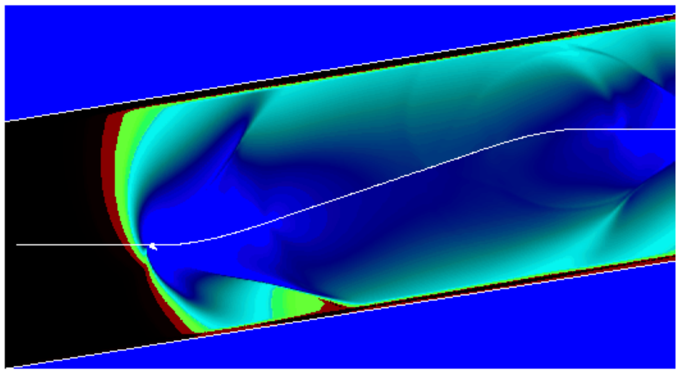
2DSPEC – 3DSPEC
Object decomposed Spectral Element Code

G. Festa, J.-P. Vilotte and E. Delavaud

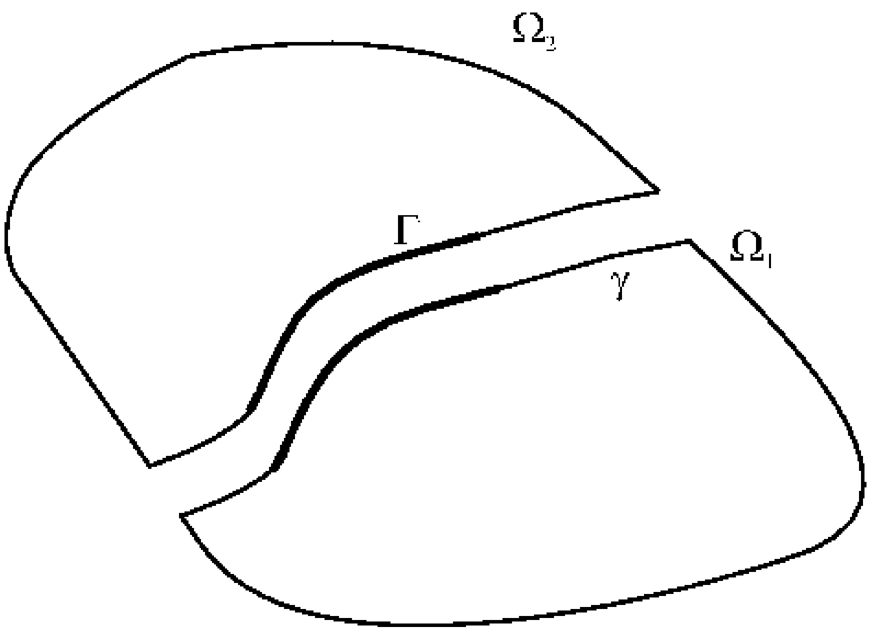
The program development is supported by the SPICE EU-project



SPICE meeting
Munich
18-20 July 2005

Weak formulation of elastodynamics

$$\int_{\Omega} \rho\, \mathbf{w}\, \dot{\mathbf{v}}\, d\Omega = \int_{\Omega} \mathbf{w}\, \mathbf{f}^{ext} d\Omega - \int_{\Omega} \nabla \mathbf{w} : \mathbf{c} : \nabla \mathbf{u}\, d\Omega + \int_{\Gamma} \mathbf{w}\, \mathbf{T}\, d\Gamma$$



Ω elastic domain with :

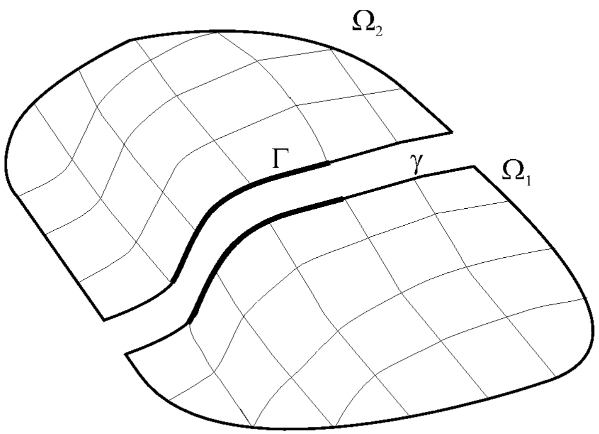
$\Omega = \Omega_1 \cup \Omega_2$

$m(\Omega_1 \cap \Omega_2) = 0$

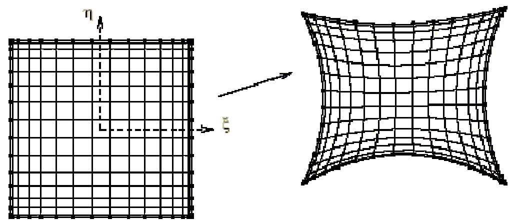
$\partial\Omega_1 \cap \partial\Omega_2 = \Gamma$

