Towards complex and realistic tokamaks geometries in Computational Plasma Physics

A. Ratnani[1]
G. Dif-Pradalier[2], V. Grandgirard[2], M. Bécoulet[2], B. Nkonga[1], G. Huysmans[3], E. Sonnendrücker[4]

[1] Université de Nice, France
[2] CEA, IRFM, Cadarache, France
[3] ITER Organization, Cadarache, France
[4] IPP, Garching, Germany

14/08/2014

Summary

Financial Support : ANR-11-MONU-002 **ANEMOS** & **EFDA** (European Fusion Development Agreement) - **Fellowship**

- **Act I — Introduction.**
  - Motivations
  - B-splines, NURBS, curves and surfaces
  - The IGA approach
  - Applications
  - Impact of the k-refinement strategy
- **Act II — Adaptive meshes :r-refinement.**
  - Alignement
  - Iterative method using a Posteriori Estimates
  - Equidistribution
- **Act III — Past, present and future.**
  - The **Πgasus** suite
  - Ongoing and forseen work
  - Conclusions.

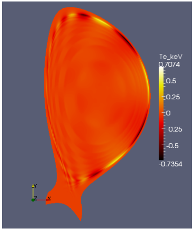
Forewords: ELMs —the need for realistic plasma conditions

Edge Localised Modes (ELMs) $\equiv$ MHD instabilities destabilised by the pressure & current gradients in the H-mode edge pedestal

- losses up to 10% plasma energy in $\sim$ 0.1–1 ms
  - major concern for the operation of ITER
- ELM control is essential in ITER
  - requires physics understanding of ELMs
  - requires ELM nonlinear MHD simulations
- challenging in Iter: $\nu_\star$, resistivity η, transp. anisotropy $\chi_\parallel/\chi_\perp$, size & shape...

What applications were in mind when writing JOREK ?

- **ELM cycle & control**
  - edge current (with bootstrap) ➠ drives peeling/kink mode
  - edge pressure gradient ➠ drives ballooning mode
  - exact geometry [plasma shape, divertor, resistive wall, coils, vacuum, ...]
  - [• RMPs, vertical kicks, pellet injection]

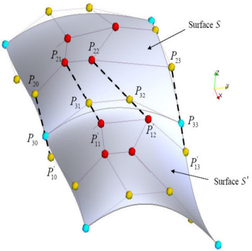


- **Disruptions, other...**
  - vertical displacement events, beta limit disruptions, density limit

Forewords: JOREK—Overview

JOREK $\equiv$ nonlinear reduced MHD in realistic toroidal geom. for ELM simulations

- **closed & open field lines domain**, **X-point geom.**
  - Cubic Finite Elements, flux aligned poloidal grid
  - Isoparametric : elements approaching geometry are used to approach unknowns
  - Fourier series in toroidal direction
  - Non-linear reduced MHD in toroidal geometry
- **time stepping, solver & parallelism**
  - fully implicit Crank-Nicholson
  - sparse matrices (PastiX): $\sim 10^7$ degrees of freedom
  - MPI/OpenMP over typically $256 - 1500$ processors
- **ELM simulations consumptions**
  - At **IRFM**, we use 7 Millions CPUH/year
  - Typical simulations : $\sim 20'000 - 200'000$ CPUH
  - A JET simulation ($n_{tor} = 21$) : $\sim 100'000 - 200'000$



Forewords: JOREK—Limitations and challenge

Some limitations and challenges

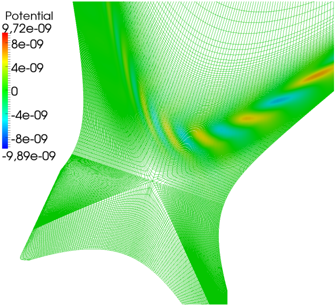
About the Geometry

- Grid Generation ➠ Automatic fields aligned Grid construction
- Exact geometry & boundary conditions ➠ Needs other tools from **CAD** : NURBS, T-Splines, ...

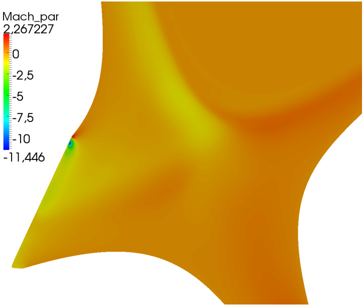
About the Numerical Scheme

- Memory limitations for realistic ITER simulations ➠
  - Use High-Order Methods,
  - Reduce the size of discrete matrices,
  - Work on structured meshes.
- Non-linear MHD in toroidal geometry over long time scales ($\mu s \rightarrow s$) ➠ Use High-regularity Elements
- Linear solvers ➠ Derive fast solvers, preconditionners

Current grid problems In Jorek



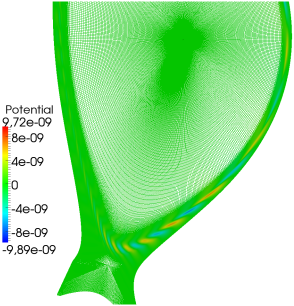
(a)



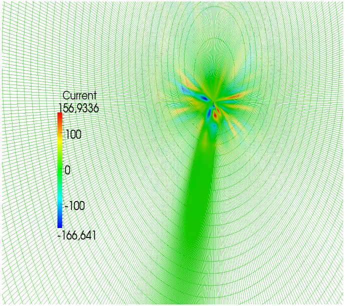
(b)

FIGURE: (a) Large number of flux surfaces on one side and the other of the separatrix. (b) A very fine grid next to the plasma center is both a waste in numerical ressources through increased memory consumption and increased computing time and a potential source for numerical instabilities.

Current grid problems In Jorek



(c)

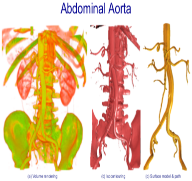


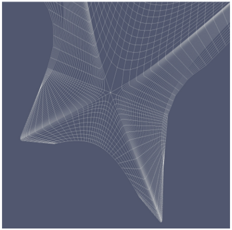
(d)

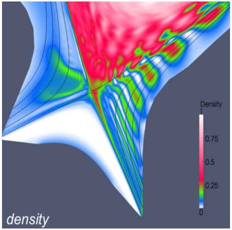
FIGURE: (c) Mesh accumulation next to the plasma center. (d) Flow instabilities due to angular points.

From CAD to Numerical Simulation

- In Computer Aided Design (**CAD**), many interesting tools (including fast algorithms) have been developped,
- The 3D animation Industry (including Medical one), is a huge one compared to Numerical Simulation's one,
- Hughes et al. made the link between the two communities with the introduction of the **IsoGeometric Analysis** approach.







