A nonlinear approximation approach for solving high-dimensional partial differential equations

Tony Lelièvre
Joint work with Claude Le Bris and Yvon Maday

CERMICS ENPC & Projet MicMac INRIA.

Plan

1. Introduction
2. Greedy algorithms
3. Convergence results
4. Some numerical experiments
5. Discussion and open problems

Introduction

High dimensional PDEs are ubiquitous: kinetic models, molecular dynamics, quantum mechanics, uncertainty quantification using PC expansions, finance, etc.

The bottom line of deterministic approaches is to represent solutions as linear combinations of tensor products of one-dimensional functions (parallelepipedic domains):

$$
\begin{aligned}
u(x_1, \dots, x_N) &= \sum_{k \geq 1} r_k^1(x_1) r_k^2(x_2) \dots r_k^N(x_N) \\
&= \sum_{k \geq 1} \left(r_k^1 \otimes r_k^2(x_2) \dots \otimes r_k^N \right) (x_1, \dots, x_N).
\end{aligned}
$$

If the number of terms in the expansion remains small, this enables to represent the solution while avoiding the curse of dimensionality.

Introduction

How to use such a representation to solve a PDE ?

One approach consists in using the so-called sparse tensor product representation (Griebel, Smolyak, Schwab, Pommier): if u is sufficiently regular, one does not need to use fine discretizations in each directions:

$$C^d \text{ terms} \longrightarrow C\, d \text{ terms}.$$

This can be used in Galerkin-like discretizations.
Main difficulties: regularity of the solution, mesh adaption, implementation.

Here, we consider another approach proposed recently by Chinesta *et al.* to solve high-dimensional Fokker-Planck equations in the context of kinetic models for polymers. The idea is to look iteratively for the best tensor product (nonlinear approximation method).

These are preliminary results:

- the application to option pricing is still an undergoing work
- the numerical analysis of the method is still in its infancy...

Algorithms

Let us consider the simple problem:

$$\text{Find } g \in H_0^1(\Omega) \text{ such that } \begin{cases} -\Delta g = f & \text{in } \Omega, \\ g = 0 & \text{on } \partial\Omega, \end{cases} \tag{1}$$

where $\Omega = \Omega_x \times \Omega_y$ is a bounded domain ($N = 2$) and $f \in L^2(\Omega)$.
Solving (1) is equivalent to:

$$\text{Find } g \in H_0^1(\Omega) \text{ such that } g = \arg \min_{u \in H_0^1(\Omega)} \left(\frac{1}{2} \int_\Omega |\nabla u|^2 - \int_\Omega fu \right). \tag{2}$$

We present the algorithm at the continuous level, but the same principle applies to the space discretized problem. Extension to high dimensional PDEs which derive from a variational principle is straightforward. For time-dependent problems, the same ideas can be used on the problem after time-discretization.

Algorithms

Algorithm 1 (Pure Greedy Algorithm): set $f_0 = f$, and at iteration $n \geq 1$,

1. Find $r_n \in H^1_0(\Omega_x)$ and $s_n \in H^1_0(\Omega_y)$ such that
$$(r_n, s_n) = \arg \min_{(r,s) \in H^1_0(\Omega_x) \times H^1_0(\Omega_y)} \left(\frac{1}{2} \int_\Omega |\nabla (r \otimes s)|^2 - \int_\Omega f_{n-1} \, r \otimes s \right).$$
2. Set $f_n = f_{n-1} + \Delta(r_n \otimes s_n)$.
3. If $\|f_n\|_{H^{-1}(\Omega)} \geq \varepsilon$, proceed to iteration $n+1$. Otherwise, stop.

The idea is to look iteratively for the best tensor product approximations. Notice that

$$f_n = f + \Delta \left(\sum_{k=1}^{n} r_k \otimes s_k \right).$$

Algorithms

Algorithm 2 (Orthogonal Greedy Algorithm): set $f_0^o = f$, and at iteration $n \geq 1$,