Γ accounts for a surface on which Neumann or Robin b.c. are imposed
(fault, Mortar …etc)

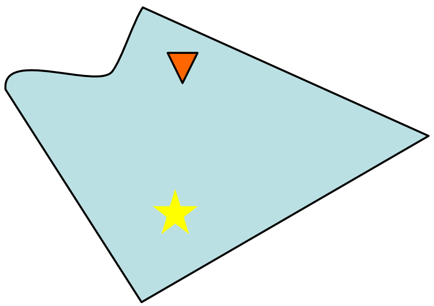
Quadrangulation of the domain



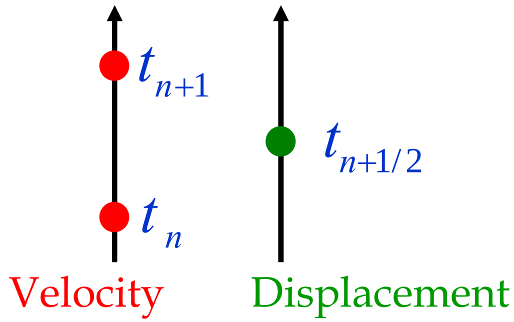
Shape function evaluation



Source and Receiver location



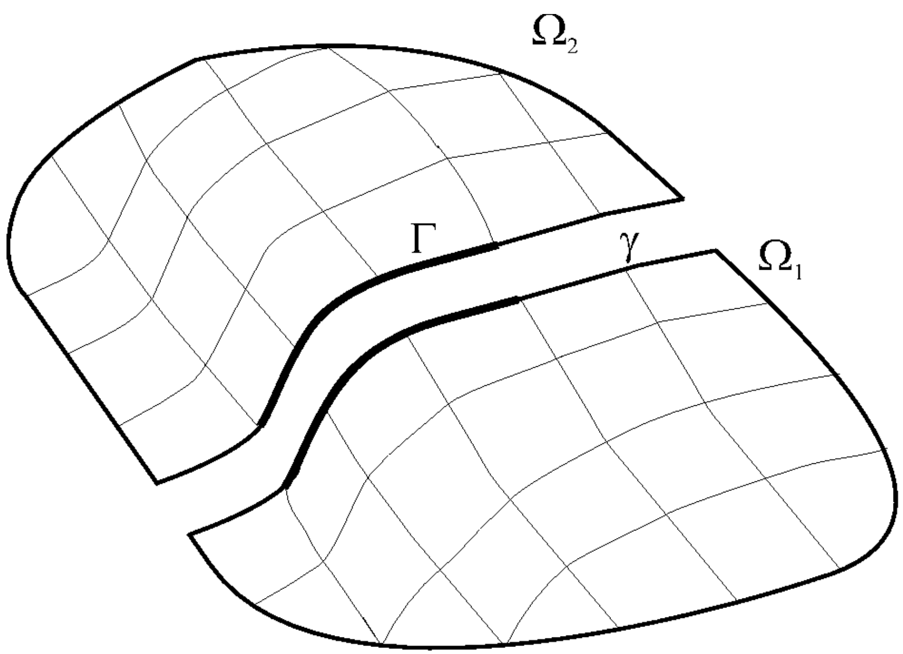
Time evolution



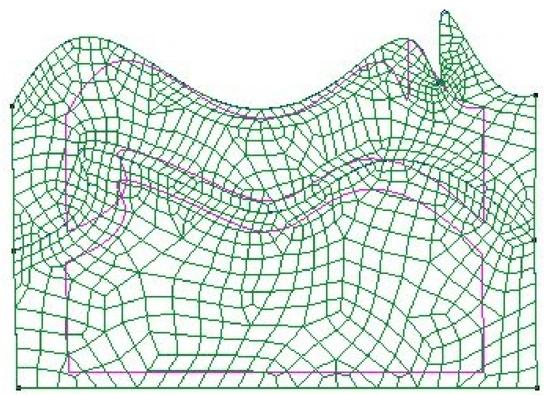
QUADRANGULATION

Separation of the domain in (quadrangluar) elements

$$\sum_e \int_{\Omega_e} \rho \mathbf{w} \dot{\mathbf{v}} d\Omega = \sum_e \int_{\Omega_e} \mathbf{w} \mathbf{f}^{ext} d\Omega - \sum_e \int_{\Omega_e} \nabla \mathbf{w} : \mathbf{c} : \nabla \mathbf{u} d\Omega + \sum_e \int_{\Gamma \cap \partial \Omega_e} \mathbf{w} \mathbf{T} d\Gamma$$