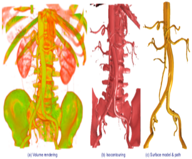
JOREK

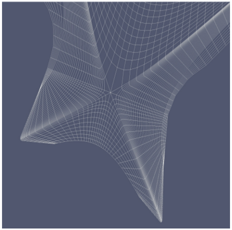
From CAD to Numerical Simulation

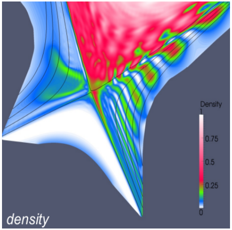
- In Computer Aided Design (**CAD**), many interesting tools (including fast algorithms) have been developpped,
- The 3D animation Industry (including Medical one), is a huge one compared to Numerical Simulation's one,
- Hughes et al. made the link between the two communities with the introduction of the **IsoGeometric Analysis** approach.

➠ Functions describing the exact geometry are used to approach unknowns,
➠ IsoGeometric Analysis contains the IsoParametric approach,
➠ Emergence of the k-refinement strategy in addition to hp-refinement

Abdominal Aorta



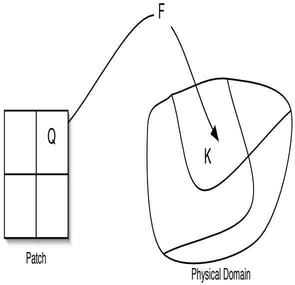




JOREK

CAD tools for complex geometries and adaptive grids

– The current Grid construction is difficult
  ➠ local treatment of each element in order to ensure the $\mathcal{C}^1$ or $\mathcal{G}^1$ contraints,
– The aligned grid are of no great interest during the nonlinear stage
  ➠ use grids that minimize a *specific error*,
+ Need to create a global mapping of arbitrary regularity,
+ The construct mapping must verify an alignement and/or equidistribution property,
+ Use CAD tools to model complex geometries,



Splines Curves

- A family of N B-splines of order k (and degree $k-1$) can be generated using a non-decreasing sequence of knots $T = (t_i)_{1 \leqslant i \leqslant N+k}$.
- Let $(P_i)_{1 \leqslant i \leqslant N} \in \mathbb{R}^d$ be a sequence of control points, forming a control polygon.
- A **B-spline curve** is defined as the following parametric curve

$$\mathbf{M}(t) = \sum_{i=1}^{N} N_i^k(t)\mathbf{P}_i$$

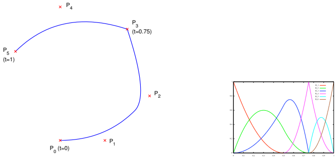


FIGURE: A B-spline curve and its control points, $T = \{000\ \frac{1}{2}\ \ \frac{3}{4}\ \frac{3}{4}\ 111\}$.

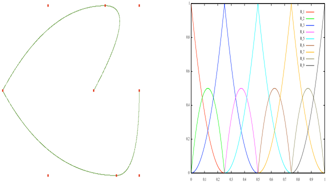


FIGURE: (left) A B-spline curve and its control points, (right) B-splines functions used to draw the curve. $N = 9$, $p = 2$, $T = \{000, \frac{1}{4}\frac{1}{4}, \frac{1}{2}\frac{1}{2}, \frac{3}{4}\frac{3}{4}, 111\}$

Sticking patchs

- Deriving a B-spline curve $\mathcal{C}(t) = \sum_{i=1}^{n} N_i^k(t)\mathbf{P}_i$ gives

$$\mathcal{C}'(t) = \sum_{i=1}^{n-1} N_i^{k-1*}(t)\mathbf{Q}_i \quad \text{where} \quad \mathbf{Q}_i = p\frac{\mathbf{P}_{i+1} - \mathbf{P}_i}{t_{i+1+p} - t_{i+1}} \tag{1}$$

  $\{N_i^{k-1*}, \quad 1 \leqslant i \leqslant n-1\}$ are generated using the knot vector T^* which is obtained from T by removing the first and the last knot.

- $\mathcal{C}'(0) = \frac{p}{t_{p+2}}(\mathbf{P}_2 - \mathbf{P}_1)$,
- $\mathcal{C}'(1) = \frac{p}{1-t_n}(\mathbf{P}_n - \mathbf{P}_{n-1})$,
- $\mathcal{C}''(0) = \frac{p(p-1)}{t_{p+2}}\left(\frac{1}{t_{p+2}}\mathbf{P}_1 - \{\frac{1}{t_{p+2}} + \frac{1}{t_{p+3}}\}\mathbf{P}_2 + \frac{1}{t_{p+3}}\mathbf{P}_3\right)$,
- $\mathcal{C}''(1) = \frac{p(p-1)}{1-t_n}\left(\frac{1}{1-t_n}\mathbf{P}_n - \{\frac{1}{1-t_n} + \frac{1}{1-t_{n-1}}\}\mathbf{P}_{n-1} + \frac{1}{1-t_{n-1}}\mathbf{P}_{n-2}\right)$.

In the case of $\mathcal{C}^1$-sticking two patchs, namely $\mathcal{P}_m$ and $\mathcal{P}_s$, on the faces f_m and f_s respectively.

$$\mathbf{c}_{:,f_m} = \mathbf{c}_{:,f_s} \quad \text{and} \quad \alpha_m\left(\mathbf{c}_{:,f_m} - \mathbf{c}_{:,\delta f_m}\right) = \beta_s\left(\mathbf{c}_{:,f_s} - \mathbf{c}_{:,\delta f_s}\right)$$

Towards the IGA

- **Fundamental geometric operations**
  - **Knot insertion :** DeBoor algorithm, Blossoming algorithms,...
  - **Order elevation :** Prautzsch, Piegl, Huang,...
- **Multivariate tensor product splines**
  - Let us consider d knot vectors $\mathcal{T} = \{T^1, T^2, \cdots, T^d\}$.
  - The basis for $\mathcal{S}_{\mathbf{k}}(\mathcal{T})$ is defined by a tensor product $N_{\mathbf{i}}^{\mathbf{k}} := N_{i_1}^{k_1} \otimes N_{i_2}^{k_2} \otimes \cdots \otimes N_{i_d}^{k_d}$

Refinement strategies

Refining the grid can be done in 3 different ways. This is the most interesting aspects of B-splines basis.

h-refinement by inserting new knots. It is the equivalent of mesh refinement of the classical finite element method.

