatrix} + \boxed{\mathbf{F}_{link}^{n+1}} = \begin{bmatrix}\mathbf{F}_{vol}^{str}\\ \mathbf{F}_{vol}^{flu}\end{bmatrix}^{n+1} - \begin{bmatrix}\mathbf{F}_{int}\left(\mathbf{Q}^{n+1}\right)\\ \mathbf{F}_{P}\left(\mathbf{U}^{n+1}\right) + \mathbf{F}_{trans}\left(\mathbf{U}^{n+1}\right)\end{bmatrix}$$

$$\mathbf{C}^{n+1}\begin{bmatrix}\dot{\mathbf{Q}}^{n+1}\\ \mathbf{U}^{n+1}\end{bmatrix} = \mathbf{S}^{n+1}$$

$$[\rho]^{n+1} = [\rho]^{n} + \mathbf{F}_{\rho}(\mathbf{U})$$

$$[E]^{n+1} = [E]^{n} + \mathbf{F}_{E}(\mathbf{U})$$

Link force computation

- Penalty approach: $\mathbf{F}_{link}^{n+1} = \mathbf{K}\left(\mathbf{C}^{n+1}\begin{bmatrix}\dot{\mathbf{Q}}^{n+1/2}\\ \mathbf{U}^{n+1}\end{bmatrix} - \mathbf{S}^{n+1}\right)$

  ☑ *Explicit link forces*

  ☒ *Choice of penalty coefficients*

  ☒ *Impact on time integration stability*

- Dual approach: $\mathbf{F}_{link}^{n+1} = {}^{t}\tilde{\mathbf{C}}^{n+1}\boldsymbol{\Lambda}$

$$\begin{bmatrix}\mathbf{M}_S & \\ & \mathbf{M}_F^{n+1}\end{bmatrix}\begin{bmatrix}\ddot{\mathbf{Q}}^{n+1}\\ \dot{\mathbf{U}}^{n+1}\end{bmatrix} + {}^{t}\tilde{\mathbf{C}}^{n+1}\boldsymbol{\Lambda} = \begin{bmatrix}\mathbf{F}_{vol}^{str}\\ \mathbf{F}_{vol}^{flu}\end{bmatrix}^{n+1} - \begin{bmatrix}\mathbf{F}_{int}\left(\mathbf{Q}^{n+1}\right)\\ \mathbf{F}_{P}\left(\mathbf{U}^{n+1}\right) + \mathbf{F}_{trans}\left(\mathbf{U}^{n+1}\right)\end{bmatrix}$$

$$\tilde{\mathbf{C}}^{n+1}\begin{bmatrix}\ddot{\mathbf{Q}}^{n+1}\\ \dot{\mathbf{U}}^{n+1}\end{bmatrix} = \tilde{\mathbf{S}}^{n+1}$$

  ☑ *Exact link verification*

  ☑ *No impact on time integration stability*

  ☒ *Matrix system to build and solve*

EPX – A QUICK DESCRIPTION (2/2)

- **Discrete system**

$$\begin{bmatrix} \mathbf{M}_S & \\ & \mathbf{M}_F^{n+1} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{Q}}^{n+1} \\ \dot{\mathbf{U}}^{n+1} \end{bmatrix} + {}^t\tilde{\mathbf{C}}^{n+1}\mathbf{\Lambda} = \underbrace{\begin{bmatrix} \mathbf{F}_{vol}^{str} \\ \mathbf{F}_{vol}^{flu} \end{bmatrix}^{n+1} - \begin{bmatrix} \mathbf{F}_{int}\left(\mathbf{Q}^{n+1}\right) \\ \mathbf{F}_P\left(\mathbf{U}^{n+1}\right) + \mathbf{F}_{trans}\left(\mathbf{U}^{n+1}\right) \end{bmatrix}}_{\mathbf{F}^{n+1}}$$

$$\tilde{\mathbf{C}}^{n+1} \begin{bmatrix} \ddot{\mathbf{Q}}^{n+1} \\ \dot{\mathbf{U}}^{n+1} \end{bmatrix} = \tilde{\mathbf{S}}^{n+1}$$

$$[\rho]^{n+1} = [\rho]^n + \mathbf{F}_\rho(\mathbf{U})$$

$$[E]^{n+1} = [E]^n + \mathbf{F}_E(\mathbf{U})$$

EPX – A QUICK DESCRIPTION (2/2)

Discrete system

$$\begin{bmatrix} \mathbf{M}_S & \\ & \mathbf{M}_F^{n+1} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{Q}}^{n+1} \\ \dot{\mathbf{U}}^{n+1} \end{bmatrix} + {}^t\tilde{\mathbf{C}}^{n+1}\mathbf{\Lambda} = \underbrace{\begin{bmatrix} \mathbf{F}_{vol}^{str} \\ \mathbf{F}_{vol}^{flu} \end{bmatrix}^{n+1} - \boxed{\begin{bmatrix} \mathbf{F}_{int}\left(\mathbf{Q}^{n+1}\right) \\ \mathbf{F}_P\left(\mathbf{U}^{n+1}\right) + \mathbf{F}_{trans}\left(\mathbf{U}^{n+1}\right) \end{bmatrix}}}_{\mathbf{F}^{n+1}}$$

$$\tilde{\mathbf{C}}^{n+1} \begin{bmatrix} \ddot{\mathbf{Q}}^{n+1} \\ \dot{\mathbf{U}}^{n+1} \end{bmatrix} = \tilde{\mathbf{S}}^{n+1}$$

$$[\rho]^{n+1} = [\rho]^n + \boxed{\mathbf{F}_\rho(\mathbf{U})}$$

$$[E]^{n+1} = [E]^n + \boxed{\mathbf{F}_E(\mathbf{U})}$$

Main task 1 = Elementary loop

- *Stress computation* (constitutive laws, equations of state)
- *Fluxes computation*
- *Heterogeneous individual costs*

EPX – A QUICK DESCRIPTION (2/2)

Discrete system

$$\begin{bmatrix} \mathbf{M}_S & \\ & \mathbf{M}_F^{n+1}\end{bmatrix}\begin{bmatrix}\ddot{\mathbf{Q}}^{n+1}\\ \dot{\mathbf{U}}^{n+1}\end{bmatrix} + {}^t\tilde{\mathbf{C}}^{n+1}\boldsymbol{\Lambda} = \underbrace{\begin{bmatrix}\mathbf{F}^{str}_{vol}\\ \mathbf{F}^{flu}_{vol}\end{bmatrix}^{n+1} - \begin{bmatrix}\mathbf{F}_{int}\left(\mathbf{Q}^{n+1}\right)\\ \mathbf{F}_P\left(\mathbf{U}^{n+1}\right)+\mathbf{F}_{trans}\left(\mathbf{U}^{n+1}\right)\end{bmatrix}}_{\mathbf{F}^{n+1}}$$

$$\tilde{\mathbf{C}}^{n+1}\begin{bmatrix}\ddot{\mathbf{Q}}^{n+1}\\ \dot{\mathbf{U}}^{n+1}\end{bmatrix} = \tilde{\mathbf{S}}^{n+1}$$

$$[\rho]^{n+1} = [\rho]^{n} + \mathbf{F}_\rho\left(\mathbf{U}\right)$$

$$[E]^{n+1} = [E]^{n} + \mathbf{F}_E\left(\mathbf{U}\right)$$

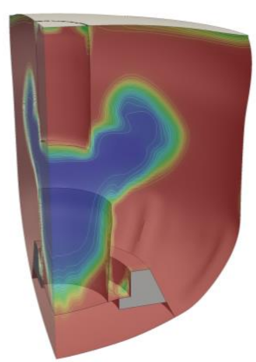
Main task 1 = Elementary loop

- *Stress computation* (constitutive laws, equations of state)
- *Fluxes computation*
- *Heterogeneous individual costs*

Main task 2 = Writing kinematic constraints

- *Candidates selection*
- *Writing the kinemtic relations*
- *Spatial sorts, inclusion and intersection computations*

Crash with self-contact



Explosion in vessels with immersed structures

EPX – A QUICK DESCRIPTION (2/2)

Discrete system

$$\begin{bmatrix} \mathbf{M}_S & \\ & \mathbf{M}_F^{n+1} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{Q}}^{n+1} \\ \dot{\mathbf{U}}^{n+1} \end{bmatrix} + {}^t\tilde{\mathbf{C}}^{n+1} \boxed{\mathbf{\Lambda}} = \underbrace{\begin{bmatrix} \mathbf{F}_{vol}^{str} \\ \mathbf{F}_{vol}^{flu} \end{bmatrix}^{n+1} - \boxed{\begin{bmatrix} \mathbf{F}_{int}\left(\mathbf{Q}^{n+1}\right) \\ \mathbf{F}_P\left(\mathbf{U}^{n+1}\right) + \mathbf{F}_{trans}\left(\mathbf{U}^{n+1}\right) \end{bmatrix}}}_{\mathbf{F}^{n+1}}$$

$$\boxed{\tilde{\mathbf{C}}^{n+1}} \begin{bmatrix} \ddot{\mathbf{Q}}^{n+1} \\ \dot{\mathbf{U}}^{n+1} \end{bmatrix} = \tilde{\mathbf{S}}^{n+1}$$

$$[\rho]^{n+1} = [\rho]^n + \boxed{\mathbf{F}_\rho\left(\mathbf{U}\right)}$$

$$[E]^{n+1} = [E]^n + \boxed{\mathbf{F}_E\left(\mathbf{U}\right)}$$

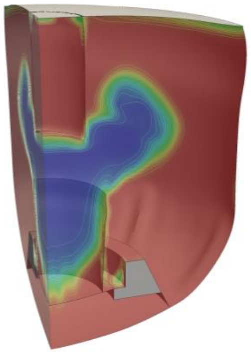
Main task 1 = Elementary loop

- *Stress computation* (constitutive laws, equations of state)
- *Fluxes computation*
- *Heterogeneous individual costs*

Main task 2 = Writing kinematic constraints

- *Candidates selection*
- *Writing the kinemtic relations*
- *Spatial sorts, inclusion and intersection computations*

Crash with self-contact



Explosion in vessels with immersed structures

Main task 3 = Link force computation

- *Condensation onto the Lagrange Multipliers*

$$\tilde{\mathbf{C}}^{n+1} \begin{bmatrix} \mathbf{M}_S & \\ & \mathbf{M}_F^{n+1} \end{bmatrix}^{-1} {}^t\tilde{\mathbf{C}}^{n+1} \mathbf{\Lambda} = \tilde{\mathbf{C}}^{n+1} \begin{bmatrix} \mathbf{M}_S & \\ & \mathbf{M}_F^{n+1} \end{bmatrix}^{-1} \mathbf{F}^{n+1} - \tilde{\mathbf{S}}^{n+1}$$

$$\Leftrightarrow \mathbf{H}^{n+1} \mathbf{\Lambda} = \mathbf{B}^{n+1}$$

- *Linear solution and scatter of the forces*

EPX – A QUICK DESCRIPTION (2/2)

