3D adaptive finite elements with high aspect ratio for the computation of dendritic growth with convection

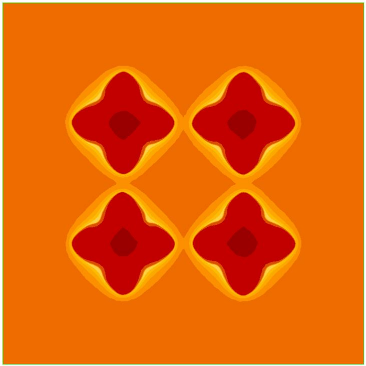
Jacek Narski and Marco Picasso

Institut d'Analyse et Calcul Scientifique,
Ecole Polytechnique Fédérale de Lausanne,
1015 Lausanne, Switzerland

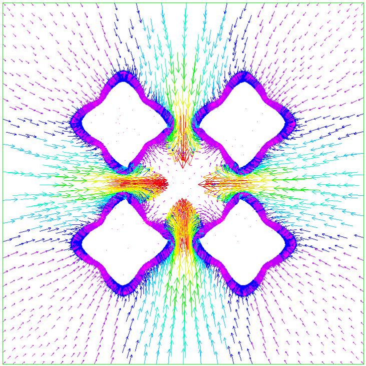
collaboration with Michel Rappaz and Alain Jacot, Material Science Institute, EPFL

Outline

- Goal: take into account fluid flow between dendrites due to variable solid and liquid densities (shrinkage).
- Tool: adaptive finite elements with high aspect ratio.

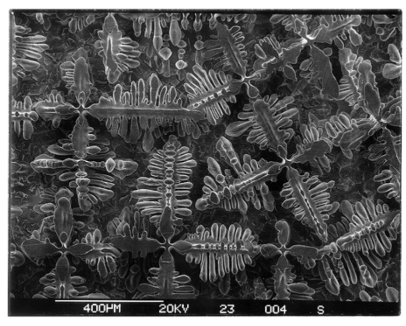


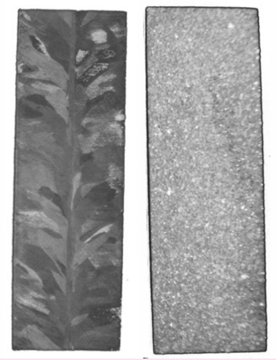
concentration



velocity

Solidification: Dendritic growth



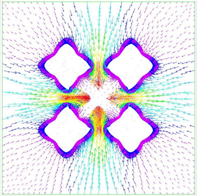


Lenght scales

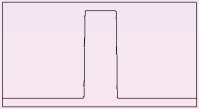
- Workpiece $\simeq 1\ m$
- Dendrite $\simeq 10^{-3}\ m$
- Solid-liquid interface $\simeq 10^{-8}\ m$ (100 atoms)

The unknowns

concentration



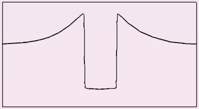
velocity



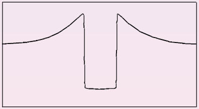
Phase field ϕ

$\phi = 0$: liquid

$\phi = 1$: solid



Concentration c



Velocity $\mathbf{u}$

Temperature $T(x, y, t)$ is a parameter

The model with convection

Governing equations

- Phase field:

$$\frac{\partial \phi}{\partial t} - \operatorname{div}\left(A(\nabla\phi)\nabla\phi\right) - S(c,\phi) = 0.$$

- Concentration:

$$\frac{\partial c}{\partial t} + \operatorname{div}\Big(G(\phi,\mathbf{u},c)\Big) - \operatorname{div}\Big(D_1(\phi)\nabla c + D_2(c,\phi)\nabla\phi\Big) = 0$$