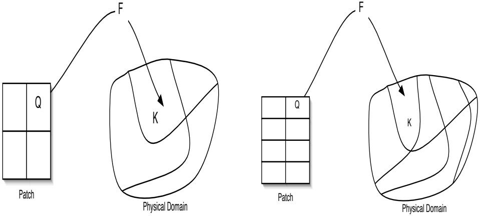
p-refinement by elevating the B-spline degree. It is the equivalent of using higher finite element order in the classical FEM.

k-refinement by increasing / decreasing the regularity of the basis functions (increasing / decreasing multiplicity of inserted knots).

the use of **k-refinement** strategy is more efficient than the classical p-refinement, as it reduces the dimension of the basis.

Towards the IGA

- The first step, is to define the patch, *i.e* the parametric domain, and the mapping **F**, that maps the patch into the physical domain.
- The minimal degree of the basis functions is imposed by the domain design.
- **Grid generation :** the use of $h/p/k$-refinement keeps the mapping **F** unchanged



Applications

We have used the IGA approach to solve many problems :

- Quasi-Neutral Equation,
- MHD Equilibrium,
- Linear and non-linear elliptic pde's
- Maxwell's equations,
- Semi-Lagrangian schemes,
- PIC method,
- Reduced MHD,
- Mesh generation using aligned magnetic fields and anisotropic adaptive mappings,
- Geometric Multigrid Solvers,

k-refinement, a blessed strategy ?

In the following table, we show the impact of the k-refinement on the resolution of the Poisson's equation, on a square domain using :

- B-splines of degree p and minimal regularity (*i.e.* $\mathcal{C}^0$)
- B-splines of degree p and maximal regularity (*i.e.* $\mathcal{C}^{p-1}$)

	number of d.o.f		number of nnz		cpu time-SuperLU		cpu-CG	
	$\mathcal{C}^{p-1}$	$\mathcal{C}^0$	$\mathcal{C}^{p-1}$	$\mathcal{C}^0$	$\mathcal{C}^{p-1}$	$\mathcal{C}^0$	$\mathcal{C}^{p-1}$	$\mathcal{C}^0$
p=2	4'096	16'129	98'596	253'009	0.23	0.35	$7\ 10^{-4}$	$4.1\ 10^{-3}$
p=3	4'225	36'481	196'249	896'809	0.61	1.64	$1.1\ 10^{-2}$	$2\ 10^{-2}$
p=5	4'489	101'761	499'849	4'923'961	2.96	49.27	$3.8\ 10^{-2}$	$3.7\ 10^{-1}$

TABLE: Impact of the k-refinement on the resolution of the Poisson equation on a grid 64×64 for quadratic, cubic and quintic B-splines.

k-refinement, a blessed strategy ?

In the following table, we show the impact of the k-refinement on the resolution of the Poisson's equation, on a square domain using :

- B-splines of degree p and minimal regularity (*i.e.* $\mathcal{C}^0$)
- B-splines of degree p and maximal regularity (*i.e.* $\mathcal{C}^{p-1}$)

	number of d.o.f		number of nnz		cpu time-SuperLU		cpu-CG	
	$\mathcal{C}^{p-1}$	$\mathcal{C}^0$	$\mathcal{C}^{p-1}$	$\mathcal{C}^0$	$\mathcal{C}^{p-1}$	$\mathcal{C}^0$	$\mathcal{C}^{p-1}$	$\mathcal{C}^0$
p=2	4'096	16'129	98'596	253'009	0.23	0.35	$7\ 10^{-4}$	$4.1\ 10^{-3}$
p=3	4'225	36'481	196'249	896'809	0.61	1.64	$1.1\ 10^{-2}$	$2\ 10^{-2}$
p=5	4'489	101'761	499'849	4'923'961	2.96	49.27	$3.8\ 10^{-2}$	$3.7\ 10^{-1}$

TABLE: Impact of the k-refinement on the resolution of the Poisson equation on a grid 64×64 for quadratic, cubic and quintic B-splines.

➠ Further Mermory consumptions must be done Coming soon

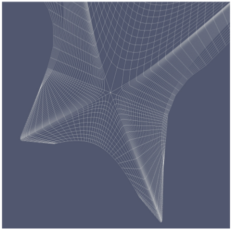
Summary

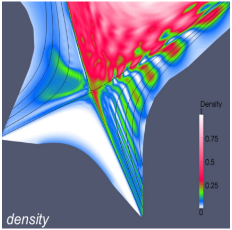
- **Act I — Introduction.**
  - Motivations
  - B-splines, NURBS, curves and surfaces
  - The IGA approach
  - Impact of the k-refinement strategy
- **Act II — Adaptive meshes :r-refinement.**
  - Alignement
  - Iterative method using a Posteriori Estimates
  - Equidistribution
- **Act III — Past, present and future.**
  - The **Πgasus** suite
  - Ongoing and forseen work
  - Conclusions.

Adaptive meshes : The r-refinement strategy — Basic ideas

For ELMs simulations, we want to

1. start with a flux aligned grid,
2. use this grids until the nonlinear stage,
3. adapt the grid in order to follow a given estimator, the density or current, . . .





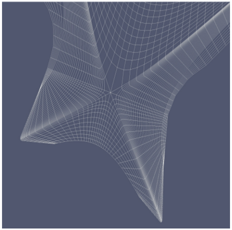
Idea : Move the control points in order to have a better resolution in high density regions. ➠ needs to define a strategy

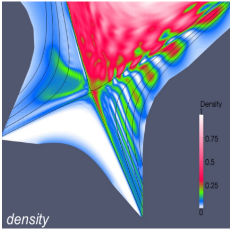
- Iteratively, by minimizing an estimation of the numerical error (difficult for MHD).
- To ensure an equi-distribution property : needs a *monitor* function (works also with a posteriori-estimates).

Depending on the method, specific conditions (on the boundary) must be hold in order to keep the exact geometry.

For ELMs simulations, we want to

1. start with a flux aligned grid,
2. use this grids until the nonlinear stage,
3. adapt the grid in order to follow a given estimator, the density or current, ...





Idea : Move the control points in order to have a better resolution in high density regions. ➠ needs to define a strategy

- Iteratively, by minimizing an estimation of the numerical error (difficult for MHD).
- To ensure an equi-distribution property : needs a *monitor* function (works also with a posteriori-estimates).

Depending on the method, specific conditions (on the boundary) must be hold in order to keep the exact geometry.