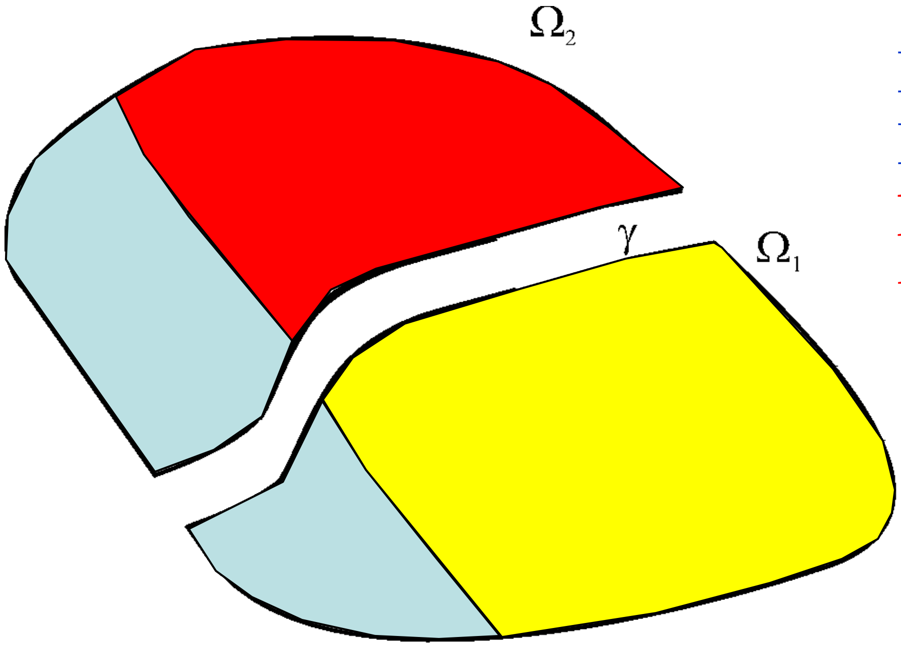


Complex domains are meshed by GiD
http://gid.cimne.upc.es



MESH PARTITIONING

Decomposition of the computational volume into subdomains

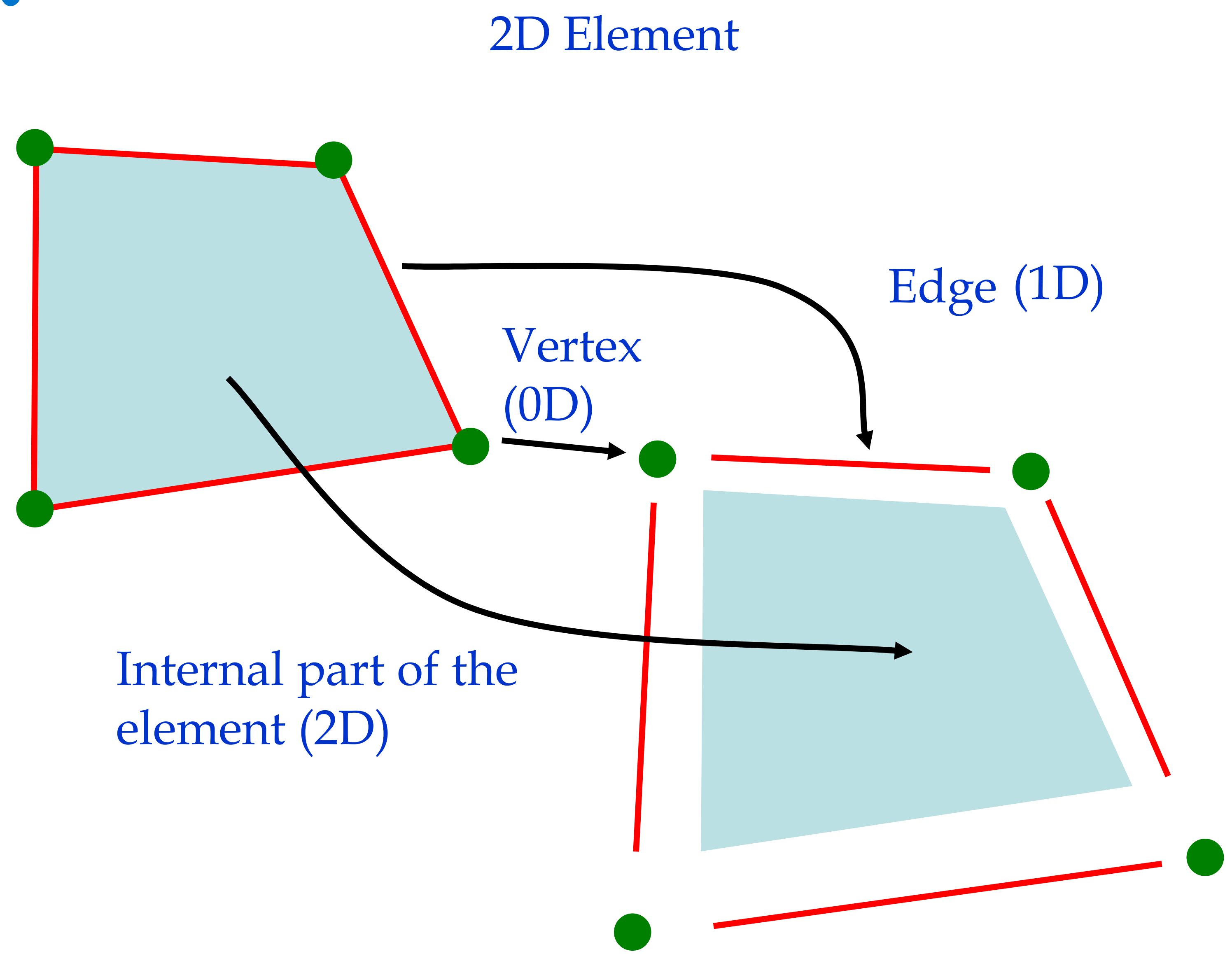


Partitioning graph software
Metis and ParMetis
http:// www-users.cs.umn.edu/~karypis/metis/