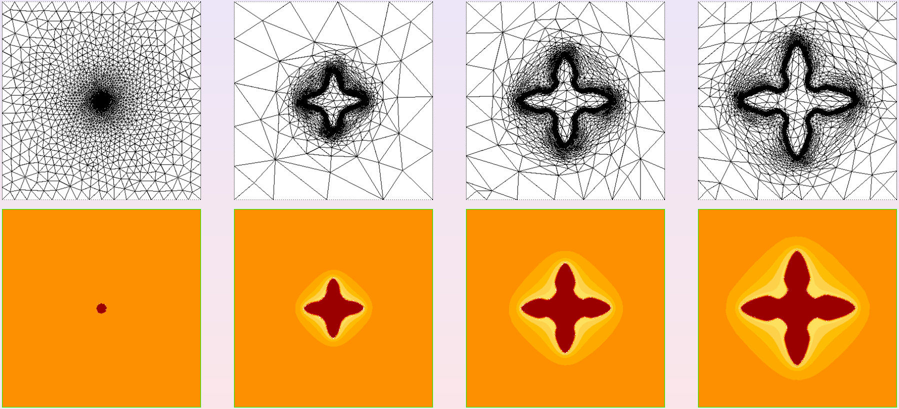
- Fluid flow:

$$\rho = \phi\rho_s + (1-\phi)\rho_l$$

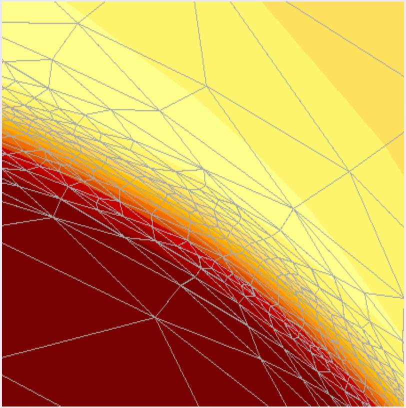
$$\rho\frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u}\cdot\nabla)\mathbf{u} - \mu\Delta\mathbf{u} + \nabla p + \operatorname{div}\Big(\mathbf{F}(\phi,\mathbf{u})\Big) + \frac{1}{\epsilon}\phi^2\mathbf{u} = 0$$

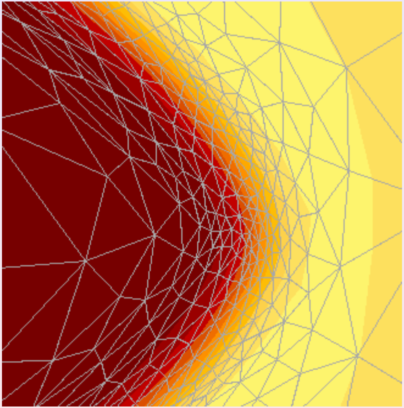
$$\frac{\partial \rho}{\partial t} + \operatorname{div}(\rho\mathbf{u}) = 0$$

Adaptive finite elements with large aspect ratio



Adaptive finite elements with large aspect ratio





Same precision with isotropic meshes: 10 to 100 times more vertices!

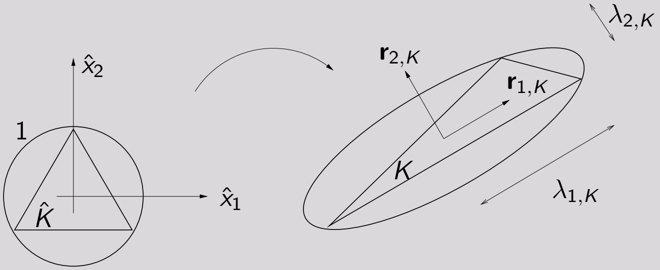
A posteriori error estimates

- Dendritic growth without convection

$$\frac{\partial \phi}{\partial t} - \text{div}\,(A(\nabla\phi)\nabla\phi)) - S(c,\phi) = 0 \qquad \text{in } \Omega \times (0,T),$$
$$\frac{\partial c}{\partial t} - \text{div}\,\Big(D_1(\phi)\nabla c + D_2(c,\phi)\nabla\phi\Big) = 0 \qquad \text{in } \Omega \times (0,T).$$