➠ **Injectivity property** : the geometric transformation **F** must be a one-to-one map

Adaptive meshes — Aligned meshes problem

INPUT $\mathbf{V}_k$ given points,

ASSUMPTIONS Fix a B-splines family and a parametrization $\mathbf{u}_k$,

- Construct the mapping $\mathbf{F}$ by minimizing an objective function :

$$\mathcal{J}_0[\mathbf{F}] + \lambda \mathcal{J}_1[\mathbf{F}] \tag{2}$$

where,

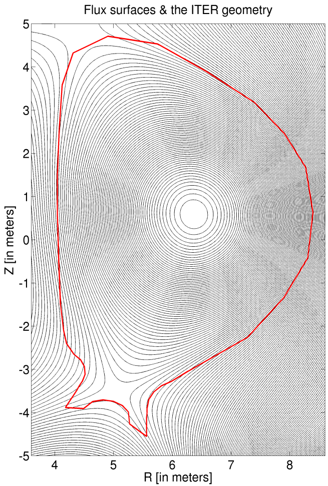
$$\mathcal{J}_0[\mathbf{F}] = \sum_{k=1}^{N} \|\mathbf{F}(\mathbf{u}_k) - \mathbf{V}_k\|^2 \tag{3}$$

λ is a given parameter.

- $\mathcal{J}_1$ is an approximation of the curvature,
- this kind of problem is known as an optimization with fair,

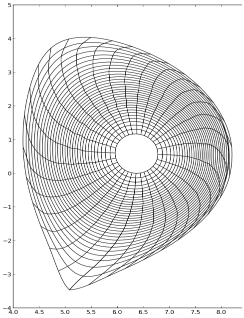
Does not preserve the injectivity !

Needs to define some parameters !

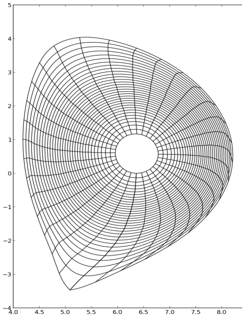


Aligned meshes — Impact of the parameters

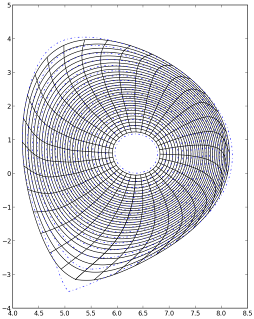
- parametrization impacts the behavior in the radius direction ((a) vs (b))
- optimal values for λ in (a,b),
- bad value for λ in (c) : forcing more smoothness impact the X-point,



(a)



(b)



(c)

Adaptive meshes — Aligned meshes problem for ITER

- The ITER wall + the aligned mapping F (defined on 6 patchs+connectivity)
- Split B-splines surface using specific algorithms,
- The final mapping can be refined if needed (while keeping exact geometry),

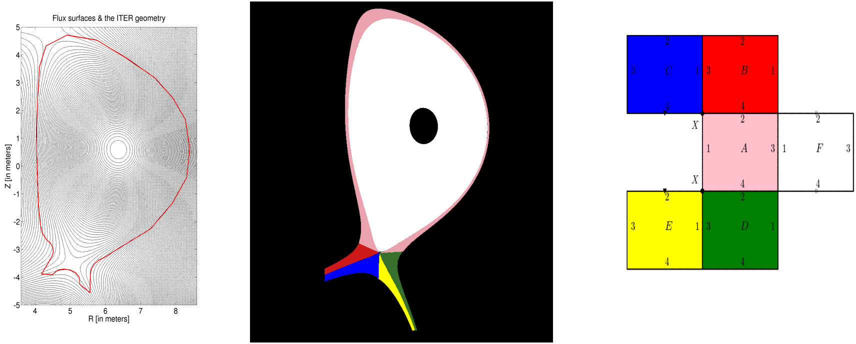
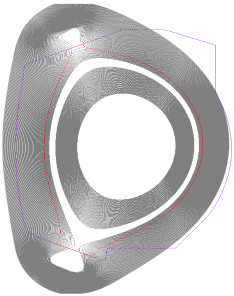
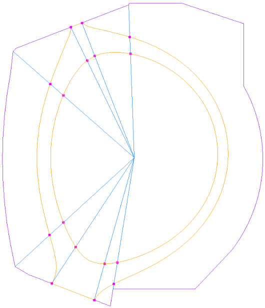


FIGURE: Generating Aligned meshes for ITER

➠ Automatic search for the last magnetic surface, using **k-means** algorithm.



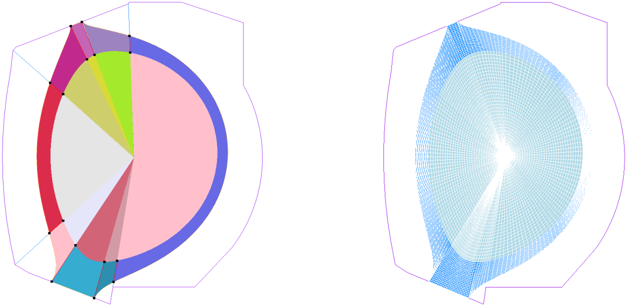
(a)



(b)

FIGURE: (a) Example of an optimal flux surface that intersects the wall 4 times. (b) Limiters are constructed using the center of the plasma. They split the whole domain into several sub-domains.

➠ use Fields aligned strategy on Internal Patchs and Adaptive (Equidistributed) mappings on External Patchs.



(c) (d)

FIGURE: (c) A 2D description is constructed using the given boundaries and the coons algorithm. (d) The alignement optimization procedure is used to construct a polar-like grid.

Adaptive meshes : The r-refinement strategy — Iterative method using a Posteriori Estimates [Xu2011]

Let us consider the following elliptic problem,

$$-\nabla \cdot (A \nabla \psi) = f, \Omega \qquad\qquad u = 0, \partial\Omega \tag{4}$$

Let ψ_h be the numerical solution and $e_h = \psi - \psi_h$ the error of the IsoGeometric approximation.
In the case of $\mathcal{C}^1$ elements, we have the following *a posteriori* estimation :

$$\|e_h\|^2 \lesssim \sum_{Q \in \mathcal{Q}_h} h_Q^2 \|f + \nabla^2 \psi_h\|_{L^2(Q)}^2 \tag{5}$$

The idea is to move the control points, in order to minimize the derived a posteriori estimation.
The algorithm can be summerized as follows :

1. Evaluation of the error for perturbed control points,
2. Estimation of the gradient $\nabla e(\mathbf{c})$,
3. Defining the search direction $\mathbf{d} = -\nabla e(\mathbf{c})$,
4. Line search : find ρ such that $e(\mathbf{c} + \rho\mathbf{d}) < e(\mathbf{c})$.

In order to reduce the computational cost, we start by removing some interior knots (while keeping exact the mapping).

As an example, we solve the Poisson problem on a square domain.