Any subdomain is the object of a processor

Computation is done at the level of the element

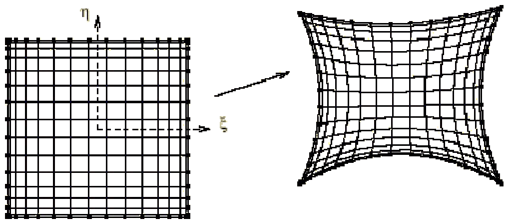
It enables different kinds of comunications


2D Element
Edge (1D)
Vertex
(0D)
Internal part of the
element (2D)

PRE-COMUPUTATION

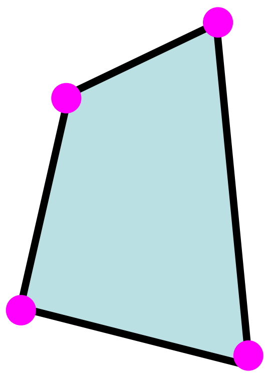
Precomputation is ascribed to the whole element, within its subparts (faces, edges and vertices)

Shape functions

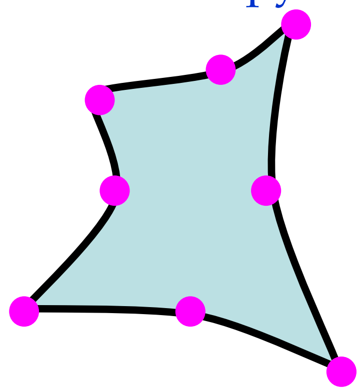


Lagrange elements

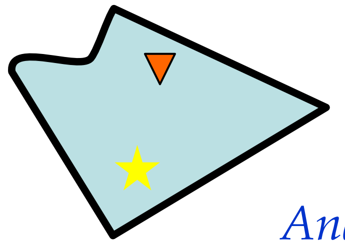
Linear



Serenditipy



Source and receiver location



Determination of the $\xi-\eta$ coordinates:

Analytically for linear elem.s
By *triangulation* for quad. el.

For the source:

$$\int_{\Omega_e} \nabla \mathbf{w} : \mathbf{M}_0 \, \delta(\mathbf{x}-\mathbf{x}_0) \, d\Omega = \nabla \mathbf{w}(\mathbf{x}_0) : \mathbf{M}_0$$

Space discretization

$$\sum_e \int_{\Omega_e} \rho\, \mathbf{w}^{eh}\, \dot{\mathbf{v}}^{eh} J_e d\square = \sum_{e_{source}} \xi_{\mathbf{x}} \nabla_{\xi} \mathbf{w}^{eh} : \mathbf{M}_0 f(t) -$$

$$\sum_e \int_{\Omega_e} \xi_{\mathbf{x}} \nabla_{\xi} \mathbf{w}^{eh} : \mathbf{c} : \nabla \mathbf{u}^{eh} J_e d\square + \sum_e \int_{\Gamma \cap \partial\Omega_e} \mathbf{w}^{eh}\, \mathbf{T}^{\partial eh} J_{\partial e} d\_$$

Numerical quadrature

- Gauss-Lobatto-Legendre (GLL) quadrature

$$\int_{-1}^{1} f(\zeta) d\zeta = \sum_{k=0}^{N} f(\zeta_k)\omega_k$$

- Lagrangian interpolation of fields on GLL points

$$\mathbf{M}\dot{\mathbf{v}} = \mathbf{F}^{ext} - \mathbf{F}^{\text{int}}(\mathbf{u}) + \mathbf{B}^T \mathbf{T}$$

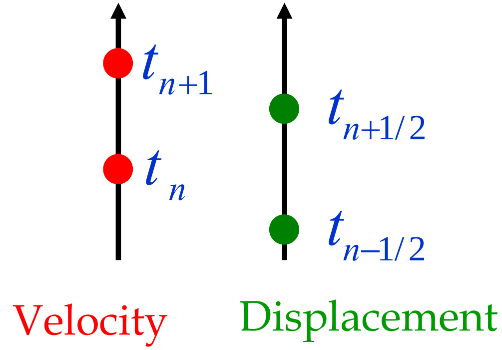
$$\mathbf{F}^{\text{int}}(\mathbf{u}) = (\mathbf{K}\mathbf{u}) = \mathbf{D}_{\xi}{}^T \mathbf{a}\, \mathbf{D}_{\xi}\, \mathbf{u}$$