- Existence for low physical anisotropy, a priori error estimates (Burman Rappaz M3AS 2003).
- A posteriori error estimates, adaptive FE with high aspect ratio, numerical study of the effectivity index (Burman Picasso J. Interfaces Free Boundaries 2003).
- No theory with convection

Error estimator for meshes with high aspect ratio



$e = c - c_h,$

$$\eta_K^2 = \int_{t^{n-1}}^{t^n} \frac{1}{\sqrt{\lambda_{2,K}}} \left\| \left[\frac{\partial c_h}{\partial n} \right] \right\|_{L^2(\partial K)} \sqrt{\lambda_{1,K}^2 \left(\mathbf{r}_{1,K}^T G_K(e) \mathbf{r}_{1,K} \right) + \lambda_{2,K}^2 \left(\mathbf{r}_{2,K}^T G_K(e) \mathbf{r}_{2,K} \right)}$$

Error estimator for meshes with high aspect ratio

$$\int_{t^{n-1}}^{t^n} \frac{1}{\sqrt{\lambda_{2,K}}} \left\| \left[\frac{\partial c_h}{\partial n} \right] \right\|_{L^2(\partial K)} \sqrt{\lambda_{1,K}^2 \left(\mathbf{r}_{1,K}^T G_K(e) \mathbf{r}_{1,K} \right) + \lambda_{2,K}^2 \left(\mathbf{r}_{2,K}^T G_K(e) \mathbf{r}_{2,K} \right)}$$

$$G_K(e) = \begin{pmatrix} \int_K \left(\frac{\partial e}{\partial x_1} \right)^2 & \int_K \frac{\partial e}{\partial x_1} \frac{\partial e}{\partial x_2} \\ \int_K \frac{\partial e}{\partial x_1} \frac{\partial e}{\partial x_2} & \int_K \left(\frac{\partial e}{\partial x_2} \right)^2 \end{pmatrix}$$

Zienkiewicz-Zhu (ZZ) error estimator $\int_K \left(\frac{\partial e}{\partial x_1} \right)^2 \to \int_K \left(\frac{\partial u_h}{\partial x_1} - \Pi \frac{\partial u_h}{\partial x_1} \right)^2$

Error estimator equivalent to the true error for adapted meshes

$$C_1 \eta_K \leq \textit{true error} \leq C_2 \eta_K$$

Error estimator for meshes with high aspect ratio

$$\int_{t^{n-1}}^{t^n} \frac{1}{\sqrt{\lambda_{2,K}}} \left\| \left[\frac{\partial c_h}{\partial n} \right] \right\|_{L^2(\partial K)} \sqrt{\lambda_{1,K}^2 \left(\mathbf{r}_{1,K}^T G_K(e) \mathbf{r}_{1,K} \right) + \lambda_{2,K}^2 \left(\mathbf{r}_{2,K}^T G_K(e) \mathbf{r}_{2,K} \right)}$$

$$G_K(e) = \begin{pmatrix} \int_K \left(\frac{\partial e}{\partial x_1} \right)^2 & \int_K \frac{\partial e}{\partial x_1} \frac{\partial e}{\partial x_2} \\ \int_K \frac{\partial e}{\partial x_1} \frac{\partial e}{\partial x_2} & \int_K \left(\frac{\partial e}{\partial x_2} \right)^2 \end{pmatrix}$$

Zienkiewicz-Zhu (ZZ) error estimator $\int_K \left(\frac{\partial e}{\partial x_1} \right)^2 \rightarrow \int_K \left(\frac{\partial u_h}{\partial x_1} - \Pi \frac{\partial u_h}{\partial x_1} \right)^2$