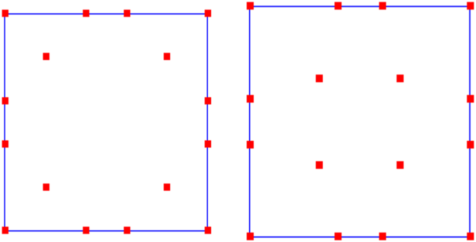


FIGURE: r-refinement : (left) the initial description of the square, (right) the final geometry after optimization.

Does not preserve the injectivity ! ➠ needs to add non-linear constraints.

Adaptive meshes — Optimal transport

Definition (L^2 Monge-Kantorovich problem)

Let ρ_0 and ρ_1 be two given densities of equal masses, defined in $\Omega \subset \mathbb{R}^d$.
Find a mapping $\mathbf{x}' = \mathbf{\Psi}(\mathbf{x})$, $\mathbf{x}, \mathbf{x}' \in \Omega$, that transfers the density ρ_0 to ρ_1 and minimizes the transport cost

$$\mathcal{J}[\mathbf{\Psi}] = \int_\Omega |\mathbf{\Psi}(\mathbf{x}) - \mathbf{x}|^2 \rho_0(\mathbf{x}) \ d\mathbf{x} \tag{6}$$

The density transfert means that, $\forall \omega \subset \Omega$,

$$\int_{\mathbf{\Psi}^{-1}(\omega)} \rho_0(\mathbf{x}) \ d\mathbf{x} = \int_\omega \rho_1(\mathbf{x}') \ d\mathbf{x}' \tag{7}$$

If the mapping $\mathbf{\Psi}$ is a smooth one-to-one map, then (Eq. 7) writes

$$\rho_1(\mathbf{\Psi}(\mathbf{x})) \det\left(\frac{\partial \mathbf{\Psi}}{\partial \mathbf{x}}\right) = \rho_0(\mathbf{x}) \tag{8}$$

Existence theorem [Brenier'91]

- There exists a unique optimal mapping that satisfies the equidistibution principle.
- This mapping can be written as the gradient of a **convex** function ϕ
- ϕ is the solution of the Monge-Ampère eq.

$$\det H(\phi) = \frac{\sigma}{|\Omega_c|\rho(\nabla_{\boldsymbol{\xi}}\phi)} \tag{9}$$

for the boundary conditions, we ensure that $\mathbf{x}(\boldsymbol{\xi})$ maps $\partial\Omega_c$ to $\partial\Omega$:

$$\nabla_{\boldsymbol{\xi}}\phi(\partial\Omega_c) = \partial\Omega \tag{10}$$

- **Dean [2003]** using an augmented Lagrangian approach,
- **Dean [2004,2006]** a least square approach,
- **Loeper [2005]** iterative solver based on the divergence form and Newton,
- **Delzanno [2008]** the authors used an iterative Newton-Krylov solver with preconditioning coupled with a Finite Difference method,
- **Awanou [2012,2013]** using singular perturbation, triangular B-splines (treate complex geometries),
- **Budd and Williams [2006]** consider the solution as the steady state of a parabolic equation namely the Parabolic Monge-Ampère equation (PMA) :

$$\tau(I - \frac{1}{\beta^2}\nabla_{\xi}^2)\phi_t = \left(\det H(\phi)\frac{|\Omega_c|\rho(\nabla_{\xi}\phi)}{\sigma}\right)^{\frac{1}{d}} \tag{11}$$

- **Sulman [2011]** used the following parabolic Monge-Ampère problem,

$$\phi_t = \log\left(\det H(\phi)\frac{|\Omega_c|\rho(\nabla_{\xi}\phi)}{\sigma}\right) \tag{12}$$

- We propose to use a method developed by Benamou-Froese-Oberman, using the IGA approach.

Adaptive meshes — Monge-Ampère equation using Benamou-Froese-Oberman Method (BFO)

Let us define the operator $T : H^2(\Omega) \to H^2(\Omega)$ by

$$T[u] = \left(\nabla^2\right)^{-1} \sqrt{(\nabla^2 u)^2 + 2(f - \det H(u))} \tag{13}$$

It is proved in **[Benamou2010]** that when $u \in H^2(\Omega)$ is a solution of the Monge-Ampère equation, then it is a fixed point of the operator T.

$$u = T[u] \tag{14}$$

In **[Benamou2010]**, Picard's method is used with a Finite Difference method.

Algorithm

- Given an initial value u^0,
- Compute u^{n+1} as the solution of $\nabla^2 u^{n+1} = \sqrt{(\nabla^2 u^n)^2 + 2(f - \det H(u^n))}$

The BFO method presents some interesting advantages :

- It needs only to invert the Laplacian at each iteration,
- Direct solvers can be used,
- For some geometries (square, periodic domains, ...) we have fast solvers based on the Kronecker tensor product,

Inserting a new knot t, where $t_j \leqslant t < t_{j+1}$, using the DeBoor algorithm leads to a new description with :

$$\widetilde{N} = N + 1, \quad \widetilde{k} = k, \quad \widetilde{T} = \{t_1, .., t_j, t, t_{j+1}, .., t_{N+k}\}$$

$$\alpha_i = \begin{cases} 1 & 1 \leqslant i \leqslant j - k + 1 \\ \frac{t - t_i}{t_{i+k-1} - t_i} & j - k + 2 \leqslant i \leqslant j \\ 0 & j + 1 \leqslant i \end{cases}$$

$$\mathbf{Q}_i = \alpha_i \mathbf{P}_i + (1 - \alpha_i)\mathbf{P}_{i-1}$$

This can be written in the matrix form as

$$\mathbf{Q} = A\mathbf{P}$$

The basis transformation A is called *the knot insertion matrix* of degree $k-1$ from T to $\widetilde{T}$.

Now let us consider a nested sequence of knot vectors $T_0 \subset T_1 \subset \ldots \subset T_n$, where $\#(T_{i+1} - T_i) = 1$. The knot insertion matrix from T_i to T_{i+1} is denoted by A_i^{i+1}. It is easy to see that the insertion matrix from T_0 to T_n is simply :

$$A := A_0^n = A_0^1 \; A_1^2 \; \ldots \; A_{n-1}^n$$

In the case of $2D$, the interpolation operator can be constructed using the Kronecker product, as follows

$$A_\eta \otimes A_\xi$$

We consider the elliptic problem :

$$-\nabla \cdot (A_\epsilon \nabla u_\epsilon) = f \qquad (15)$$

with Dirichlet boundary conditions. The matrix A_ϵ is of the form :

$$A_\epsilon = \epsilon \mathbb{I} + (1-\epsilon)\mathbf{b} \otimes \mathbf{b} \qquad (16)$$

where $\mathbf{b} = \frac{\mathbf{B}}{\|\mathbf{B}\|}$ and $\epsilon << 1$ is the characteristic parameter for the anisotrpy problem.
In the following tests, we consider $\mathbf{b} = (\cos(\phi), \sin(\phi))$, $\phi = \frac{2\pi}{3}$ and $\epsilon = 10^{-3}, 10^{-6}, 10^{-9}$. The physical domain is either a unit square or an X-point-like domain, as shown in fig 9.