1. Find $r_n^o \in H_0^1(\Omega_x)$ and $s_n^o \in H_0^1(\Omega_y)$ such that
$$(r_n^o, s_n^o) = \arg \min_{(r,s) \in H_0^1(\Omega_x) \times H_0^1(\Omega_y)} \left(\frac{1}{2} \int_\Omega |\nabla (r \otimes s)|^2 - \int_\Omega f_{n-1}^o \, r \otimes s \right).$$
2. Solve the following Galerkin problem on the basis $(r_1^o \otimes s_1^o, \ldots, r_n^o \otimes s_n^o)$:
$$(\alpha_1, \ldots, \alpha_n) = \arg \min_{(\beta_1, \ldots, \beta_n) \in \mathbb{R}^n} \left(\frac{1}{2} \int_\Omega \left| \nabla \left(\sum_{k=1}^n \beta_k r_k^o \otimes s_k^o \right) \right|^2 - \int_\Omega f \sum_{k=1}^n \beta_k r_k^o \otimes s_k^o \right).$$
3. Set $f_n^o = f + \Delta \left(\sum_{k=1}^n \alpha_k r_k^o \otimes s_k^o \right)$.
4. If $\|f_n^o\|_{H^{-1}(\Omega)} \geq \varepsilon$, proceed to iteration $n+1$. Otherwise, stop.

In the OGA, one adds an orthogonalization step, consisting in solving a linear problem: Galerkin method on the basis of functions built so far.

The terminology Pure / Orthogonal Greedy Algorithm is borrowed from approximation theory (Cohen, DeVore, Mallat, Avellaneda, Temlyakov).

One can check that the iterations are well defined: At each iteration of the PGA (resp. OGA), the minimization problem admits (at least) one minimizer (r_n, s_n) (resp. (r_n^o, s_n^o)).

In practice, one does not solve the minimization problems but the associated Euler-Lagrange equations.

Algorithms

The functions $(r_n, s_n) \in H_0^1(\Omega_x) \times H_0^1(\Omega_y)$ solving

$$(r_n, s_n) = \arg \min_{(r,s)\in H_0^1(\Omega_x)\times H_0^1(\Omega_y)} \left(\frac{1}{2} \int_\Omega |\nabla(r \otimes s)|^2 - \int_\Omega f_{n-1}\, r \otimes s \right)$$

satisfy: for any functions $(r, s) \in H_0^1(\Omega_x) \times H_0^1(\Omega_y)$

$$\int_\Omega \nabla(r_n \otimes s_n) \cdot \nabla(r_n \otimes s + r \otimes s_n) = \int_\Omega f_{n-1}(r_n \otimes s + r \otimes s_n). \qquad (3)$$

This can be written equivalently as

$$\begin{cases} -\left(\displaystyle\int_{\Omega_y} |s_n|^2\right) r_n'' + \left(\displaystyle\int_{\Omega_y} |s_n'|^2\right) r_n = \displaystyle\int_{\Omega_y} f_{n-1}\, s_n, \\ -\left(\displaystyle\int_{\Omega_x} |r_n|^2\right) s_n'' + \left(\displaystyle\int_{\Omega_x} |r_n'|^2\right) s_n = \displaystyle\int_{\Omega_x} f_{n-1}\, r_n. \end{cases}$$

Algorithms

In practice, the PGA is implemented as: set $f_0 = f$, and at iteration $n \geq 1$,

1. Find $r_n \in H_0^1(\Omega_x)$ and $s_n \in H_0^1(\Omega_y)$ satisfying (3).
2. Set $f_n = f_{n-1} + \Delta(r_n \otimes s_n)$.
3. If $\|f_n\|_{H^{-1}(\Omega)} \geq \varepsilon$, proceed to iteration $n+1$. Otherwise, stop.

Notice that (3) is a nonlinear system of N equations (in dimension N). In practice, it is solved using a simple fixed point procedure: iterate on $k \geq 0$,

$$\begin{cases} -\int_{\Omega_y} |s_n^k|^2 (r_n^{k+1})'' + \int_{\Omega_y} |(s_n^k)'|^2 r_n^{k+1} = \int_{\Omega_y} f_{n-1} s_n^k, \\ -\int_{\Omega_x} |r_n^{k+1}|^2 (s_n^{k+1})'' + \int_{\Omega_x} |(r_n^{k+1})'|^2 s_n^{k+1} = \int_{\Omega_x} f_{n-1} r_n^{k+1}. \end{cases}$$

The same remarks apply for the OGA.