Diagonality of mass matrix leads to explicit schemes

Time discretization : Newmark velocity scheme

$$\mathbf{M}\,\frac{\mathbf{v}_{n+1}-\mathbf{v}_{n}}{\Delta t}=\mathbf{F}^{ext}_{n+1/2}-\mathbf{F}^{\text{int}}\left(\mathbf{u}_{n+1/2}\right)+\mathbf{B}^{T}\mathbf{T}_{n+1/2}$$

$$\mathbf{u}_{n+1/2}=\mathbf{u}_{n-1/2}+\Delta t\,\mathbf{v}_{n}$$

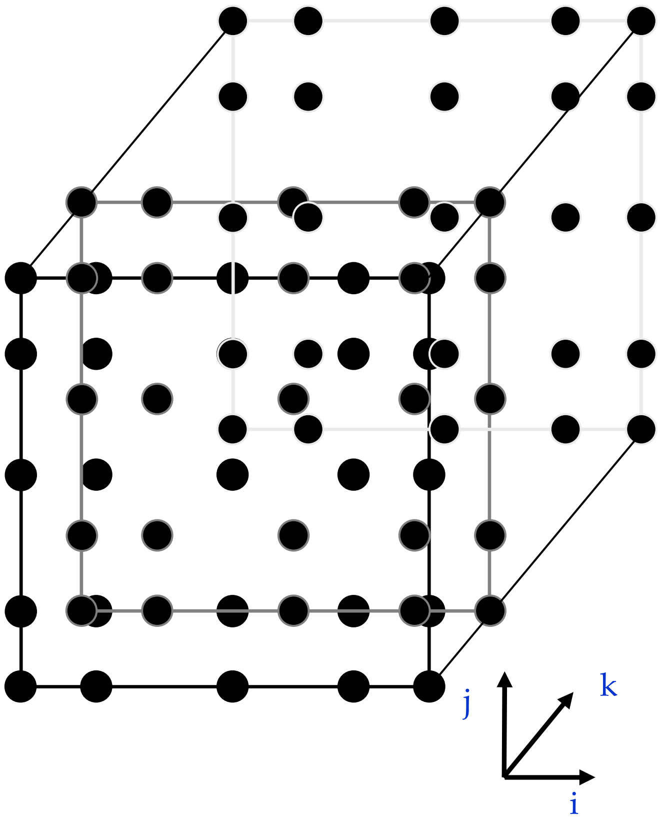


Time stepping – Computation time is spent in

- Computation of derivatives – Internal Forces
- Communication of internal forces between elements
- Upgrade of kinematic fields (One – to – one)

ORDERING OF THE POINTS

- Ordering of GLL plays an important role in the computation
- Tensorization requires the value of the displacement over all the element (boundaries included)



u_{000}	u_{100}	$\cdots$	u_{N00}	u_{010}	u_{110}	$\cdots$	u_{NNN}

N^3

$$\mathbf{D}_1 u^e = D_{ip} u^e_{pjk}$$

$$\mathbf{D}_2 u^e = D_{ip} u^e_{ipk}$$

$$\mathbf{D}_3 u^e = D_{ip} u^e_{ijp}$$

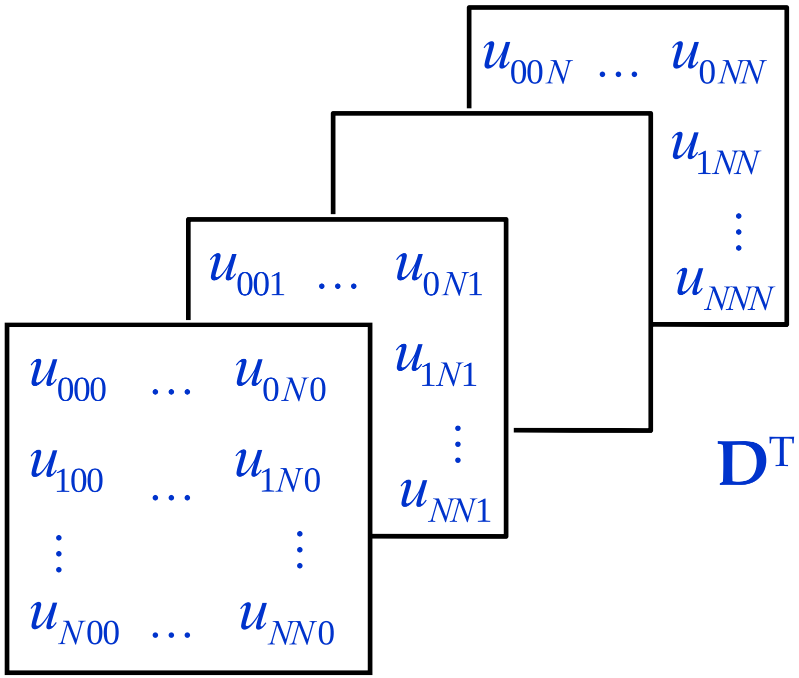
DERIVATIVES

$$\mathbf{D}_1 u^e = D_{ip} u^e_{pjk}$$

$$\mathbf{D} \quad \begin{bmatrix} u_{000} & u_{010} & \cdots & \cdots & u_{0NN} \\ u_{100} & u_{110} & \cdots & \cdots & u_{1NN} \\ & & & & \vdots \\ u_{N00} & u_{N10} & \cdots & \cdots & u_{NNN} \end{bmatrix}$$