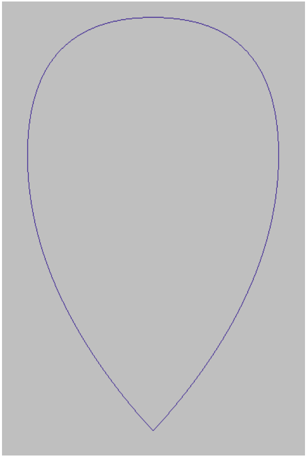


FIGURE: X-point domain.

In figures 10 and 11, we plot the history of residuals until convergence for the geometric multigrid and the accelerated *gmres* preconditioned with our geometric multigrid.

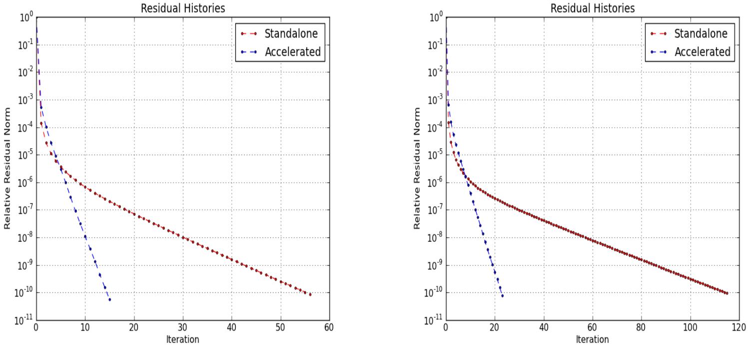


FIGURE: Square domain : Convergence of the Multigrid and accelerated GMRES for $\epsilon = 10^{-3}, 10^{-6}$ on a 128x128 grid.

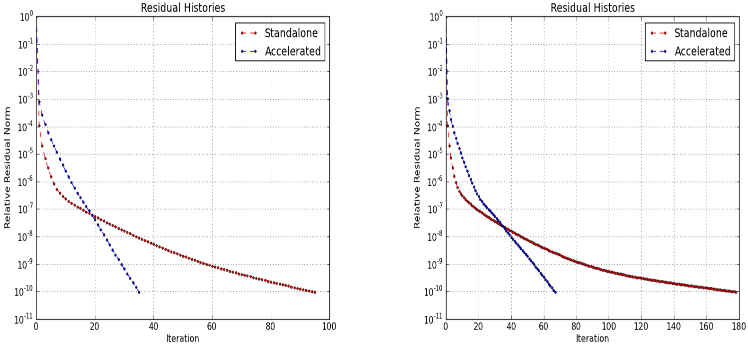
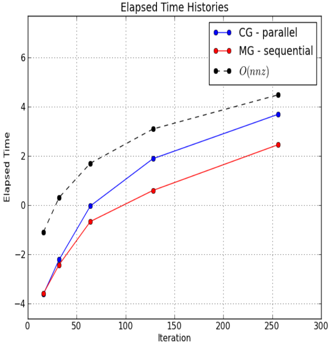


FIGURE: X-point domain : Convergence of the Multigrid and accelerated GMRES for $\epsilon = 10^{-3}, 10^{-6}$ on a 128x128 grid.

A Geometric Multigrid for IGA — Numerical results

The current tests were done using a parallel version of *Petsc* preconditioned with algebraic multigrid (AMG) and the implemented sequential version in *pigasus*. In figures 12 and 13, we plot the elapsed time before convergence.



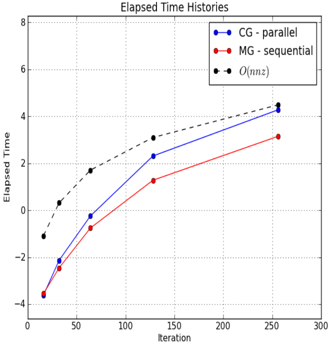


FIGURE: Square domain : elapsed time for $\epsilon = 10^{-3}, 10^{-6}$

A Geometric Multigrid for IGA — Numerical results

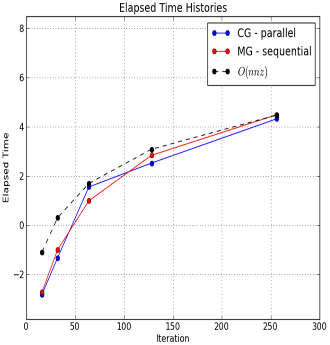


FIGURE: X-point domain : elapsed time for $\epsilon = 10^{-3}, 10^{-6}$

In order to solve a nonlinear PDE, in our case the Monge-Ampère equation, we start by :

- solve Monge-Ampère equation on a coarse grid,
- use h-refinement, iterativelly or by matrix multiplication, to transport the field onto the finest grid,
- use the latter field as an initial guess for the nonlinear PDE.

The physical domain is the unit square. The analytical solution is $u(x, y) = \exp\left(\frac{x^2+y^2}{2}\right)$. For which the source term is $f(x, y) = (1 + x^2 + y^2) \exp\left(x^2 + y^2\right)$.

quadratic	2.05	2.01	2.003
cubic	3.88	3.94	3.97
quartic	4.88	4.96	4.98

TABLE: Test : Convergence orders for the BFO method using Picard's algorithm.

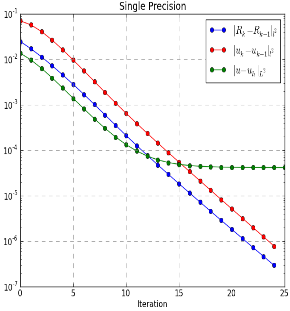
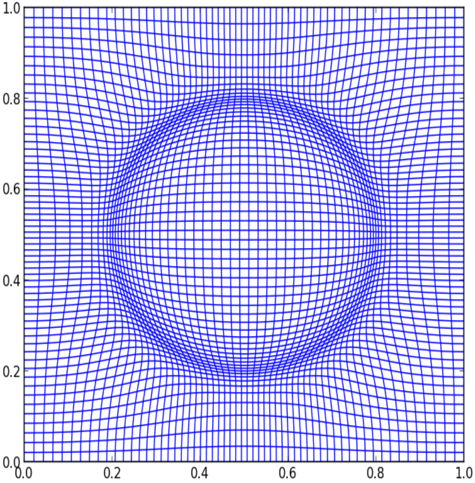


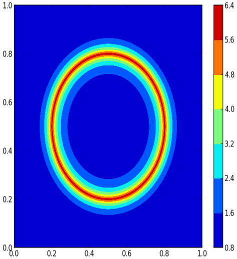
FIGURE: Test : Evolution of the error for the BFO method using Picard's algorithm in the case of $[16 \times 16]$ grid using quadratic B-splines.