Algorithms

Remarks:

- Starting from a linear problem with exponential complexity wrt N, one ends up with a nonlinear problem with linear complexity (?) wrt N.
- The rhs f needs to enjoy some appropriate separation property with respect to the different coordinates in order for the integrals $\int_\Omega f(r_n \otimes s + r \otimes s_n)$ to be computable.
- The space discretized version of the algorithm consists in solving the discretized Euler-Lagrange equations: find $(r_n^h, s_n^h) \in V_x^h \times V_y^h$ such that, for any functions $(r^h, s^h) \in V_x^h \times V_y^h$

$$\int_\Omega \nabla(r_n^h \otimes s_n^h) \cdot \nabla(r_n^h \otimes s^h + r^h \otimes s_n^h) = \int_\Omega f_{n-1}^h (r_n^h \otimes s^h + r^h \otimes s_n^h),$$

where V_x^h and V_y^h denote *e.g.* finite element spaces.

Convergence

Theorem (Convergence for the PGA)

Consider the Pure Greedy Algorithm, and assume first that (r_n, s_n) satisfies the Euler-Lagrange equations (3). Denote by

$$E_n = \frac{1}{2}\int_\Omega |\nabla(r_n \otimes s_n)|^2 - \int_\Omega f_{n-1}\, r_n \otimes s_n$$

the energy at iteration n. We have

$$\sum_n \int_\Omega |\nabla(r_n \otimes s_n)|^2 = -2\sum_n E_n < \infty,$$

and thus $\lim_{n\to\infty} E_n = 0$. Assume in addition that (r_n, s_n) is a minimizer of (1). Then,

$$\lim_{n\to\infty} f_n = 0 \text{ in } H^{-1}(\Omega).$$

A similar result holds for the OGA.

Convergence

This result is based on the following Lemma which shows that the solutions to the minimization problem are the closest in terms of the $H_0^1(\Omega)$ scalar product.

Lemma

The functions (r_n, s_n) satisfying (1) are such that:
$\forall (r,s) \in H_0^1(\Omega_x) \times H_0^1(\Omega_y)$

$$\|r_n \otimes s_n\| = \frac{\langle r_n \otimes s_n, g_{n-1}\rangle}{\|r_n \otimes s_n\|} \geq \frac{\langle r \otimes s, g_{n-1}\rangle}{\|r \otimes s\|},$$

where $g_n \in H_0^1(\Omega)$ is defined by $-\Delta g_n = f_n$ and $\langle \cdot \rangle$ denotes the scalar product in $H_0^1(\Omega)$:

$$\langle f, g\rangle := \int_\Omega \nabla f \cdot \nabla g,$$

and $\| \cdot \|$ the associated norm.

Convergence

What is the rate of convergence of the method ?

We use general results for greedy algorithms from approximation theory (which holds for general dictionaries).

Let us introduce the functional space:

$$\mathcal{L}^1 = \left\{ g = \sum_{k \geq 0} c_k u_k \otimes v_k, \text{ s.t. } u_k \in H^1_0(\Omega_x), v_k \in H^1_0(\Omega_y), \|u_k \otimes v_k\| = 1 \right.$$
$$\left. \text{and} \sum_{k \geq 0} |c_k| < \infty \right\},$$

and we define the $\mathcal{L}^1$-norm as

$$\|g\|_{\mathcal{L}^1} = \inf \left\{ \sum_{k \geq 0} |c_k|, g = \sum_{k \geq 0} c_k u_k \otimes v_k, \text{ where } \|u_k \otimes v_k\| = 1 \right\},$$

for $g \in \mathcal{L}^1$.

Convergence

The following properties may be established:

- The space $\mathcal{L}^1$ is a Banach space.
- The space $\mathcal{L}^1$ is continuously embedded in $H_0^1(\Omega)$.
- In dimension N, for any $m > 1 + N/2$, $H^m(\Omega) \cap H_0^1(\Omega) \subset \mathcal{L}^1$.

We do not know how to precisely characterize functions in $\mathcal{L}^1$.

Theorem (Rate of convergence)

- PGA: For $g \in \mathcal{L}^1$, we have

$$\|g_n\| \leq \|g\|^{2/3} \|g\|_{\mathcal{L}^1}^{1/3} n^{-1/6}.$$

- OGA: For $g \in \mathcal{L}^1$, we have

$$\|g_n^o\| \leq \|g\|_{\mathcal{L}^1} n^{-1/2}.$$

where $g_n, g_n^o \in H_0^1(\Omega)$ are defined by $-\Delta g_n = f_n$ and $-\Delta g_n^o = f_n^o$.