$$\mathbf{D}_3 u^e = D_{ip} u^e_{ijp}$$

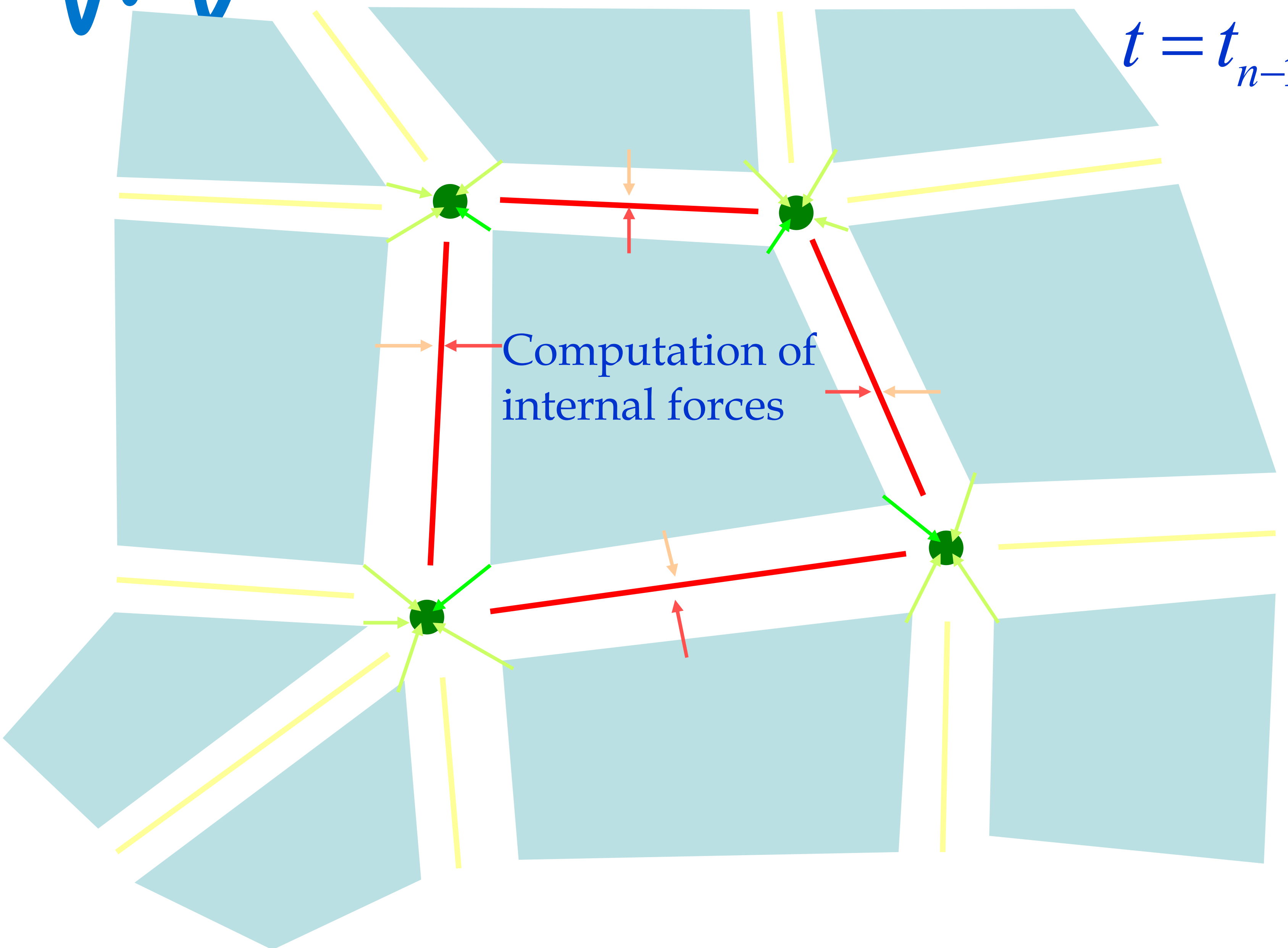
$$\mathbf{D}_2 u^e = D_{ip} u^e_{ipk}$$

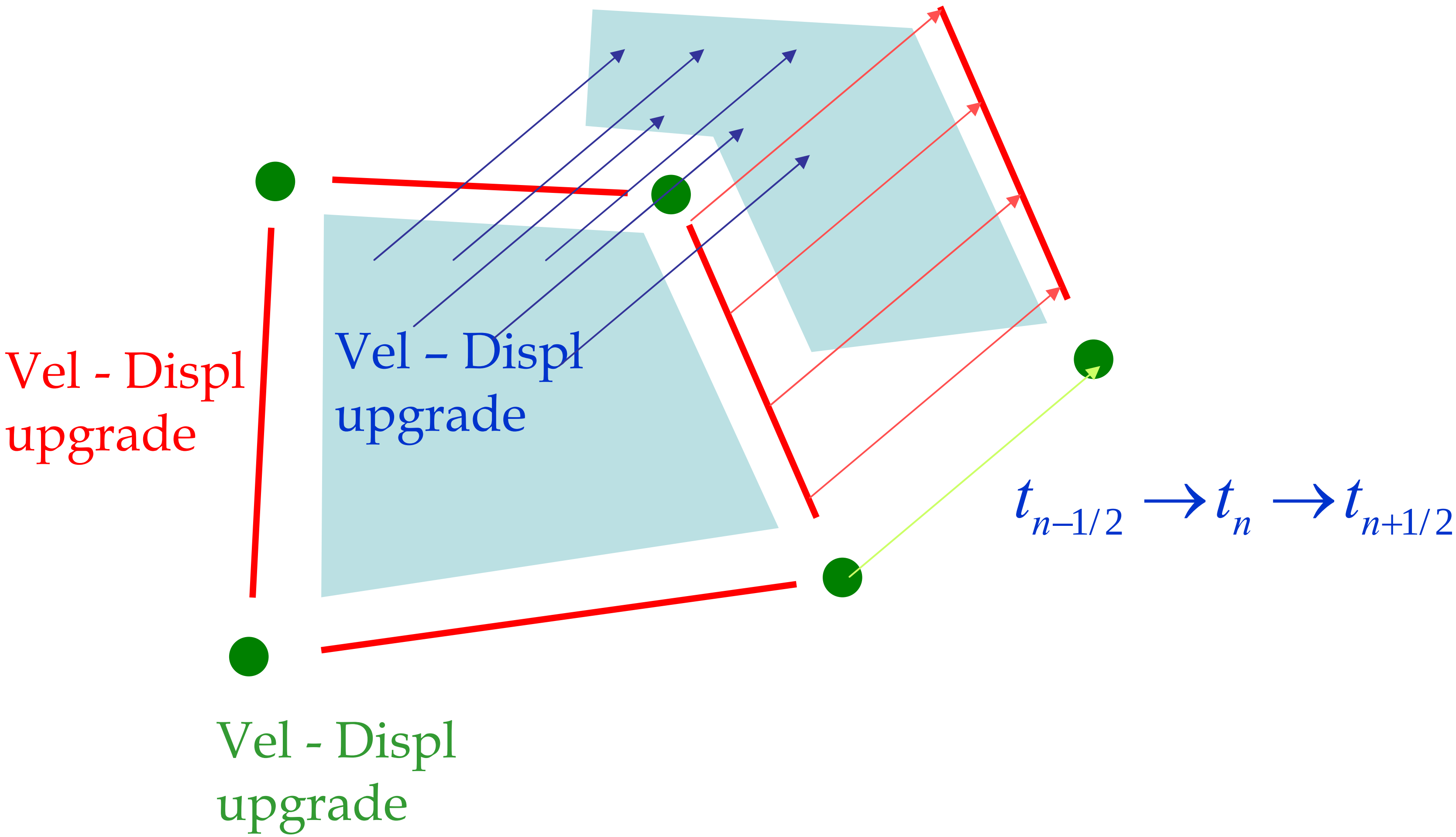


$\mathbf{D}^{\mathrm{T}}$

$$\begin{bmatrix} u_{000} & u_{001} & \cdots & u_{00N} \\ u_{100} & u_{101} & \cdots & u_{10N} \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ u_{NN0} & u_{NN1} & \cdots & u_{NNN} \end{bmatrix} \quad \mathbf{D}^{\mathrm{T}}$$