N	8×8	16×16	32×32	64×64
Iterations	25	25	25	26

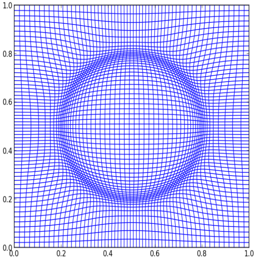
TABLE: Test : Number of iterations until convergence for the BFO method using Picard algorithms with quadratic B-splines.

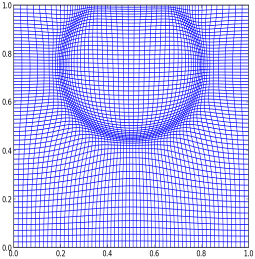
Adaptive meshes — Anistropic case, on a square domain

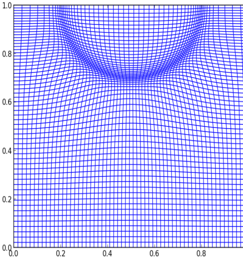




Adaptive meshes — Anistropic case, on a square domain





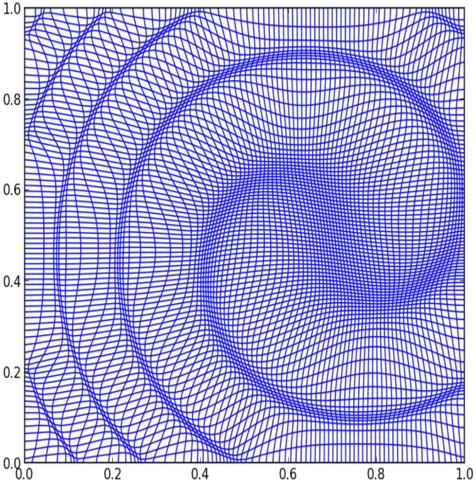


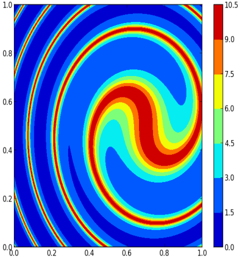
	[Delzanno08]		[Sulman11]	
Number of cells	Error	CPU-time	Error	CPU-time
16×16	4.22×10^{-1}	0.2	6.8×10^{-2}	0.01
32×32	2.17×10^{-1}	0.5	3.28×10^{-2}	0.09
64×64	9.45×10^{-2}	1.8	1.66×10^{-2}	0.6
128×128	2.88×10^{-2}	6.9	8.7×10^{-3}	5
256×256	7.16×10^{-3}	27	4.5×10^{-3}	33.6
512×512	1.76×10^{-3}	109		

	p=2		p=3		p=5	
Grid	Error	CPU-time	Error	CPU-time	Error	CPU-time
8×8	6.22×10^{-2}	0.74	2.05×10^{-2}	0.85	5.48×10^{-3}	1.96
16×16	1.45×10^{-2}	1.81	1.62×10^{-3}	3.13	6.19×10^{-4}	5.9
32×32	2.97×10^{-3}	8.2	1.06×10^{-4}	12.21	5.17×10^{-6}	22.84
64×64	7.19×10^{-4}	37.31	5.27×10^{-6}	50.62	7.46×10^{-8}	92.19
128×128	1.77×10^{-4}	163.26	3.3×10^{-7}	213.1	1.49×10^{-8}	386.53

TABLE: Example 1 : CPU-time needed to solve the Monge-Ampère equation and the adaptive Error $\mathcal{E}_{adp}$ using the two grids Picard algorithm using quadratic, cubic and quintic B-splines.

Adaptive meshes — Anistropic case, on a square domain





Adaptive meshes — Anistropic case, on a square domain

	[Delzanno08]		[Sulman11]	
Number of cells	Error	CPU-time	Error	CPU-time
16×16	4.22×10^{-1}	0.2	2.8×10^{-2}	0.03
32×32	2.17×10^{-1}	0.5	5.6×10^{-3}	0.8
64×64	9.45×10^{-2}	1.8	1.4×10^{-3}	14.6
128×128	2.88×10^{-2}	6.9	3.5×10^{-4}	169
256×256	7.16×10^{-3}	27		
512×512	1.76×10^{-3}	109		

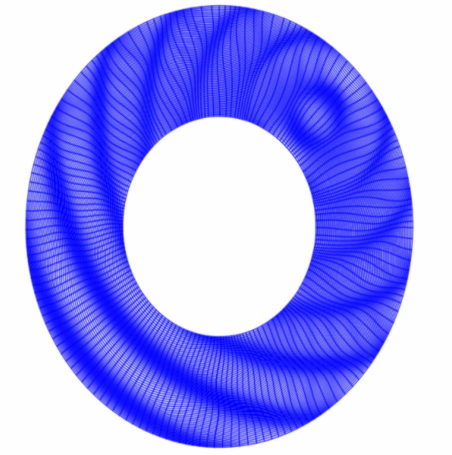
	p=2		p=3		p=5	
Grid	Error	CPU-time	Error	CPU-time	Error	CPU-time
8×8	2.37×10^{-1}	0.46	2.12×10^{-1}	1.29	1.78×10^{-1}	2.59
16×16	1.38×10^{-1}	1.65	1.31×10^{-1}	2.96	1.16×10^{-1}	10.65
32×32	6.35×10^{-2}	9.59	1.75×10^{-2}	17.81	8.06×10^{-3}	38.23
64×64	1.31×10^{-2}	43.21	7.9×10^{-4}	76.58	3.95×10^{-4}	166.74
128×128	2.3×10^{-3}	188.61	2.14×10^{-5}	360.3	1.57×10^{-5}	768.91

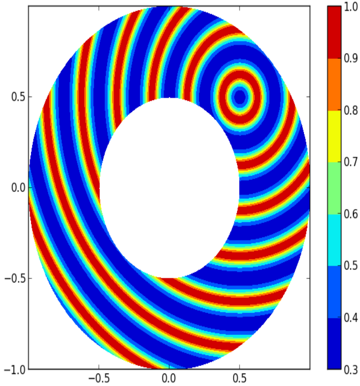
TABLE: Example 2 : CPU-time needed to solve the Monge-Ampère equation and the adaptive Error $\mathcal{E}_{adp}$ using the Picard algorithm with the two grids method.