Convergence

Remark:

- For both algorithms, there exists dictionaries and rhs f (even simple ones, like the sum of two elements of the dictionary) such that the rate of convergence $Cn^{-1/2}$ is attained. We do not know if similar examples can be shown for the specific case when the dictionary is the set of tensor products of one-dimensional functions.

The SVD case

A particular case of interest is when $-\Delta$ is replaced by the identity operator. The PGA writes: set $g_0 = g$, and at iteration $n \geq 1$,

1. Find $r_n \in L^2(\Omega_x)$ and $s_n \in L^2(\Omega_y)$ such that

$$(r_n, s_n) = \arg \min_{(r,s) \in L^2(\Omega_x) \times L^2(\Omega_y)} \left(\int_\Omega |g_{n-1} - r \otimes s|^2 \right).$$

2. Set $g_n = g_{n-1} - r_n \otimes s_n$.
3. If $\|g_n\|_{L^2(\Omega)} \geq \varepsilon$, proceed to iteration $n+1$. Otherwise, stop.

If one considers the discrete version ($\Omega_x = \{1, \ldots, p\}$, $\Omega_y = \{1, \ldots, q\}$, $\int_\Omega \longrightarrow \sum_{(i,j) \in \{1,\ldots,p\} \times \{1,\ldots,q\}}$, $G \in \mathbb{R}^{p \times q}$, $(R_n, S_n) \in \mathbb{R}^p \times \mathbb{R}^q$ and $R_n \otimes S_n = R_n (S_n)^T$), this algorithm amounts to computing the singular value decomposition of the matrix G.

The SVD case

In practice, considering the SVD case is fundamental since one needs to decompose the data (rhs, ic) as (a sum of) tensor products to compute the integrals which appears in the Euler-Lagrange equations (using Fubini).

In the SVD case, one can check that: $\forall n \neq m$

$$\int_{\Omega_x} r_n r_m = \int_{\Omega_y} s_n s_m = 0.$$

This orthogonality property has several consequences:

- PGA=OGA,
- the SVD decomposition is unique (if no degeneracies of the singular values) and the PGA yields the SVD,
- at iteration n, $\sum_{k=1}^n r_k \otimes s_k$ is the minimizer of $\int_\Omega |g - \sum_{k=1}^n \phi_k \otimes \psi_k|^2$ over all possible $(\phi_k, \psi_k)_{1 \leq k \leq n} \in \left(L^2(\Omega_x) \times L^2(\Omega_y)\right)^n$.

All these are lost for the Poisson equation.

Numerical experiments: SVD

Thinking about decomposing the initial condition for BS PDE, let us try to decompose the function

$$\varphi(x_1, \ldots, x_N) = (x_1 + \ldots + x_N - K)_+$$

as a sum of tensor products using the SVD approach.

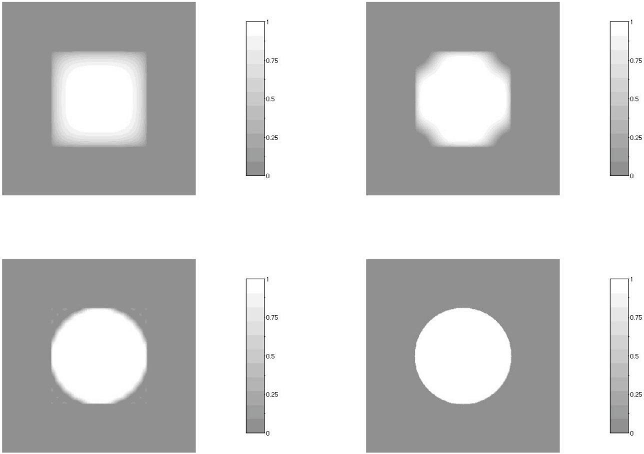
We use $d = 20$ intervals of discretization in each directions, and $\Omega = (0,1)^N$ ($h = 1/(d-1)$). Two approaches: variational approach $G_{i_1,\ldots,i_N} = \int_\Omega \varphi \, \psi^h_{i_1,\ldots,i_N}$, collocation approach $G_{i_1,\ldots,i_N} := \varphi(i_1 h, \ldots, i_N h)$.