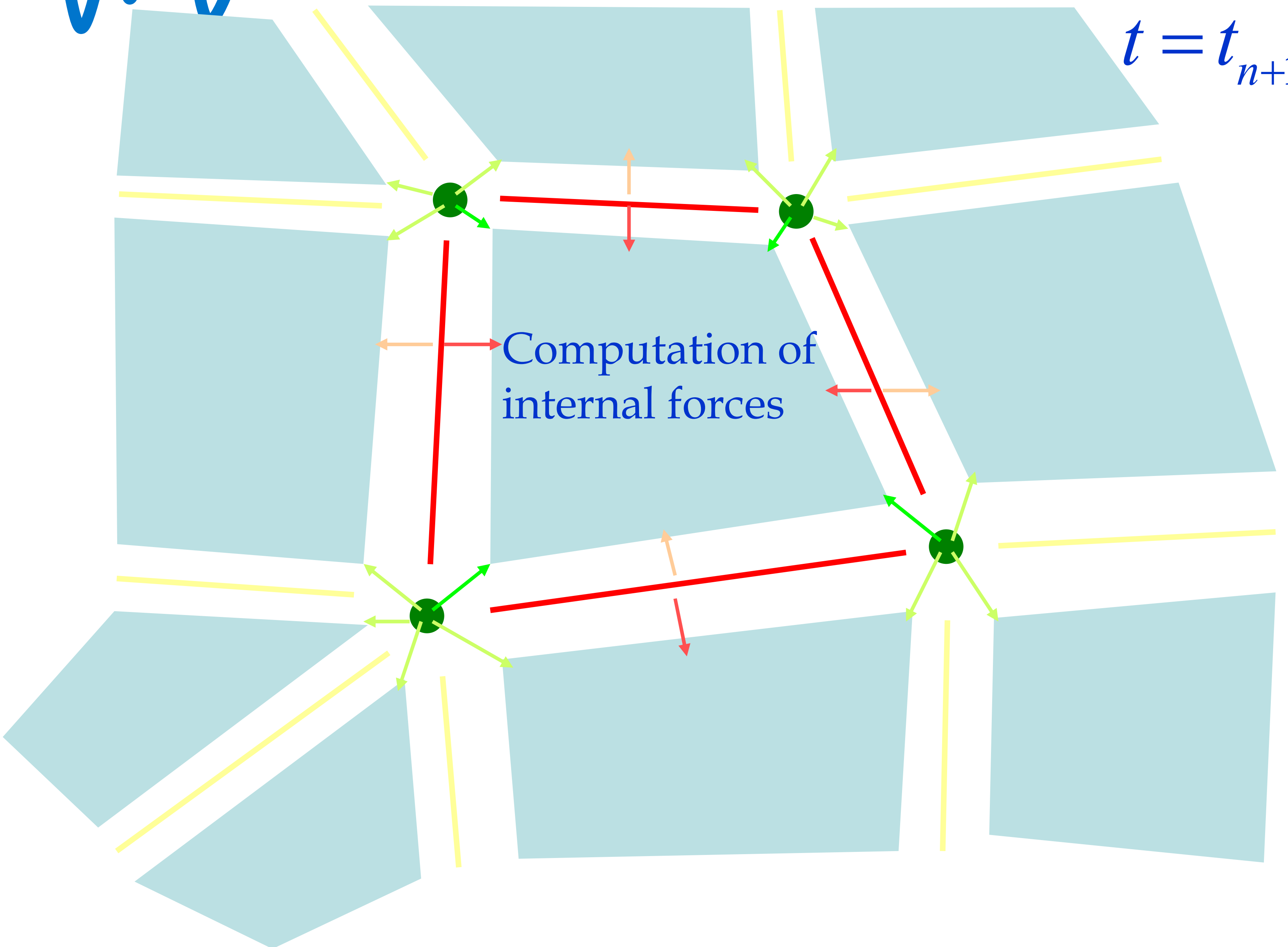
$t = t_{n-1/2}$




Vel - Displ
upgrade
Vel – Displ
upgrade
Vel - Displ
upgrade
$t_{n-1/2} \rightarrow t_n \rightarrow t_{n+1/2}$

COMMUNICATIONS

$t = t_{n+1/2}$

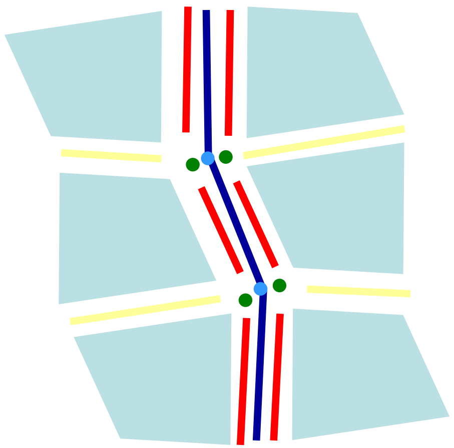


SPECIAL OBJECTS

Special objects (s-o's), such as faults, Neumann conditions, Mortar interfaces and plane waves are introduced via the term

$$\int_{\Gamma} \mathbf{w}\,\mathbf{T}\,d\Gamma$$

They are thought as a composition of objects of minor dimension



- For the computation, s-o's need to know the IF from both media
- S-o's give back the value of the traction, that has to be sommed to the partial IF computation

PARALLELIZATION

- MPI Communication requires the exchange of the IF between processors
- A special-object on the face of any subdomain collects IF from the bulk
- Asynchronous SEND-RECEIVE is performed for the interchanges

CONCLUSIONS

- 2DSPEC – 3DSPEC is an object-decomposed code, with independent, different dimensions objects (internal elements, faces, edges, vertices)
- Derivative computation is perfomed at the element level to take advantage from the tensorization
- Hierarchic communication is performed
- Flexibility results from the same procedure used to include faults, mortar interface, plane waves, parallelization

SOON

- Soon on the SPICE website ….