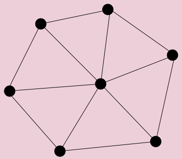
Error estimator equivalent to the true error for adapted meshes

$$C_1 \eta_K \leq true\ error \leq C_2 \eta_K$$

Error estimator for meshes with high aspect ratio

$$\int_{t^{n-1}}^{t^n} \frac{1}{\sqrt{\lambda_{2,K}}} \left\| \left[\frac{\partial c_h}{\partial n} \right] \right\|_{L^2(\partial K)} \sqrt{\lambda_{1,K}^2 \left(\mathbf{r}_{1,K}^T G_K(e) \mathbf{r}_{1,K} \right) + \lambda_{2,K}^2 \left(\mathbf{r}_{2,K}^T G_K(e) \mathbf{r}_{2,K} \right)}$$

$$G_K(e) = \begin{pmatrix} \int_K \left(\frac{\partial e}{\partial x_1} \right)^2 & \int_K \frac{\partial e}{\partial x_1} \frac{\partial e}{\partial x_2} \\ \int_K \frac{\partial e}{\partial x_1} \frac{\partial e}{\partial x_2} & \int_K \left(\frac{\partial e}{\partial x_2} \right)^2 \end{pmatrix}$$



Zienkiewicz-Zhu (ZZ) error estimator $\int_K \left(\frac{\partial e}{\partial x_1} \right)^2 \rightarrow \int_K \left(\frac{\partial u_h}{\partial x_1} - \Pi \frac{\partial u_h}{\partial x_1} \right)^2$

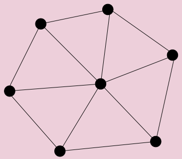
Error estimator equivalent to the true error for adapted meshes

$$C_1 \eta_K \leq \mathit{true\ error} \leq C_2 \eta_K$$

Error estimator for meshes with high aspect ratio

$$\int_{t^{n-1}}^{t^n} \frac{1}{\sqrt{\lambda_{2,K}}} \left\| \left[\frac{\partial c_h}{\partial n} \right] \right\|_{L^2(\partial K)} \sqrt{\lambda_{1,K}^2 \left(\mathbf{r}_{1,K}^T G_K(e) \mathbf{r}_{1,K} \right) + \lambda_{2,K}^2 \left(\mathbf{r}_{2,K}^T G_K(e) \mathbf{r}_{2,K} \right)}$$

$$G_K(e) = \begin{pmatrix} \int_K \left(\frac{\partial e}{\partial x_1} \right)^2 & \int_K \frac{\partial e}{\partial x_1} \frac{\partial e}{\partial x_2} \\ \int_K \frac{\partial e}{\partial x_1} \frac{\partial e}{\partial x_2} & \int_K \left(\frac{\partial e}{\partial x_2} \right)^2 \end{pmatrix}$$

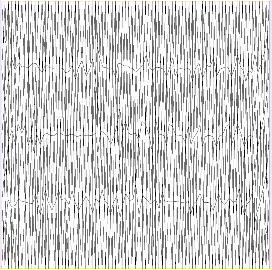


Zienkiewicz-Zhu (ZZ) error estimator $\int_K \left(\frac{\partial e}{\partial x_1} \right)^2 \rightarrow \int_K \left(\frac{\partial u_h}{\partial x_1} - \Pi \frac{\partial u_h}{\partial x_1} \right)^2$

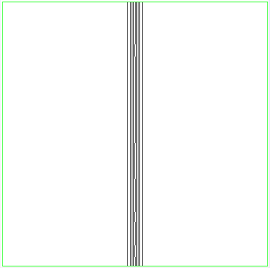
Error estimator equivalent to the true error for adapted meshes

$$C_1 \eta_K \leq \textit{true error} \leq C_2 \eta_K$$

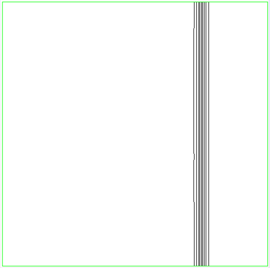
Numerical validation of the error indicator



mesh



ϕ, c at initial time



ϕ, c at final time

$h1 - h2$	error	ei
0.000005 – 0.0001	0.29	1.85
0.0000025 – 0.00005	0.13	1.74
0.00000125 – 0.000025	0.066	1.74

ei: effectivity index, error indicator / true error.