Some results for the variational approach:

N	3	4
Number of terms	13	16
Error	$4.5\ 10^{-3}$	$3.7\ 10^{-3}$

Numerical experiments: SVD

Another toy example: approximate the characteristic function of a ball ($d = 100$). Pictures obtained after 1, 2, 5, 60 iterations.



The PGA and OGA are able to find the solution in one iteration if it is a tensor product.

$N = 3$ with analytical solution $g_3(x_1, x_2, x_3) = (x_1^2 - 1)\sin(\pi x_2)\exp(x_3)\sin(\pi x_3)$.

d	20	40	80	100	200
Error PGA	0.040265(2)	0.0102295(2)	0.002567(2)	0.001644(2)	0.000411(2)
Error OGA	0.040264(2)	0.010229(2)	0.002567(2)	0.001644(2)	0.000411(2)

$N = 4$ with analytical solution
$g_4(x_1, x_2, x_3, x_4) = (x_1^2 - 1)\sin(\pi x_2)\exp(x_3)\sin(\pi x_3)(1 - x_4^2)$.

d	20	40	80	100	200
Erreur PGA	0.039723(2)	0.010088(2)	0.002533(2)	0.001625(2)	0.000405(2)
Erreur OGA	0.039710(2)	0.010103(2)	0.002533(2)	0.001622(2)	0.000405(2)

$N = 5$ with analytical solution
$g_5(x_1, x_2, x_3, x_4, x_5) = (x_1^2 - 1)\sin(\pi x_2)\exp(x_3)\sin(\pi x_3)(1 - x_4^2)\sin(\pi x_5)$

d	20	40	80	100	200
Erreur PGA	0.050871(2)	0.013017(2)	0.003271(2)	0.002096(2)	0.000524(2)
Erreur OGA	0.050864(2)	0.013015(2)	0.003271(2)	0.002095(2)	0.000524(2)

Numerical experiments: Poisson

Convergence with respect to the mesh size in dimension $N = 2$ for a non trivial problem:

$g_2(x_1, x_2) = (x_1^3 - x_1).e^{(0.8x_2)}.(x_2^2 - 1) + 5(x_1^3 - x_1 + sin(2\pi x_1)^2).(e^{(0.8x_2)}.(x_2^2 - 1) + e^{(-0.8x_2)}.(x_2^3 - x_2))$.

d	20	40	80	100	200
Error PGA	0.701 (16)	0.0559 (13)	0.00370 (13)	0.001526 (13)	0.000096 (12)
Error OGA	0.701 (15)	0.0559 (13)	0.00370 (12)	0.001526 (12)	0.000096 (12)

$\longrightarrow$ no real difference in practice between PGA and OGA...

Thanks to G. Degrange, I. Hammad and R. Joubaud for these preliminary results.

Numerical experiments: general 2d problem

Let us now consider the discrete problem in 2d. It writes ("$D \simeq -\partial_{xx}$"):

$$\text{Find } G \in \mathbb{R}^{d\times d} \text{ such that } DG + GD = F.$$

The PGA is: Set $F_0 = F$ and at iteration $n \geq 1$,

1. Find R_n and S_n two vectors in $\mathbb{R}^d$ such that:

$$(R_n, S_n) = \arg \min_{(R,S)\in(\mathbb{R}^d)^2} \left(\frac{D(RS^T) + (RS^T)D}{2} - F_{n-1} \right) : (RS^T).$$

2. Set $F_n = F_{n-1} - (DR_nS_n^T + R_nS_n^TD)$.
3. If $\|F_n\| > \varepsilon$, proceed to iteration $n+1$. Otherwise stop.