We can get better result if we have a good initiale guess (more than 50% of time spent far from the solution)

Adaptive meshes — Example on annulus domain

The annulus domain is contructed using NURBS. (NURBS curves model all conics)





Adaptive meshes — A brief comparison

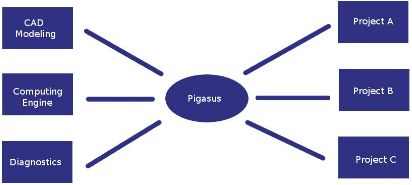
Our Method	Delzanno[2008]
Direct pb	Inverse pb : needs to invert the map
Finite Element, arbitrary degree	Finite Difference
u_h is always convex	-
BFO-method (Picard)	Jacobian Free Newton Krylov
	+ Precond. Multigrid
No parameters are needed	Needed for JFNK
Arbitrary regularity of the final map	Only $\mathcal{C}^0$
Final map is knwon everywhere	Only on the grid
hp-refinement keeps the final map unchanged	-

Summary

- **Act I — Introduction.**
  - Motivations
  - B-splines, NURBS, curves and surfaces
  - The IGA approach
  - Impact of the k-refinement strategy
- **Act II — Adaptive meshes :r-refinement.**
  - Alignement
  - Iterative method using a Posteriori Estimates
  - Equidistribution
- **Act III — Past, present and future.**
  - The **Пgasus** suite
  - Ongoing and forseen work
  - Conclusions.

The Πgasus suite

Πgasus is a generic Python-Fortran PDE solver



- A common Computing Engine,
- More than $50'000$ lines,
- Models : Anisotropic-Diffusion, Reaction-Diffusion, Non-linear Poisson, Monge-Ampère, Grad-Shafranov, 2D reduced MHD,
- Boundary conditions : Dirichlet, Neumann, mixed, periodic,
- Parallelization using MPI,
- Linear solvers : scipy, petsc, pastix, Geometric Multigrid solver,

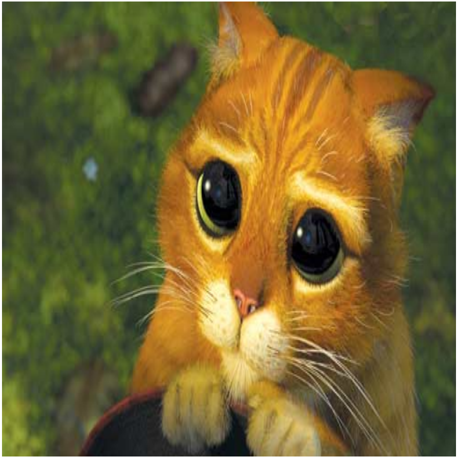
— needs some optimization,

Who is Pigasus ?

In August 1968, he was nominated for President of the United States during the Democratic National Convention as a theatrical gesture by the Youth International Party. At a rally announcing his candidacy, Pigasus was seized by Chicago policemen and his Yippie backers were arrested for disorderly conduct. *Wikipedia*

Conclusions

- ✓ CAD tools well suited to create adaptive geometries for MHD/Kinetic problems,
- ✓ Work on structured meshes, (good for parallelization)
- ✓ Use k-refinement $\rightarrow$ reduce both memory and time costs,
- ✓ Reduce the size of discrete matrices, (But matrices are less sparse ☹)
- ✓ The **Πgasus** suite was developed to write and solve new models (easily)
- ✓ Adaptive mapping scripts are written using **Πgasus** and integrated into **caid** (A CAD tool that can be used as a metric generator)



Thank You !

Foreseen work : Achieve realistic simulations

- ✓ Build IGA meshes for complex and realistic tokamak geometries
- Solve the 3D Monge-Ampère equation (domain decomposition, using a Poisson fast solver),
- Apply the adaptive equidistribution for solving Maxwell's equations,
- Generate Kinetic equilibrium,
- Use Moving FEM for ELMs simulations,
- Understand the impact of the *exact/accurate* geometry on the development of ELMs,
- Upgrade **GYSELA** to treate complex geometries (with L. Mendoza, A. Back, V. Grandgirard, E. Sonnendrücker),
- Numerical studies of Anisotropic diffusion problems using IGA.
- Fast solvers, preconditionners,
- The equistribution technique can be used just after the alignement procedure → reduce the parametrization dependecy,
- Using mixed Aligned - Equidistribution strategies may be of great interest for ELMs simulations,
- ...

APPENDIX-Selected references

- A. Ratnani, et al.. **Пgasus** : Python for IsoGeometric Analysis simulations in Plasmas Physics. (In preparation)
- A. Ratnani, et al.. Alignement and equidistribution for two-dimensional grid adaptation using B-splines. (In preparation)
- A. Ratnani, et al.. Application of the IsoGeometric mesh adaptation for solving the Anistropic Diffusion problem. (In preparation)
- M. J. Baines. Least squares and approximate equidistribution in multidimensions. Numerical Methods for Partial Differential Equations, 1999.
- J. D. Benamou, B. D. Froese, and A. M. Oberman. Two numerical methods for the elliptic monge-ampEre equation. ESAIM : Mathematical Modelling and Numerical Analysis, 2010.
- Y. Brenier. Polar factorization and monotone rearrangement of vector-valued functions. Communications on Pure and Applied Mathematics, 1991.
- C.J. Budd, M.J.P. Cullen, and E.J. Walsh. Monge-amp're based moving mesh methods for numerical e weather prediction, with applications to the eady problem. Journal of Computational Physics, 2013.
- G.L. Delzanno, L. Chacòn, J.M. Finn, Y. Chung, and G. Lapenta. An optimal robust equidistribution method for two-dimensional grid adaptation based on monge-kantorovich optimization. Journal of Com- putational Physics, 2008.
- G. E. Fasshauer and Larry L. Schumaker. Minimal energy surfaces using parametric splines. Computer Aided Geometric Design, 1996.
- M. S. Floater and K. Hormann. Surface parameterization : a tutorial and survey. In N. A. Dodgson, M. S. Floater, and M. A. Sabin, editors, Advances in Multiresolution for Geometric Modelling, Mathematics and Visualization. 2005.
- W. Huang and R. D. Russell. Adaptive moving mesh methods. Applied mathematical sciences. 2011.