Adaptive algorithm

Goal of the adaptive algorithm: build a sequence of triangulations such that

$$0.75\ TOL \leq \frac{\left(\sum_K \eta_K^2\right)^{1/2}}{\left(\int_\Omega |\nabla u_h|^2\right)^{1/2}} \leq 1.25\ TOL$$

Sufficient condition : $0.75^2\, TOL^2 \int_K |\nabla u_h|^2 \leq \eta_K^2 \leq 1.25^2\, TOL^2 \int_K |\nabla u_h|^2$

Equidistribute the error in directions 1 and 2

Align the triangle with the eigenvectors of $G_K(e)$

Use the BL2D mesh generator (INRIA)
(Meshadapt software (INRIA) in 3D)

Adaptive algorithm

Goal of the adaptive algorithm: build a sequence of triangulations such that

$$0.75\ TOL \leq \frac{\left(\sum_K \eta_K^2\right)^{1/2}}{\left(\int_\Omega |\nabla u_h|^2\right)^{1/2}} \leq 1.25\ TOL$$

Sufficient condition : $0.75^2\, TOL^2 \int_K |\nabla u_h|^2 \leq \eta_K^2 \leq 1.25^2\, TOL^2 \int_K |\nabla u_h|^2$

Equidistribute the error in directions 1 and 2

Align the triangle with the eigenvectors of $G_K(e)$

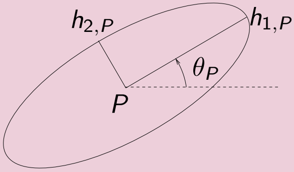
Use the BL2D mesh generator (INRIA)
(Meshadapt software (INRIA) in 3D)

Adaptive algorithm

Goal of the adaptive algorithm: build a sequence of triangulations such that

$$0.75\ TOL \leq \frac{\left(\sum_K \eta_K^2\right)^{1/2}}{\left(\int_\Omega |\nabla u_h|^2\right)^{1/2}} \leq 1.25\ TOL$$

Sufficient condition : $0.75^2\, TOL^2 \int_K |\nabla u_h|^2 \leq \eta_K^2 \leq 1.25^2\, TOL^2 \int_K |\nabla u_h|^2$



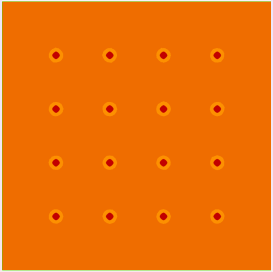
Equidistribute the error in directions 1 and 2

Align the triangle with the eigenvectors of $G_K(e)$

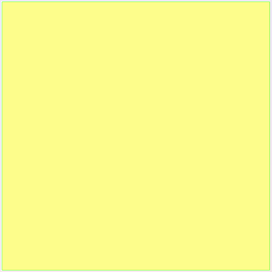
Use the BL2D mesh generator (INRIA)
(Meshadapt software (INRIA) in 3D)