Discrete system

$$\begin{bmatrix} M_S & \\ & M_F^{n+1}\end{bmatrix}\begin{bmatrix}\ddot{Q}^{n+1}\\ \dot{U}^{n+1}\end{bmatrix} + {}^t\tilde{C}^{n+1}\boxed{\Lambda} = \underbrace{\begin{bmatrix}F_{vol}^{str}\\ F_{vol}^{flu}\end{bmatrix}^{n+1} - \boxed{\begin{bmatrix}F_{int}\left(Q^{n+1}\right)\\ F_\rho\left(U^{n+1}\right)+F_{trans}\left(U^{n+1}\right)\end{bmatrix}}}_{F^{n+1}}$$

$$\boxed{\tilde{C}^{n+1}}\begin{bmatrix}\ddot{Q}^{n+1}\\ \dot{U}^{n+1}\end{bmatrix} = \tilde{S}^{n+1}$$

$$[\rho]^{n+1} = [\rho]^{n} + \boxed{F_\rho(U)}$$

$$[E]^{n+1} = [E]^{n} + \boxed{F_E(U)}$$

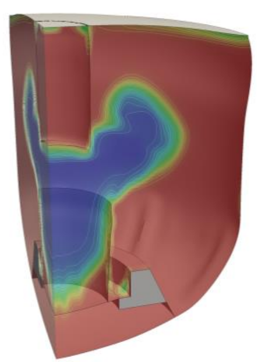
Main task 1 = Elementary loop

- *Stress computation* (constitutive laws, equations of state)
- *Fluxes computation*
- *Heterogeneous individual costs*

Main task 2 = Writing kinematic constraints

- *Candidates selection*
- *Writing the kinemtic relations*
- *Spatial sorts, inclusion and intersection computations*

Crash with self-contact



Explosion in vessels with immersed structures

Main task 3 = Link force computation

- *Condensation onto the Lagrange Multipliers*

$$\tilde{C}^{n+1}\begin{bmatrix} M_S & \\ & M_F^{n+1}\end{bmatrix}^{-1} {}^t\tilde{C}^{n+1}\Lambda = \tilde{C}^{n+1}\begin{bmatrix} M_S & \\ & M_F^{n+1}\end{bmatrix}^{-1} F^{n+1} - \tilde{S}^{n+1}$$

$$\Leftrightarrow \boxed{H^{n+1}\Lambda = B^{n+1}}$$

- *Linear solution and scatter of the forces*

EPX – A QUICK DESCRIPTION (2/2)

Discrete system

$$\begin{bmatrix} M_S & \\ & M_F^{n+1} \end{bmatrix} \begin{bmatrix} \ddot{Q}^{n+1} \\ \dot{U}^{n+1} \end{bmatrix} + {}^t\tilde{C}^{n+1}\Lambda = \underbrace{\begin{bmatrix} F_{vol}^{str} \\ F_{vol}^{flu} \end{bmatrix}^{n+1} - \begin{bmatrix} F_{int}\left(Q^{n+1}\right) \\ F_\rho\left(U^{n+1}\right) + F_{trans}\left(U^{n+1}\right) \end{bmatrix}}_{F^{n+1}}$$

$$\tilde{C}^{n+1} \begin{bmatrix} \ddot{Q}^{n+1} \\ \dot{U}^{n+1} \end{bmatrix} = \tilde{S}^{n+1}$$

$$[\rho]^{n+1} = [\rho]^n + F_\rho(U)$$

$$[E]^{n+1} = [E]^n + F_E(U)$$

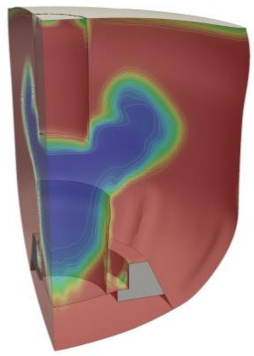
Main task 1 = Elementary loop

- *Stress computation* (constitutive laws, equations of state)
- *Fluxes computation*
- *Heterogeneous individual costs*

Main task 2 = Writing kinematic constraints

- *Candidates selection*
- *Writing the kinemtic relations*
- *Spatial sorts, inclusion and intersection computations*

Crash with self-contact



Explosion in vessels with immersed structures

Main task 3 = Link force computation

- *Condensation onto the Lagrange Multipliers*

$$\tilde{C}^{n+1}\begin{bmatrix} M_S & \\ & M_F^{n+1} \end{bmatrix}^{-1} {}^t\tilde{C}^{n+1}\Lambda = \tilde{C}^{n+1}\begin{bmatrix} M_S & \\ & M_F^{n+1} \end{bmatrix}^{-1} F^{n+1} - \tilde{S}^{n+1}$$

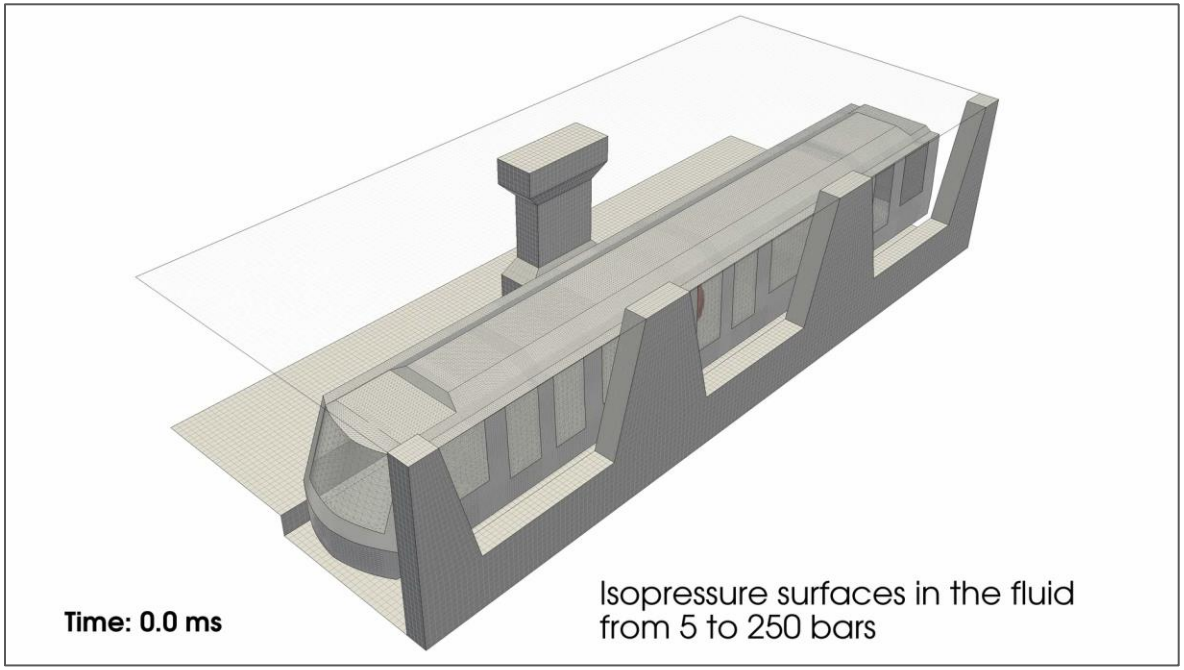
$$\Leftrightarrow \boxed{H^{n+1}\Lambda = B^{n+1}}$$

- *Linear solution and scatter of the forces*

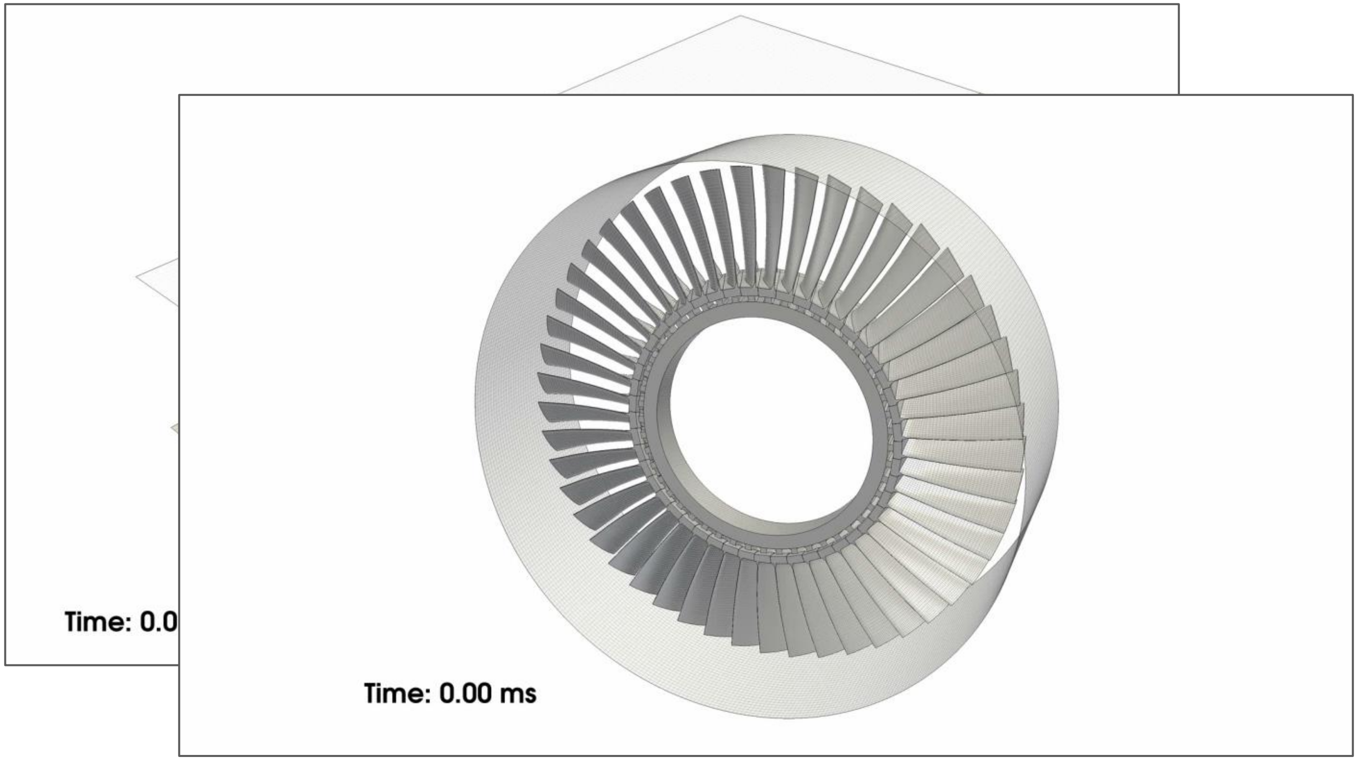
Principal characteristics

- Many specific algorithms (formulations and links)
- Various relative weights of the main tasks
- Very specific parallel strategy