Numerical results in 2D : 16 dendrites

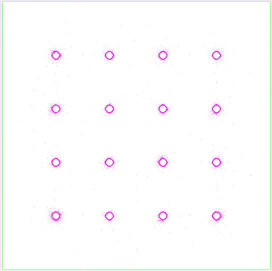
Time 0.025 s



concentration



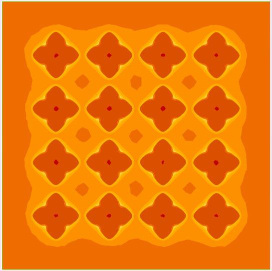
pressure



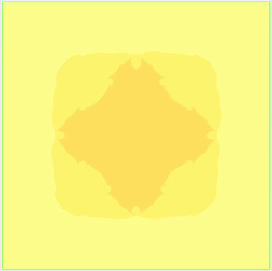
velocity

Numerical results in 2D : 16 dendrites

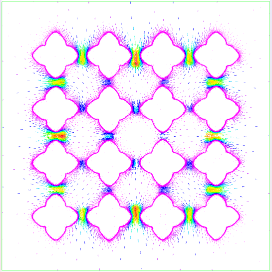
Time 0.15 s



concentration



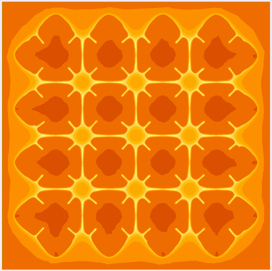
pressure

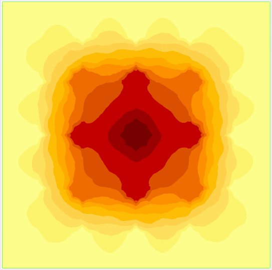


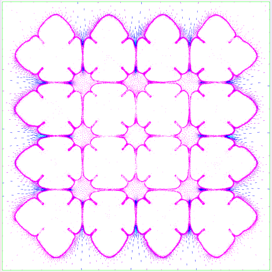
velocity

Numerical results in 2D : 16 dendrites

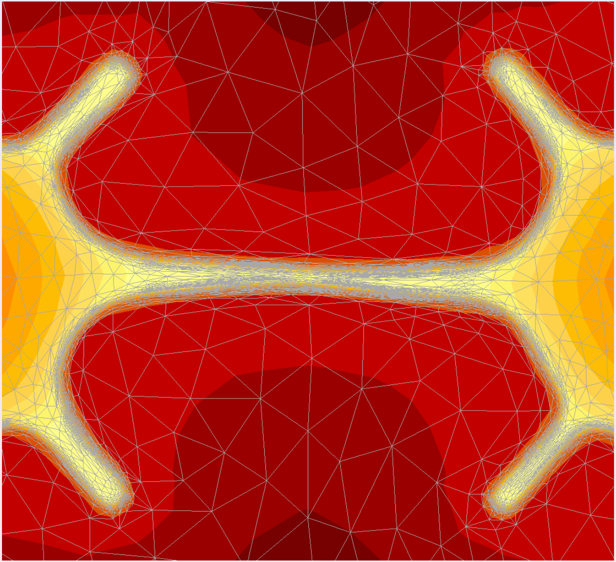
Time 0.225 s


concentration

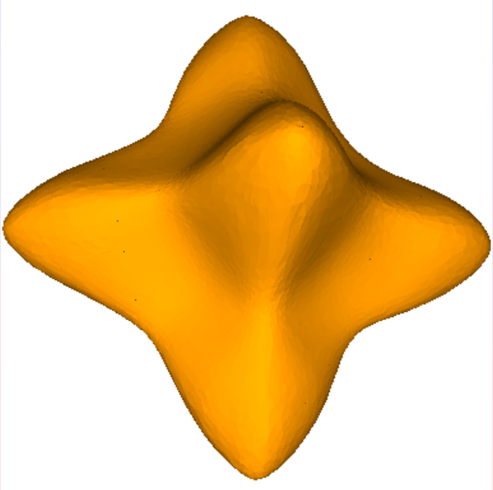

pressure


velocity

Numerical results in 2D : 16 dendrites

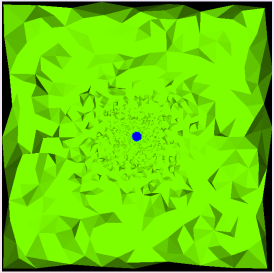


Numerical results in 3D : 1 dendrite

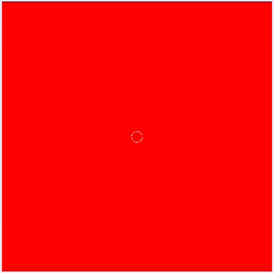