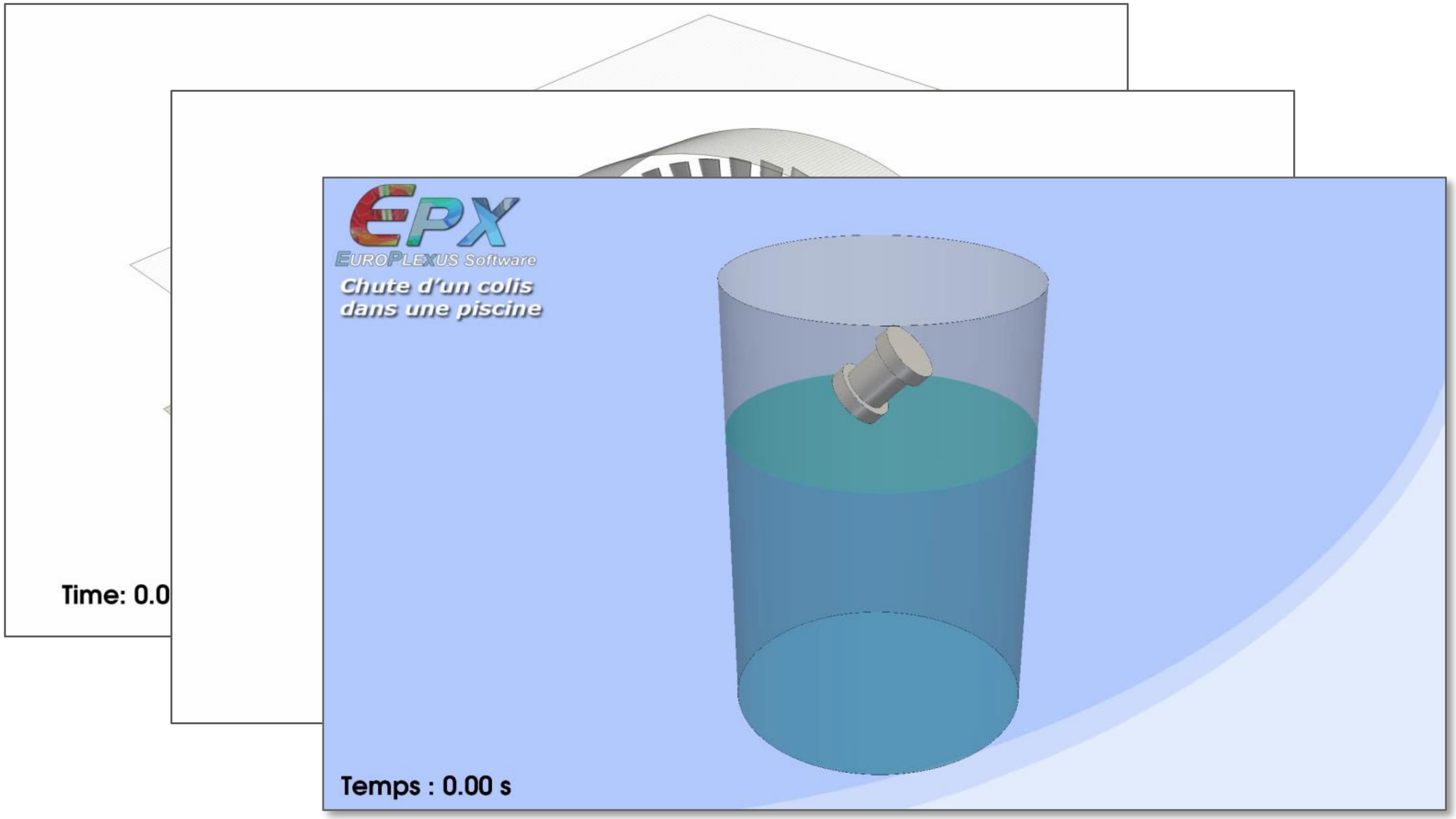
EPX: SOME ILLUSTRATIONS



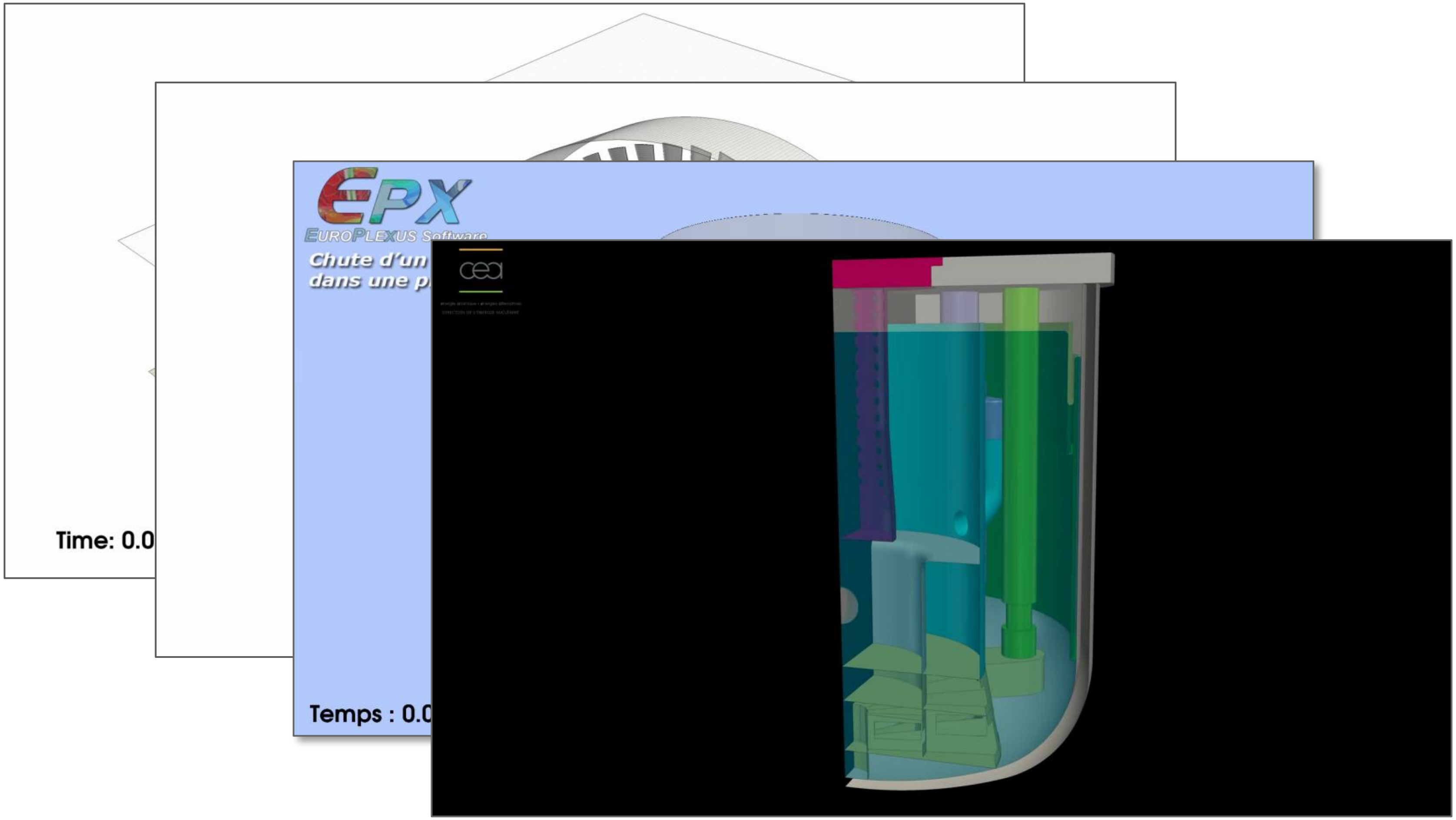
EPX: SOME ILLUSTRATIONS



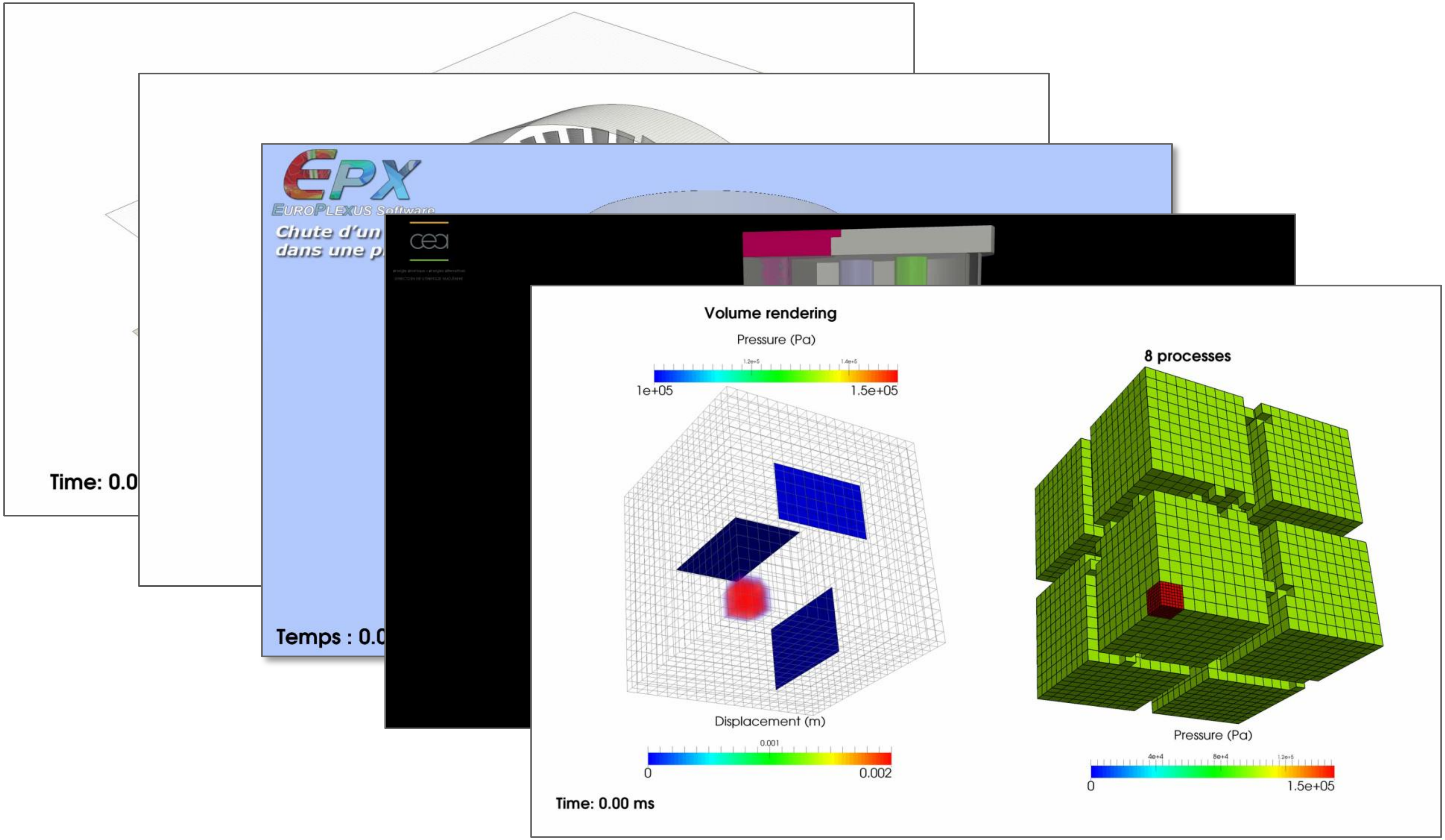
EPX: SOME ILLUSTRATIONS



EPX: SOME ILLUSTRATIONS



EPX: SOME ILLUSTRATIONS



EPX: GLOBAL PARALLEL STRATEGY

- **Distributed memory on cluster nodes**
  - Domain Decomposition through *Recursive Orthogonal Bisection*
  - Generic approach for any kind of kinematic constraints
    - Inter-domain detection of connected entities
    - Specific solver preserving scalability for the computation of link forces
  - Dynamic Domain Decomposition
    - Mandatory to handle large topological changes in the system

Initial situation on a given subdomain

Same subdomain after half a turn of the rotor

EPX: GLOBAL PARALLEL STRATEGY

Distributed memory on cluster nodes

- Domain Decomposition through *Recursive Orthogonal Bisection*
- Generic approach for any kind of kinematic constraints
  - Inter-domain detection of connected entities
  - Specific solver preserving scalability for the computation of link forces
- Dynamic Domain Decomposition
  - Mandatory to handle large topological changes in the system

Initial situation on a given subdomain

Same subdomain after half a turn of the rotor

(Advanced) shared memory inside subdomains

- Use and development of the KAAPI library (INRIA, http://kaapi.gforge.inria.fr)
- Dynamic scheduling through work stealing
- *Data Flow Graph* programming
- Adaptive task creation
- Handling affinities and heterogeneous architectures (GPU(s), MIC(s))

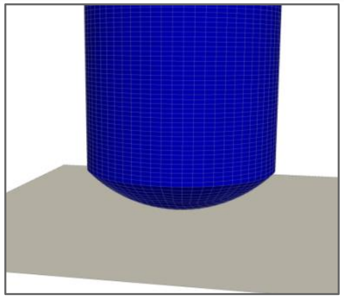
EPX: GLOBAL PARALLEL STRATEGY

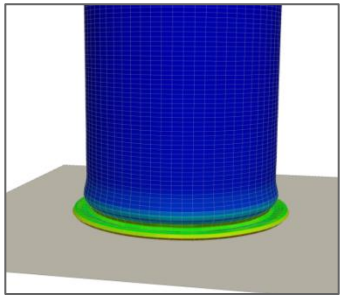
- **Distributed memory on cluster nodes**
  - Domain Decomposition through *Recursive Orthogonal Bisection*
  - Generic approach for any kind of kinematic constraints
    - Inter-domain detection of connected entities
    - Specific solver preserving scalability for the computation of link forces
  - Dynamic Domain Decomposition
    - Mandatory to handle large topological changes in the system

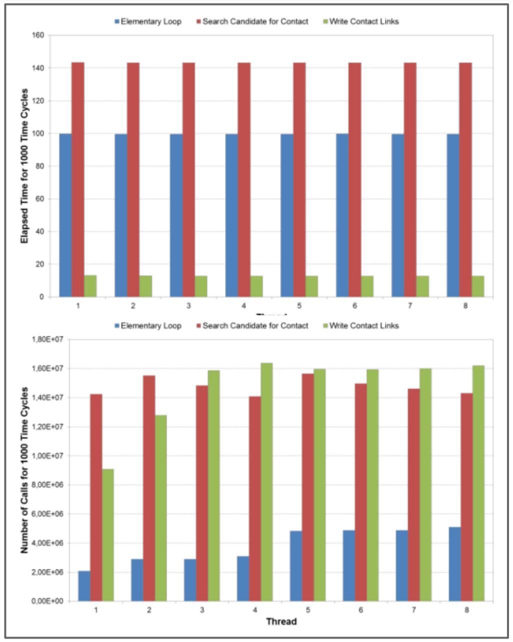
Initial situation on a given subdomain

Same subdomain after half a turn of the rotor

- **(Advanced) shared memory inside subdomains**
  - Use and development of the KAAPI library (INRIA, http://kaapi.gforge.inria.fr)
  - Dynamic scheduling through work stealing
  - *Data Flow Graph* programming
  - Adaptive task creation
  - Handling affinities and heterogeneous architectures (GPU(s), MIC(s))







EPX: GLOBAL PARALLEL STRATEGY

- **Distributed memory on cluster nodes**
  - Domain Decomposition through *Recursive Orthogonal Bisection*
  - Generic approach for any kind of kinematic constraints
    - Inter-domain detection of connected entities
    - Specific solver preserving scalability for the computation of link forces
  - Dynamic Domain Decomposition
    - Mandatory to handle large topological changes in the system

Initial situation on a given subdomain

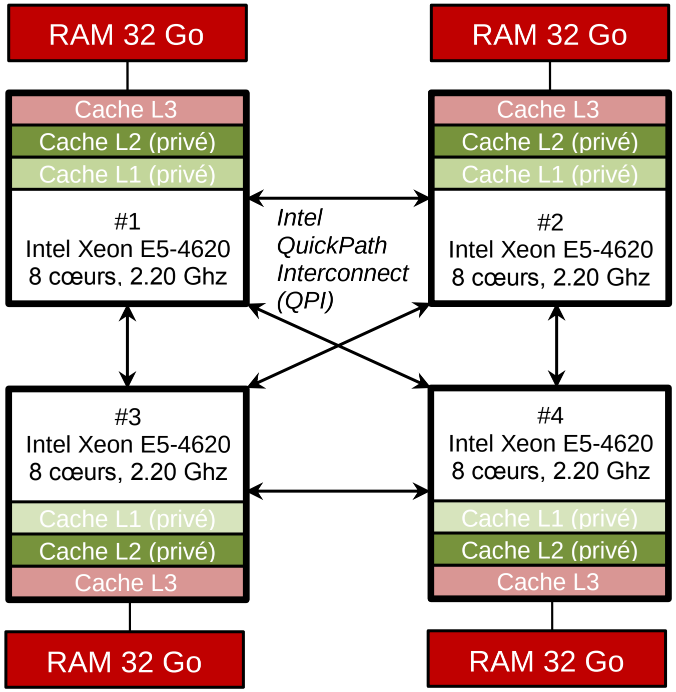
Same subdomain after half a turn of the rotor

- **(Advanced) shared memory inside subdomains**
  - Use and development of the KAAPI library (INRIA, http://kaapi.gforge.inria.fr)
  - Dynamic scheduling through work stealing
  - *Data Flow Graph* programming
  - Adaptive task creation
  - Handling affinities and heterogeneous architectures (GPU(s), MIC(s))

- **Tests on PRACE/Curie supercomputer**
  - Preparatory Access in 2012
  - Scalability achieved for >= 1 024 cores

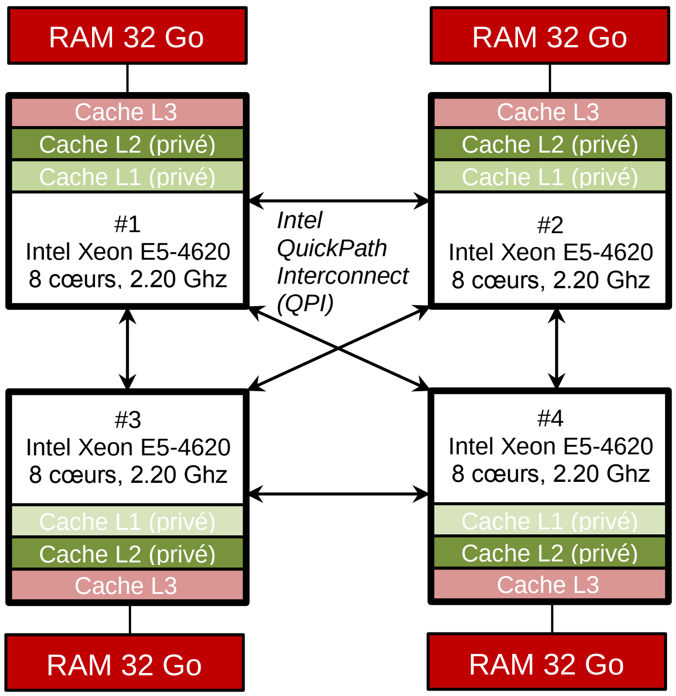
EPX: SHARED MEMORY OPTIMIZATION FOR MULTI-PROCESSOR NODES (1/2)

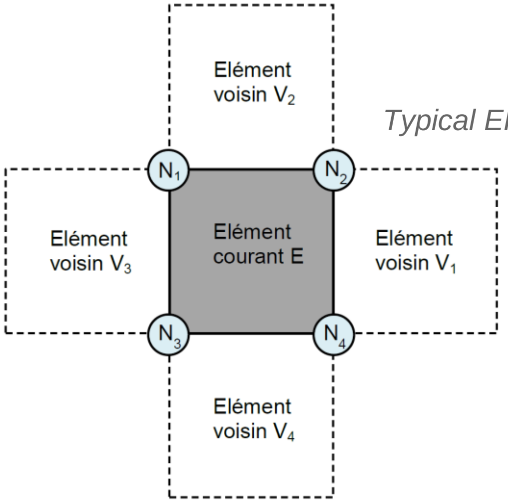
- **Target architecture**



EPX: SHARED MEMORY OPTIMIZATION FOR MULTI-PROCESSOR NODES (1/2)

Target architecture

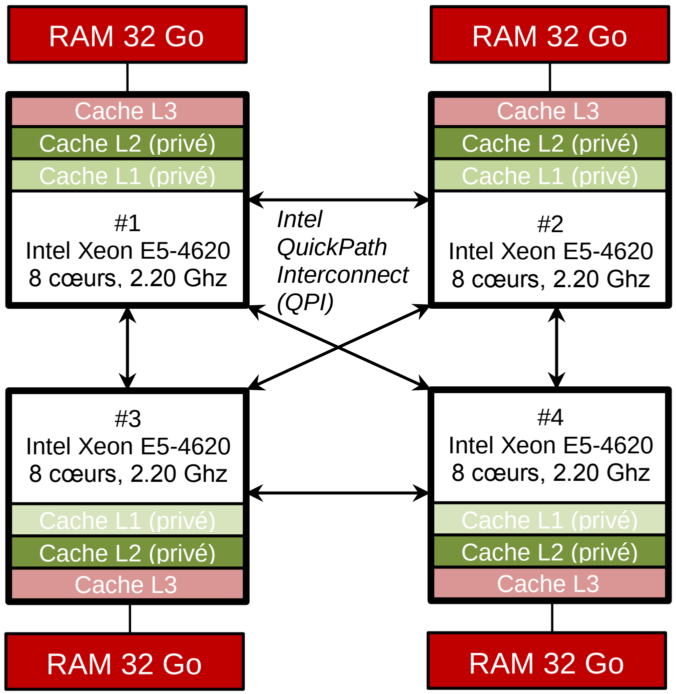




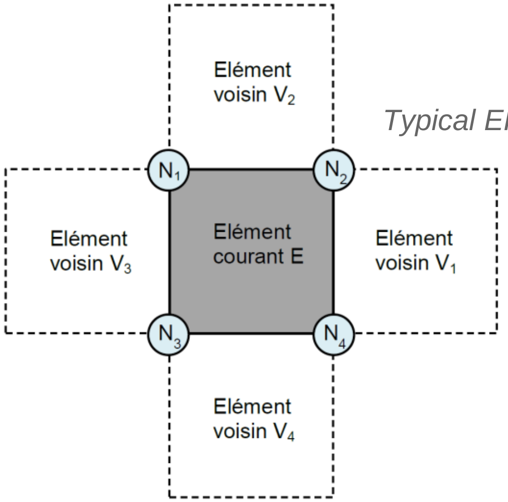
Typical EPX cell

EPX: SHARED MEMORY OPTIMIZATION FOR MULTI-PROCESSOR NODES (1/2)