Numerical results in 3D : 1 dendrite

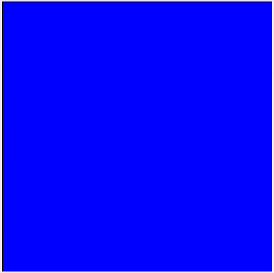
Time 0.25 s



concentration



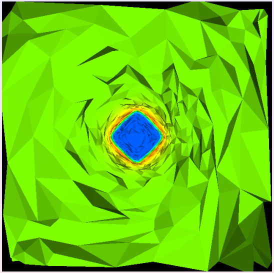
pressure



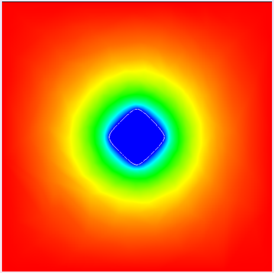
velocity

Numerical results in 3D : 1 dendrite

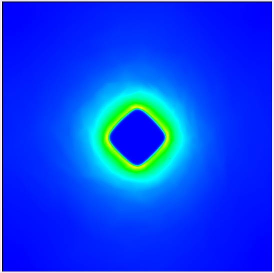
Time $0.50\ s$



concentration



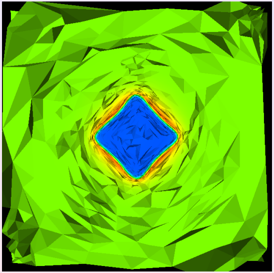
pressure



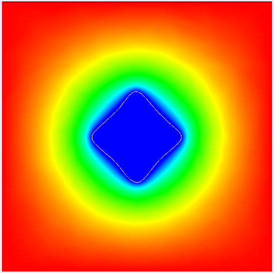
velocity

Numerical results in 3D : 1 dendrite

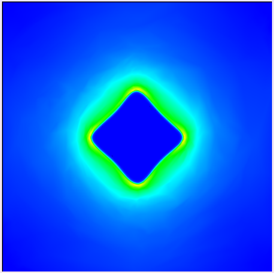
Time 0.50 s



concentration



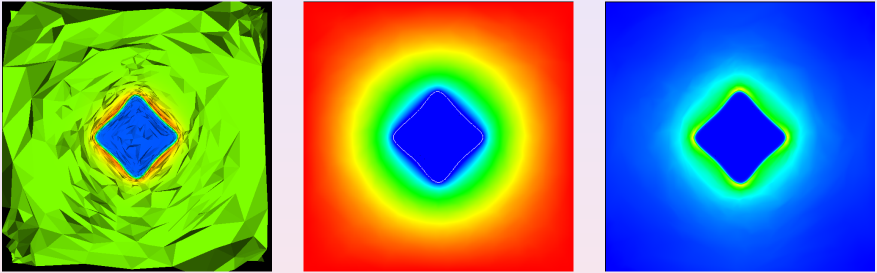
pressure



velocity

Numerical results in 3D : 1 dendrite

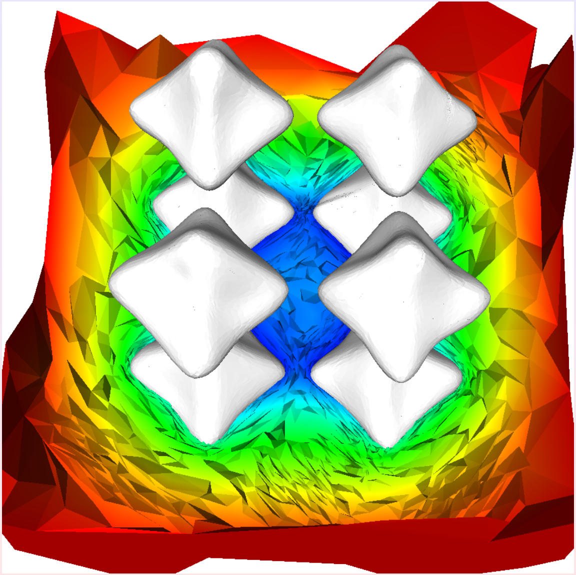
Time 0.50 s



concentration pressure velocity

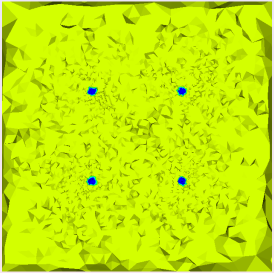
Maximal number of vertices : 138487
Average aspect ratio : 77.68