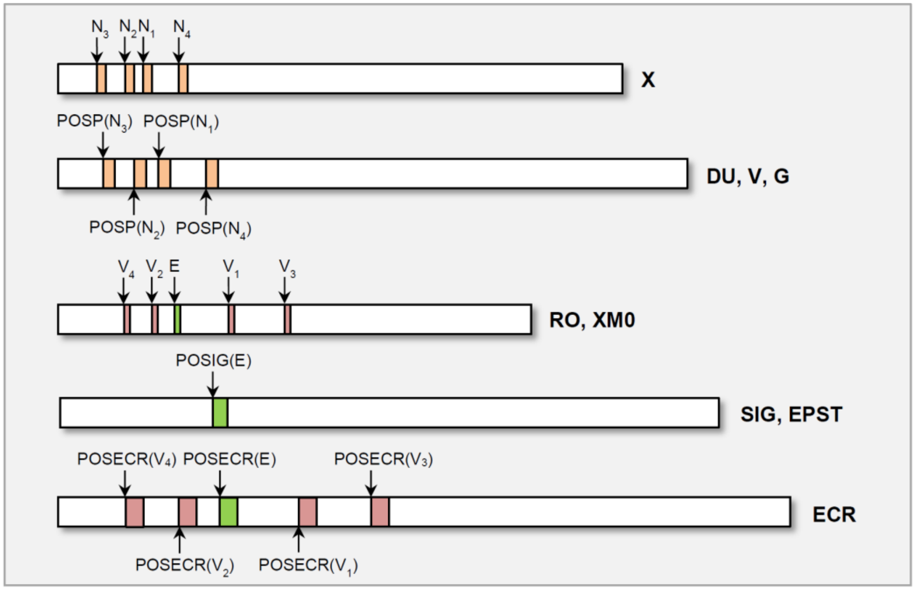
- Target architecture



Typical EPX cell

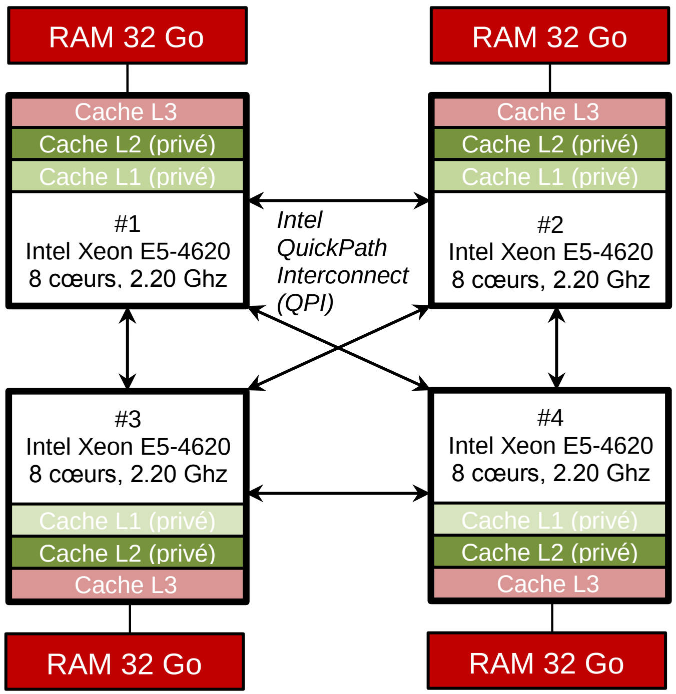


Data placement in memory



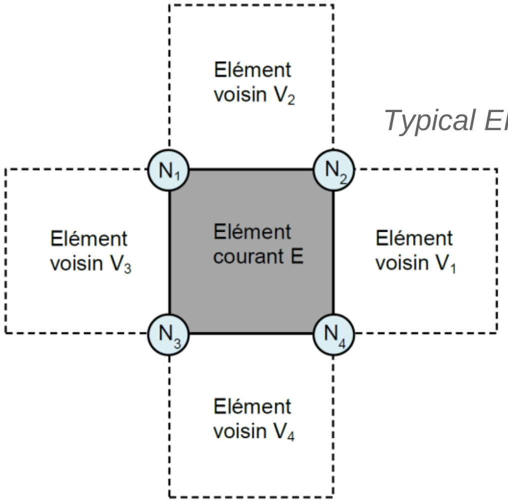
EPX: SHARED MEMORY OPTIMIZATION FOR MULTI-PROCESSOR NODES (1/2)

Target architecture



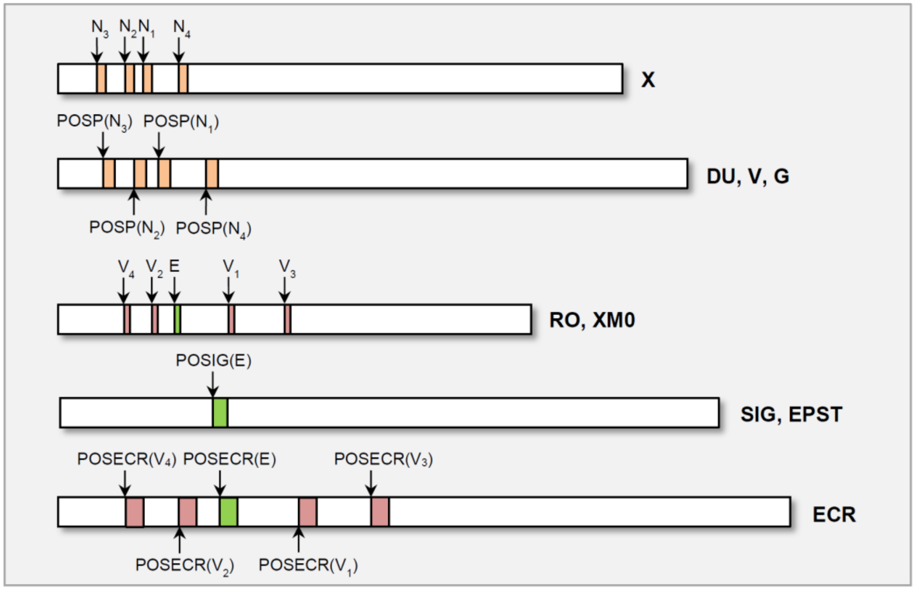
Problematics

- Global data structure inherited from past choices
- Lack of scalability for the elementary loop using threads on several sockets



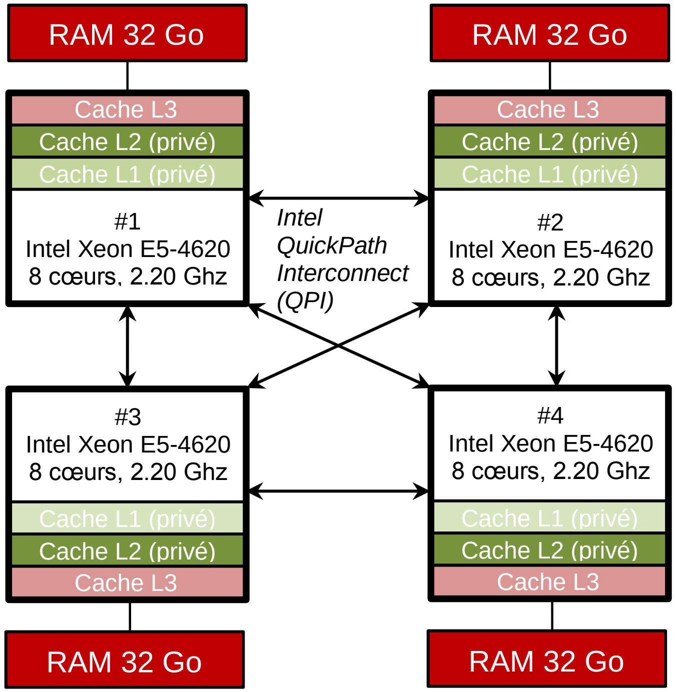
Typical EPX cell

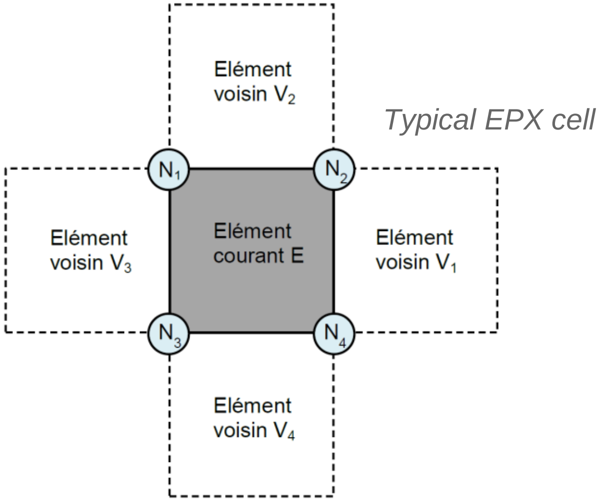
Data placement in memory



EPX: SHARED MEMORY OPTIMIZATION FOR MULTI-PROCESSOR NODES (1/2)

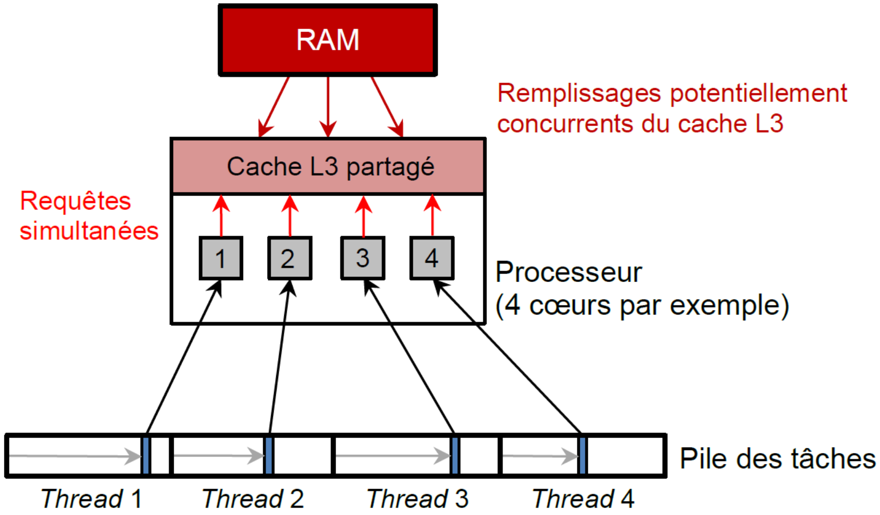
Target architecture





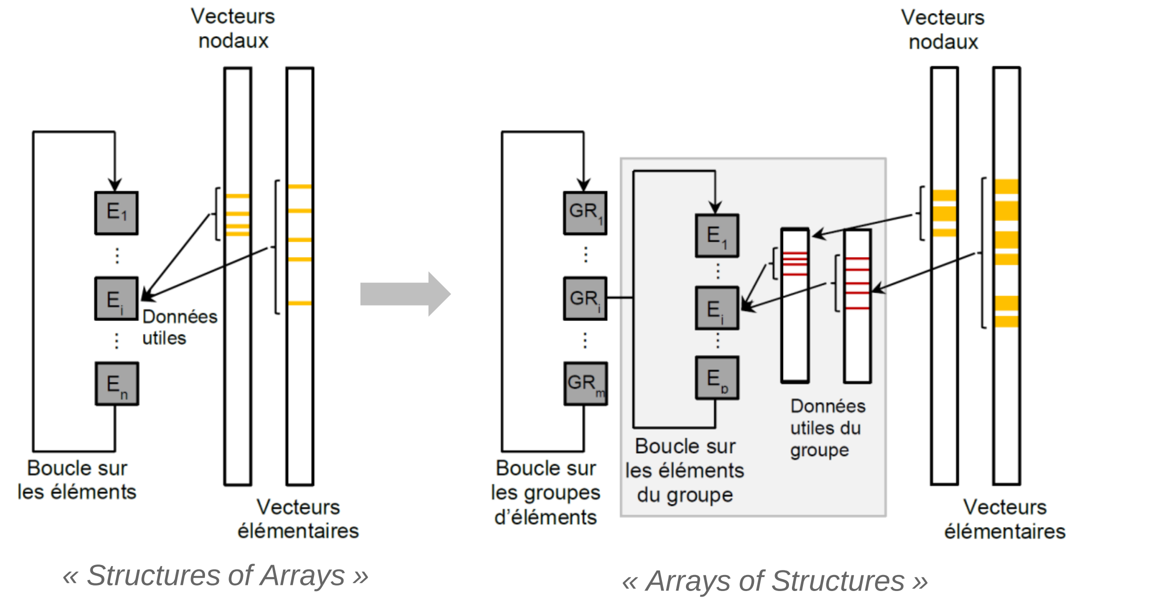
Problematics

- Global data structure inherited from past choices
- Lack of scalability for the elementary loop using threads on several sockets



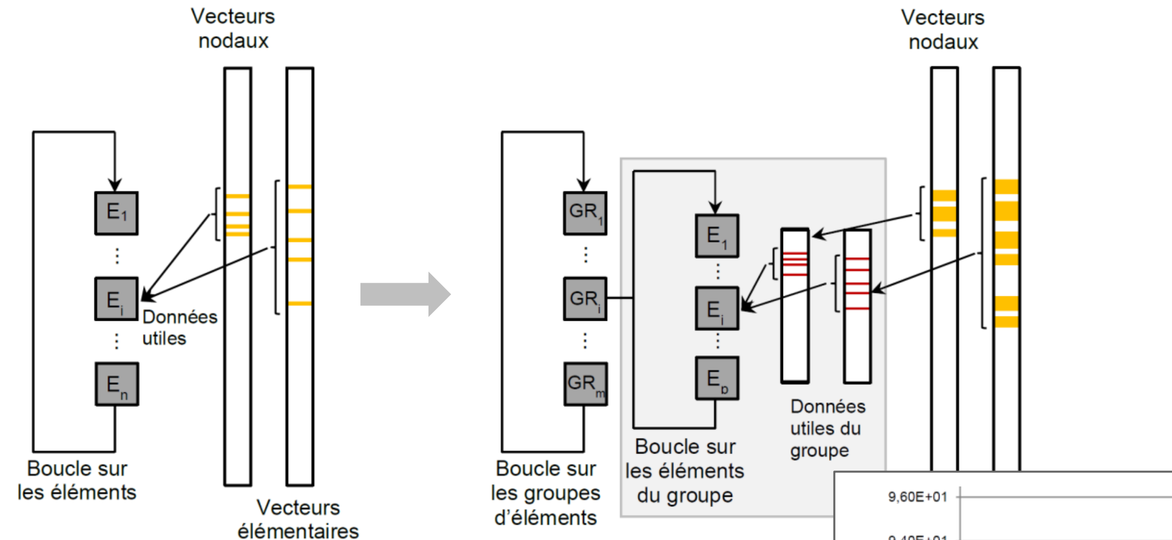
EPX: SHARED MEMORY OPTIMIZATION FOR MULTI-PROCESSOR NODES (2/2)

- On-going optimization (PhD M. Sridi, CEA, INRIA, 2013-2016)



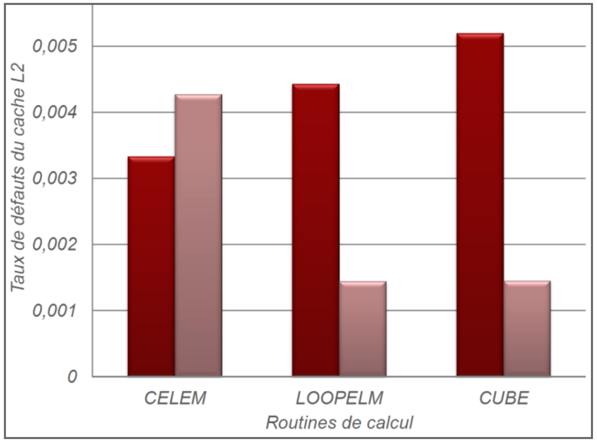
EPX: SHARED MEMORY OPTIMIZATION FOR MULTI-PROCESSOR NODES (2/2)

- **On-going optimization (PhD M. Sridi, CEA, INRIA, 2013-2016)**

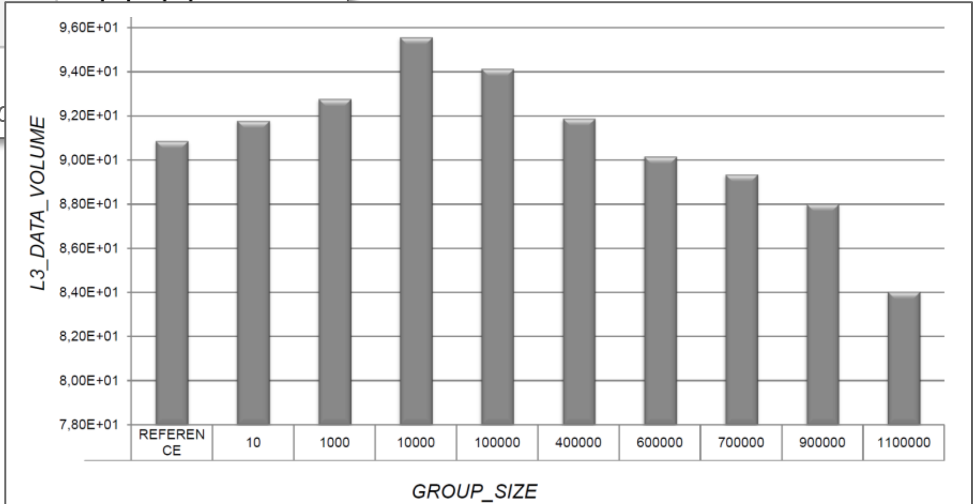


« Structures of Arrays »

« Arrays of Struc



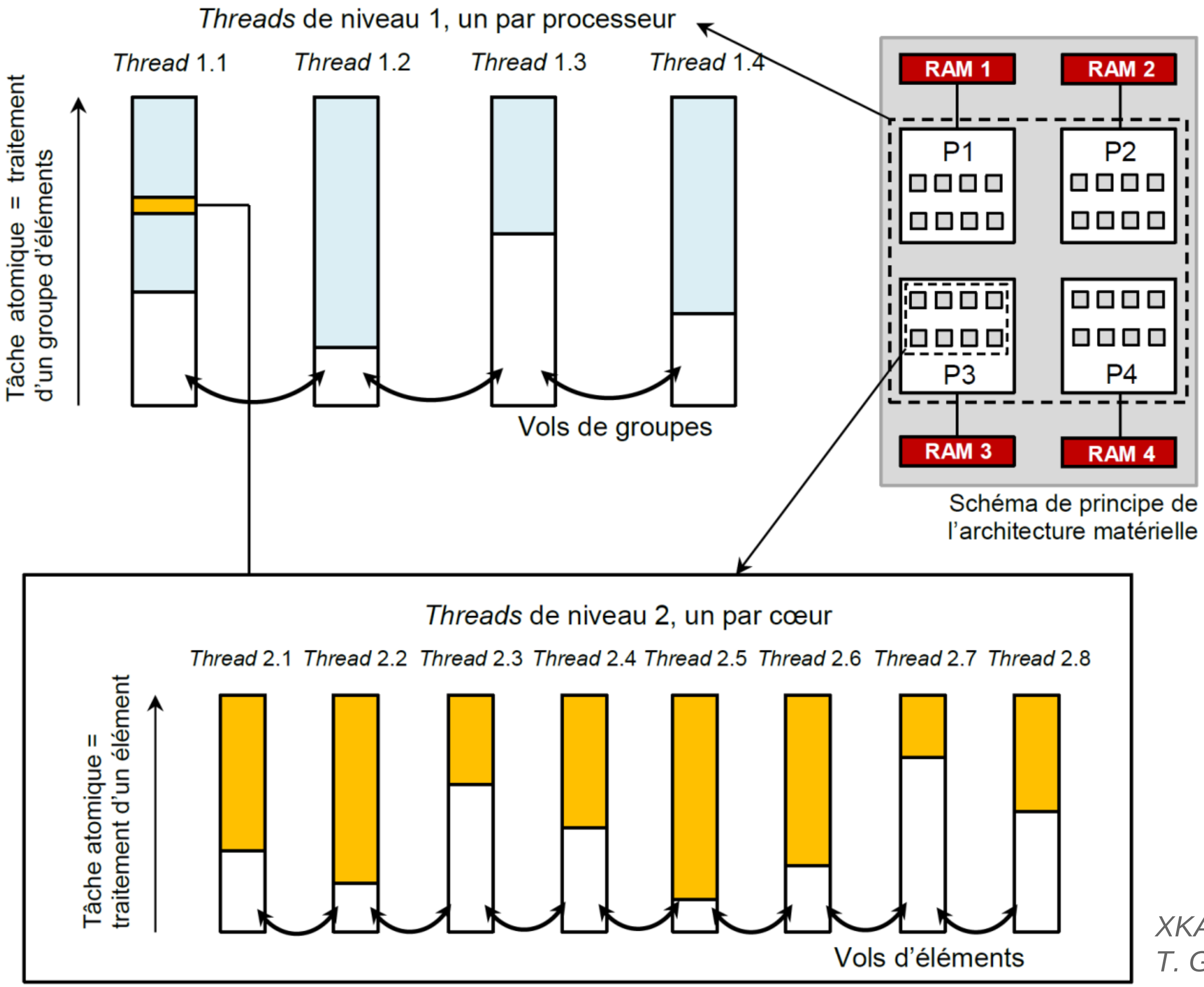
L2-Cache misses reduction



L3-Cache filling improvement

EPX: SHARED MEMORY OPTIMIZATION FOR MULTI-PROCESSOR NODES (2/2)

- **On-going optimization (PhD M. Sridi, CEA, INRIA, 2013-2016)**



XKAAPI – v3, 2015, January
T. Gautier, INRIA

STRATEGY TOPICS FOR EXASCALE COMPUTING

STRONG OPTIMIZATION VS GENERALITY & FLEXIBILITY

- **Petascale = (big) clusters of PCs + accelerators**

  - Performance obtained from precisely mastering data exchanges and memory accesses
  - Relevant due to relative architectural uniformity
  - Proximity between large scale supercomputers and local development hardware

STRONG OPTIMIZATION VS GENERALITY & FLEXIBILITY

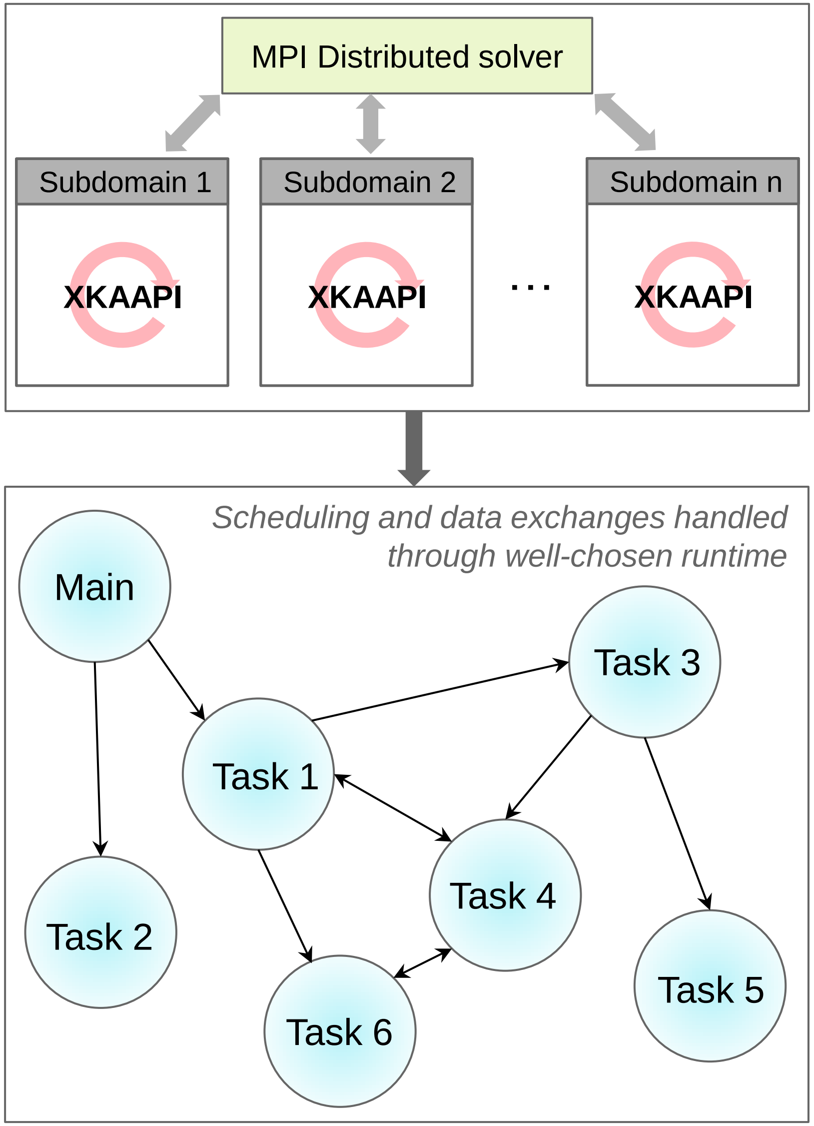
- **Petascale = (big) clusters of PCs + accelerators**