Numerical results in 3D : 8 dendrites

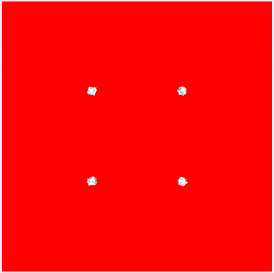


Numerical results in 3D : 8 dendrites

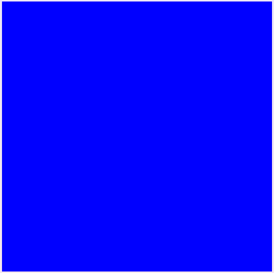
Time 0.00 s



concentration



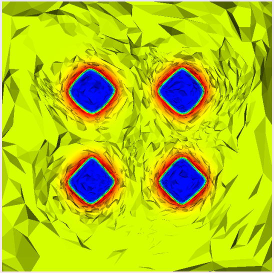
pressure



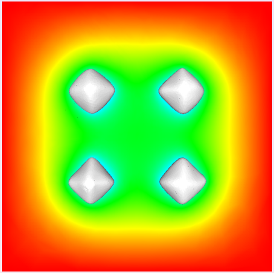
velocity

Numerical results in 3D : 8 dendrites

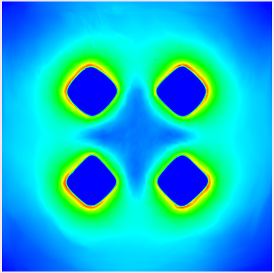
Time 0.25 s



concentration



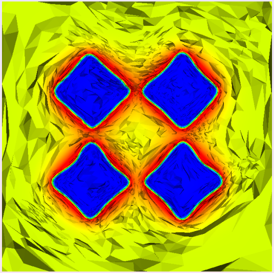
pressure



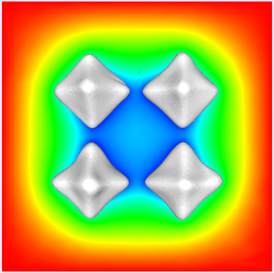
velocity

Numerical results in 3D : 8 dendrites

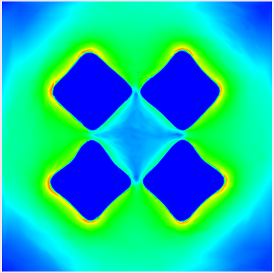
Time 0.50 *s*



concentration



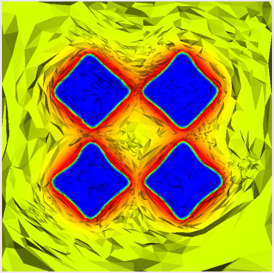
pressure



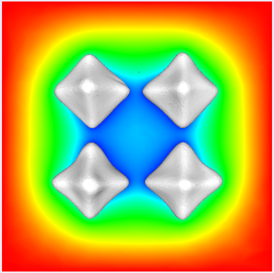
velocity

Numerical results in 3D : 8 dendrites

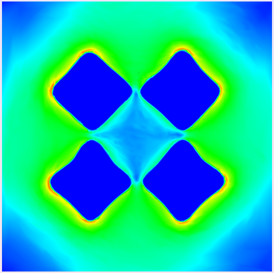
Time 0.50 s



concentration



pressure



velocity

Maximal number of vertices : 256263
Average aspect ratio : 11.51

Conclusions and perspectives

- Anisotropic, adaptive meshes make 3D simulations possible on a single PC.
- Theoretical justification ? Use of several error estimators ?
- Micro - macro coupling ?