  - Performance obtained from precisely mastering data exchanges and memory accesses
  - Relevant due to relative architectural uniformity
  - Proximity between large scale supercomputers and local development hardware

- **Exascale = *[undefined variable]***

  - Local performance constraints due to power management policies
  - No guaranty on where the tasks are executed
  - Probable strong heterogeneity
    - From one supercomputer to another
    - Inside one supercomputer

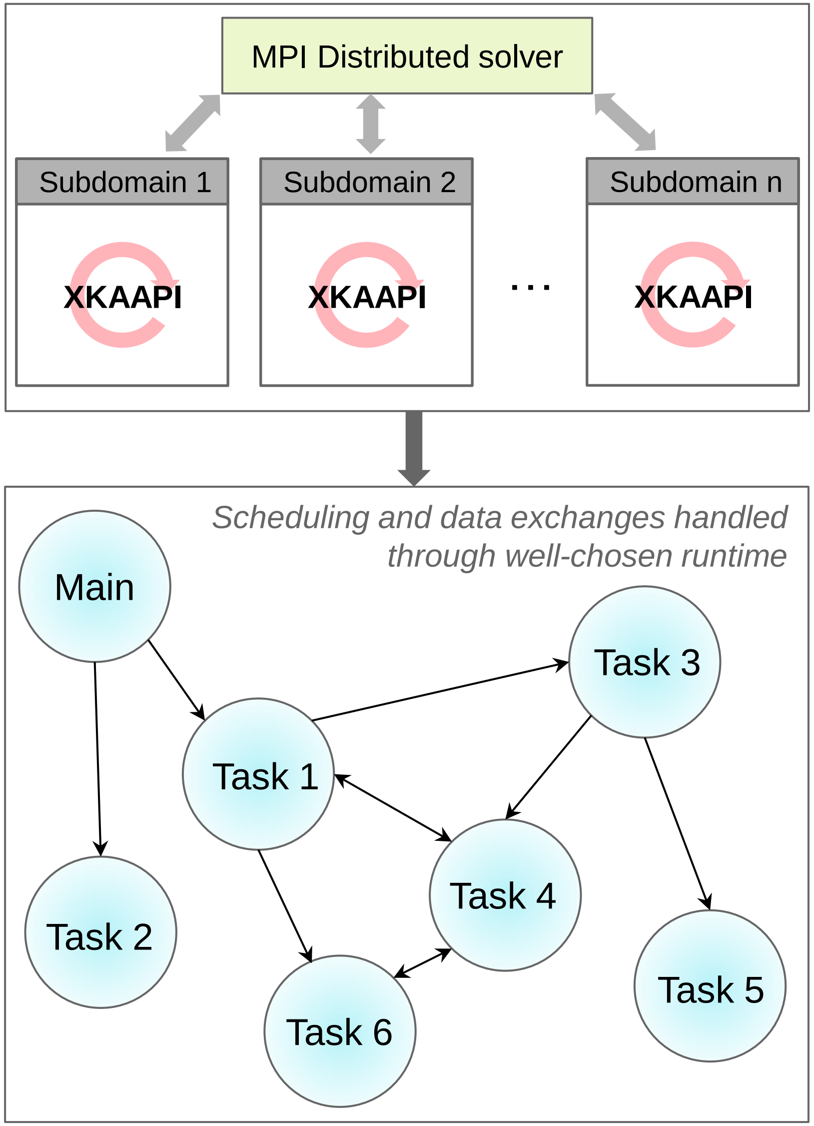
STRONG OPTIMIZATION VS GENERALITY & FLEXIBILITY



- **Exascale = *[undefined variable]***

  - Local performance constraints due to power management policies
  - No guaranty on where the tasks are executed
  - Probable strong heterogeneity
    - From one supercomputer to another
    - Inside one supercomputer

STRONG OPTIMIZATION VS GENERALITY & FLEXIBILITY



- **Exascale = *[undefined variable]***

  - Local performance constraints due to power management policies
  - No guaranty on where the tasks are executed
  - Probable strong heterogeneity
    - From one supercomputer to another
    - Inside one supercomputer

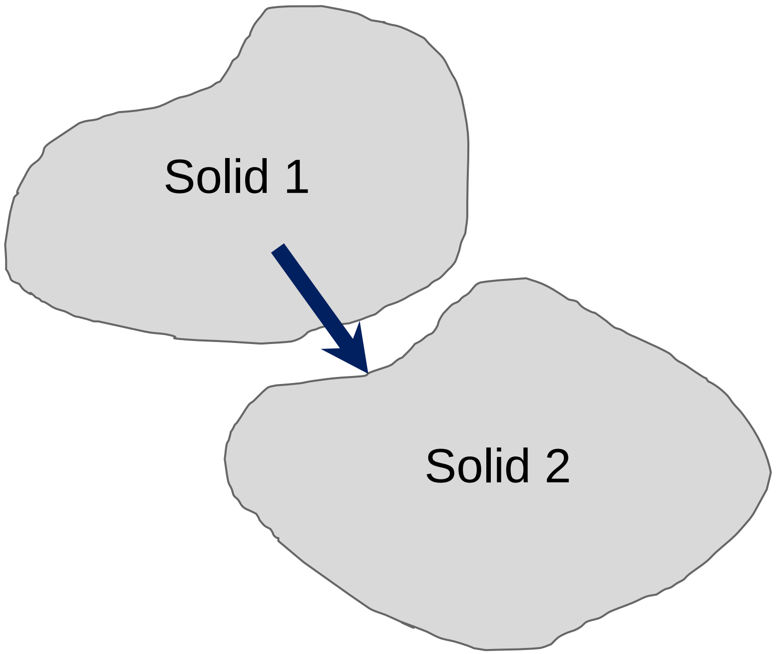
- **Shortcomings**
  - Runtime must handle efficiently shared and distributed exchanges
  - Deep evolution of the data structure to expect…

NEED FOR ASYNCHRONICITY AND ASSOCIATED ISSUES

- **Need for asynchronicity**
  - Minimum constraints on data flow
  - Fill-in the computing cores at their maximum

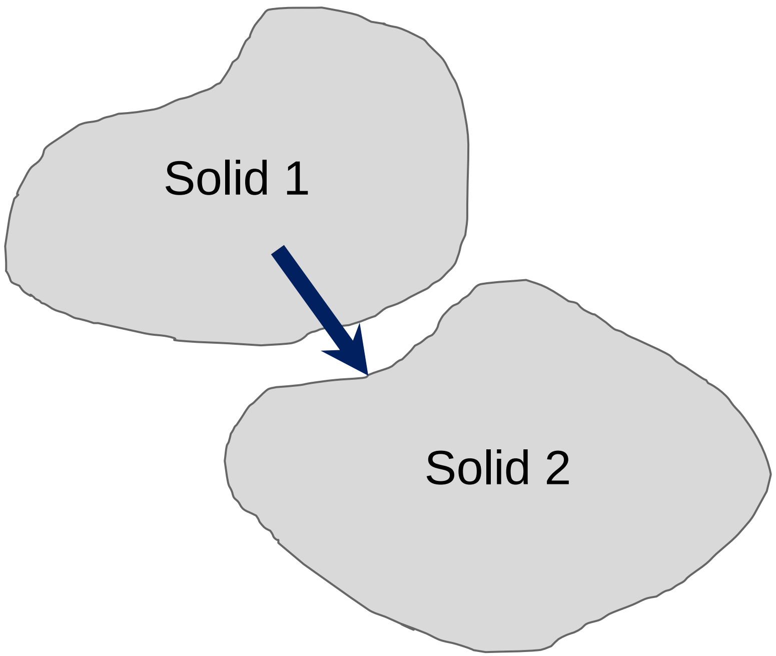
NEED FOR ASYNCHRONICITY AND ASSOCIATED ISSUES

- **Need for asynchronicity**
  - Minimum constraints on data flow
  - Fill-in the computing cores at their maximum



NEED FOR ASYNCHRONICITY AND ASSOCIATED ISSUES

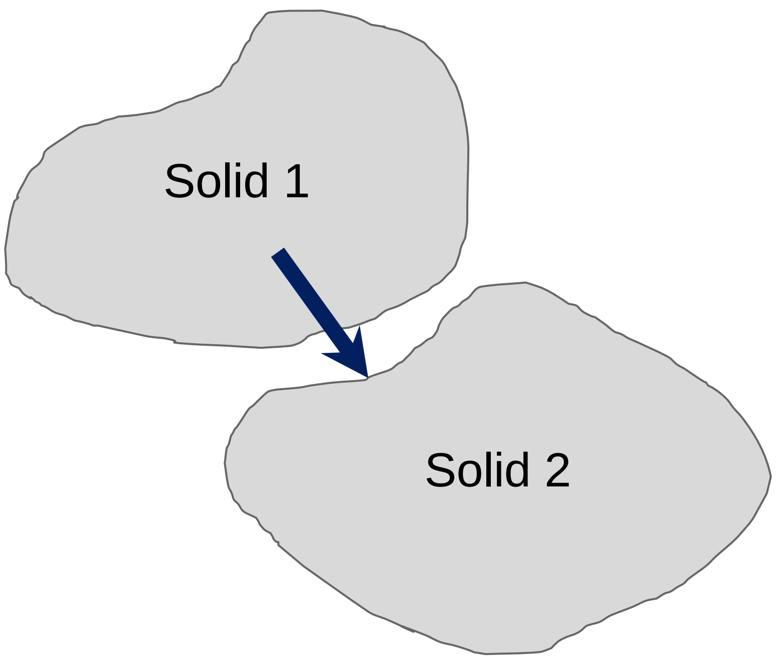
- **Need for asynchronicity**
  - Minimum constraints on data flow
  - Fill-in the computing cores at their maximum

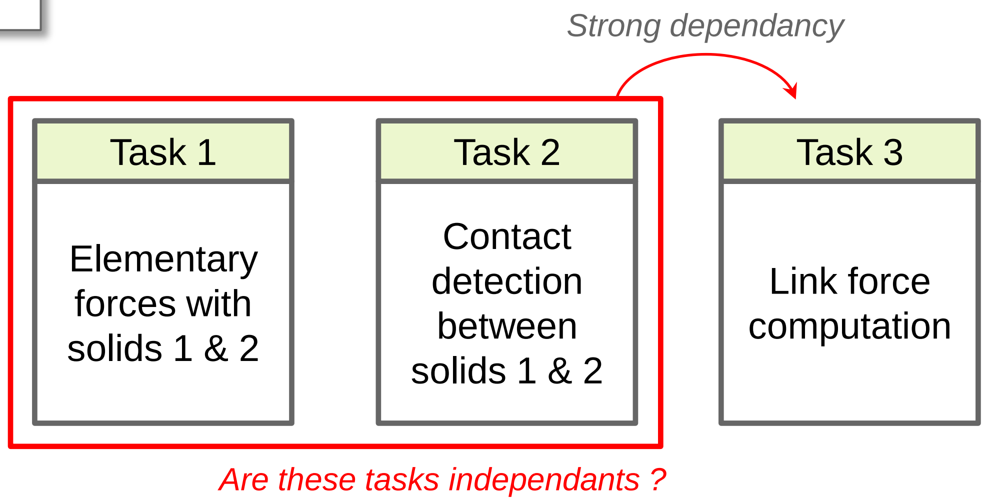


Task 1	Task 2	Task 3
Elementary forces with solids 1 & 2	Contact detection between solids 1 & 2	Link force computation

NEED FOR ASYNCHRONICITY AND ASSOCIATED ISSUES

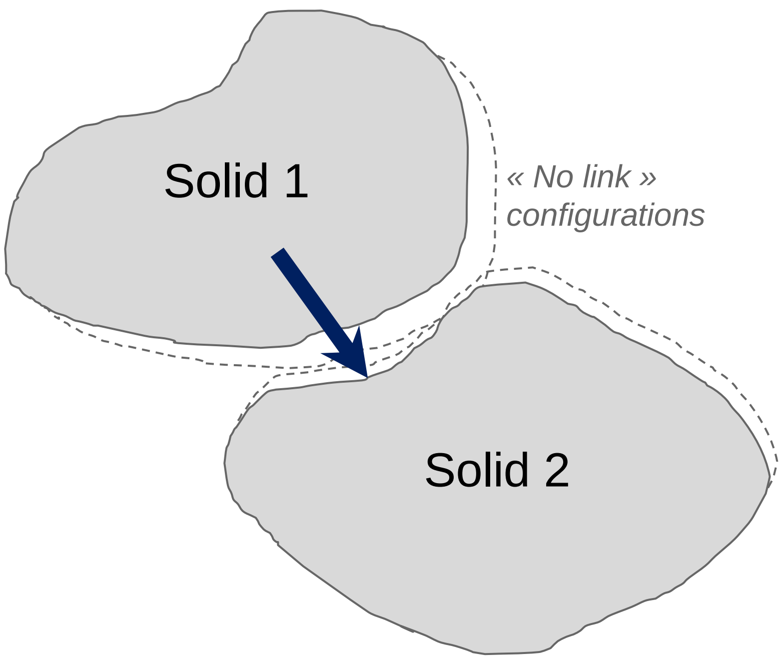
- **Need for asynchronicity**
  - Minimum constraints on data flow
  - Fill-in the computing cores at their maximum

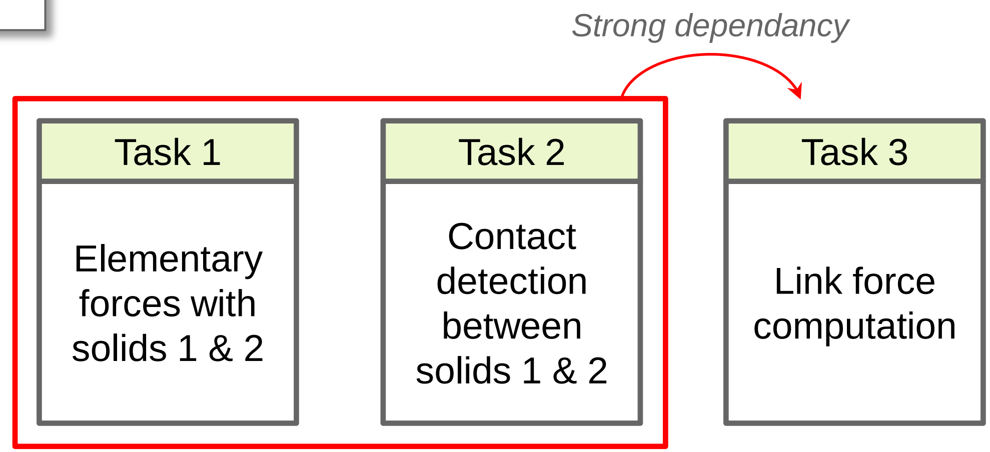




NEED FOR ASYNCHRONICITY AND ASSOCIATED ISSUES

- **Need for asynchronicity**
  - Minimum constraints on data flow
  - Fill-in the computing cores at their maximum

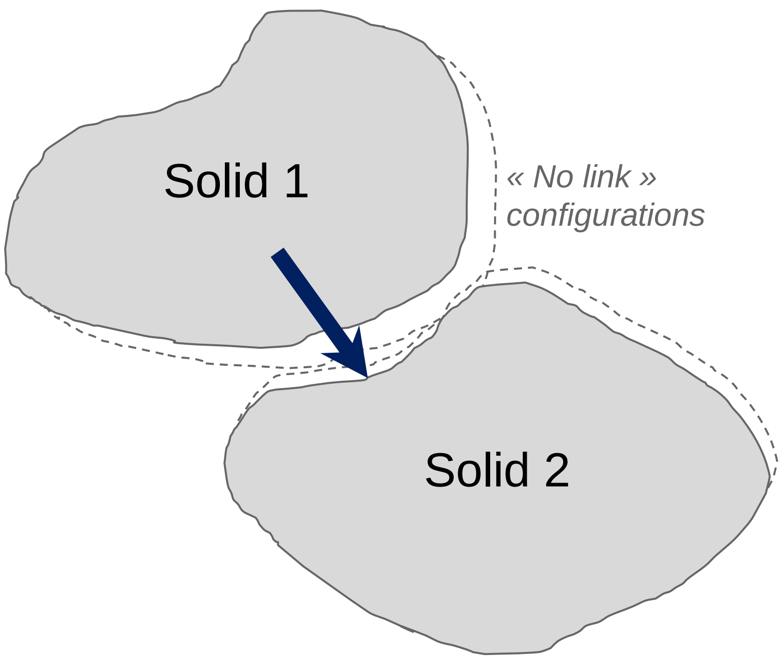


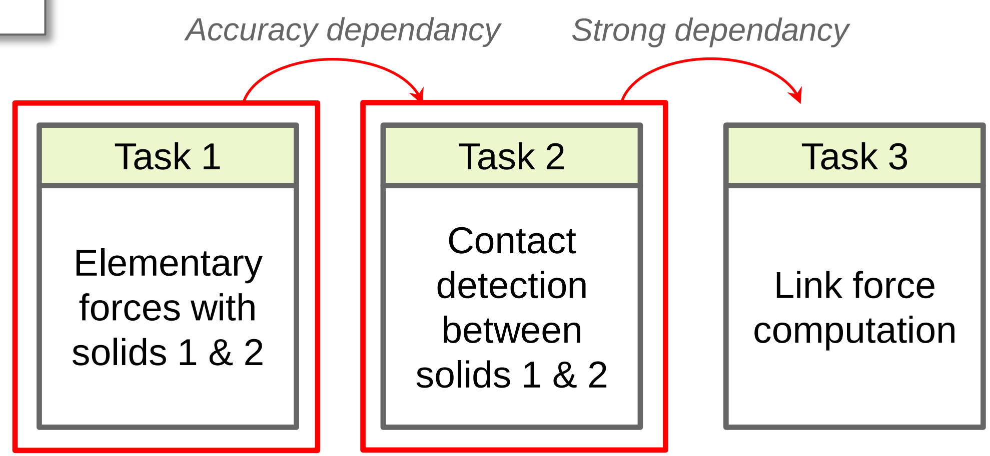


Are these tasks independants ?

NEED FOR ASYNCHRONICITY AND ASSOCIATED ISSUES

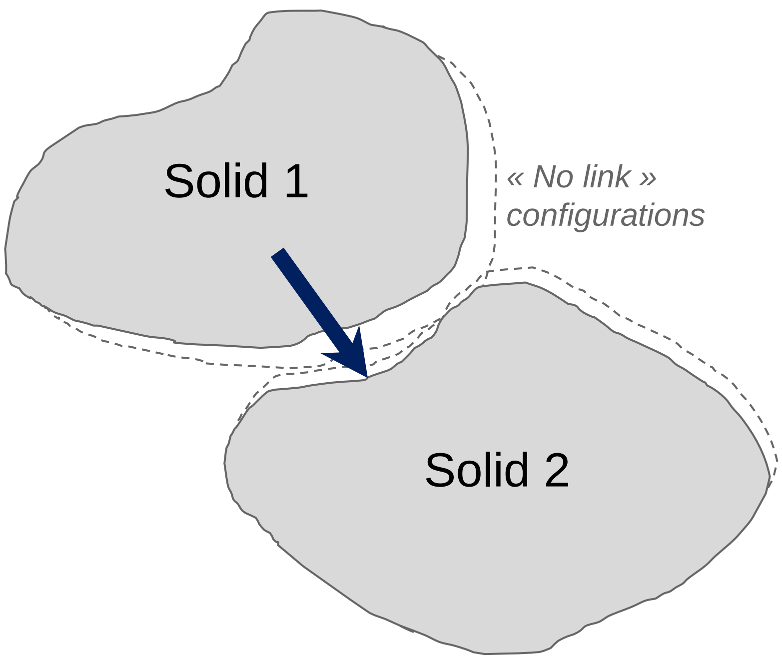
- **Need for asynchronicity**
  - Minimum constraints on data flow
  - Fill-in the computing cores at their maximum

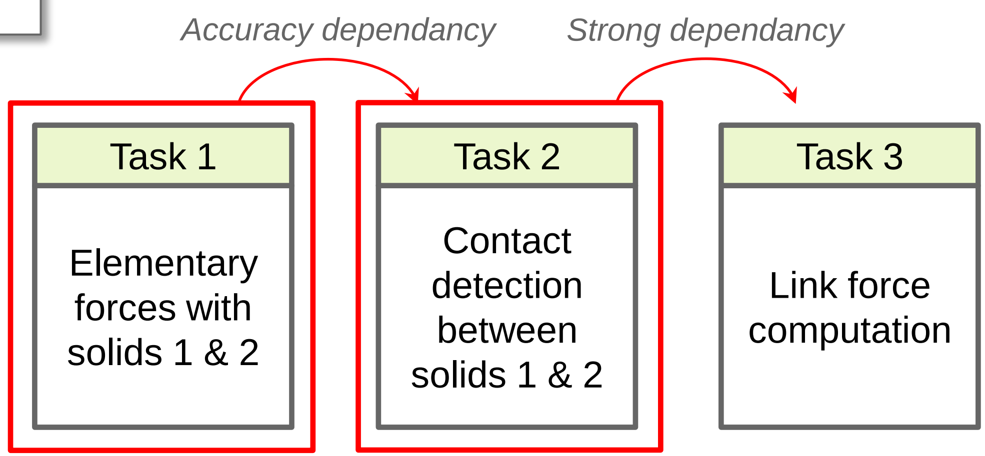




NEED FOR ASYNCHRONICITY AND ASSOCIATED ISSUES

- **Need for asynchronicity**
  - Minimum constraints on data flow
  - Fill-in the computing cores at their maximum





- **Beware the priorities**
  - Final cut in the hands of physics and applied mathematics
  - Asynchronicity implies strong algorithmic innovation

SOME CONCLUSIONS AND PROSPECTS

CONCLUSIONS AND PROSPECTS

- **Main conclusions for explicit fluid-structure dynamics**
  - Petascale
    - Efficiency for Petascale obtained through combining distributed and shared memory
    - Optimization sticking closely to the hardware main characteristics
    - Use of middleware runtime for dynamic scheduling anytime possible
  - Extension to exascale
    - Need for asynchronous algorithms preserving the accuracy of the physical solution
    - Potential restructuration of the data flow to transfer the adaptation to hardware to a generic runtime

- **Prospects**
  - Autotuning for the application to correct itself seeking optimal parallel performance
  - Need for robust on-the-fly diagnostics (internal or external)
  - Continuous algorithmic innovation for load-balancing (growing interest for AMR…)

EUROPLEXUS Software

http://www-epx.cea.fr

THANK YOU FOR YOUR ATTENTION

Commissariat à l'énergie atomique et aux énergies alternatives
Centre de Saclay | 91191 Gif-sur-Yvette Cedex
T. +33 (0)1 69 08 40 18| F. +33 (0)1 69 08 76 19

Etablissement public à caractère industriel et commercial | RCS Paris B 775 685 019

Direction de l'Energie Nucléaire
Département de Modélisation des Systèmes et Structures
Service d'Etudes Mécaniques et Thermiques