In practice, Step 1 is replaced by: find R_n and S_n in $\mathbb{R}^d$ such that

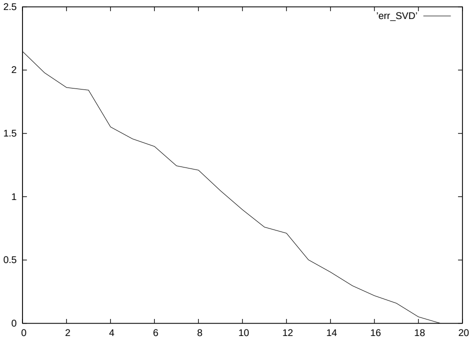
$$\begin{cases} \|S_n\|^2\, DR_n + \|S_n\|_D^2\, R_n = F_{n-1}\, S_n, \\ \|R_n\|^2\, DS_n + \|R_n\|_D^2\, S_n = F_{n-1}^T\, R_n, \end{cases}$$

where, for any vectors $R \in \mathbb{R}^d$, we set $\|R\|_D^2 = R^T DR$.

Numerical experiments: general 2d problem

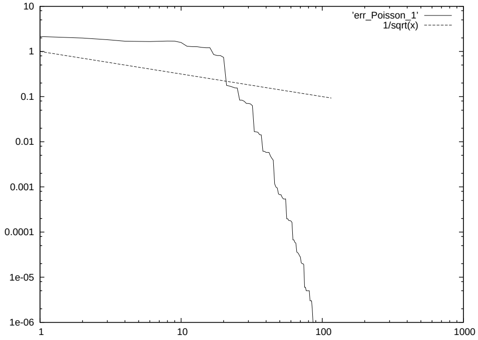
We draw the exact solution G^{exact} at random, compute the associated rhs $F = DG^{\text{exact}} + G^{\text{exact}}D$, and plot the error $\|\sum_{k=1}^{n} R_k S_k^T - G^{\text{exact}}\|$ as a function of the number of iterations n.

SVD case: $D = \mathrm{Id}$, $d = 20$, $\longrightarrow$ Convergence in d iterations.



Numerical experiments: general 2d problem

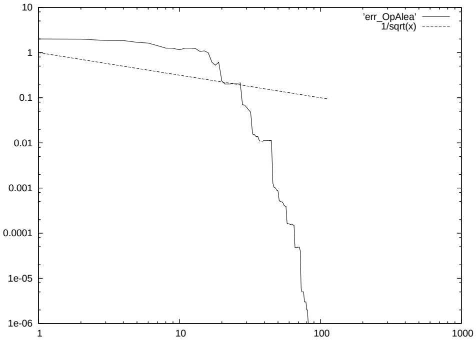
"Poisson" case: D is tridiagonal with $(1, -2, 1)$, $d = 20$



$\longrightarrow$ Slower convergence, the rate of convergence has nothing to do with $1/\sqrt{n}$.

Numerical experiments: general 2d problem

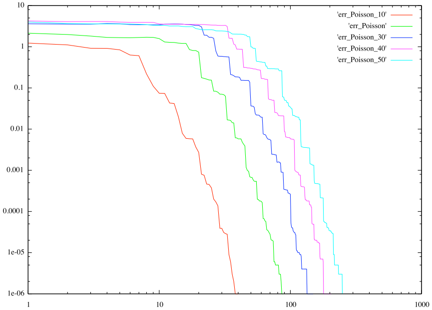
Random case: D is symmetric and random, $d = 20$



$\longrightarrow$ Similar behaviour as for Poisson, but we observe difficulties of convergence for the fixed point procedure.

Numerical experiments: general 2d problem

"Poisson" case: for $d = 10, 20, 30, 40, 50$



Number of iterations to get an error of 10^{-6}:

d	10	20	30	40	50
Nb iterations	39	87	135	180	250

$\longrightarrow$ Convergence roughly linear with respect to d.

Discussion

Using greedy algorithms: For symmetric problems, the method is theoretically on a rather sound basis (except that we solve the EL equations rather than the minimization problem...). However, the numerical analysis does not meet the practical observations (scaling wrt n and d, difference between OGA and PGA), and does not cover the analysis of convergence of the fixed point procedure to solve the nonlinear EL equations.

Going to practical applications: to price basket options, one needs to play with the BS PDE: get rid of boundary conditions, recast it as a symmetric problem, etc. Can the price be efficiently represented as a sum of tensor products ? *Undergoing work...*

Discussion: the non symmetric case

Extensions to non symmetric problems (OGA): set $f_0 = f$, and at iteration $n \geq 1$,

1. Find $r_n \in H_0^1(\Omega_x)$ and $s_n \in H_0^1(\Omega_y)$ such that, for all functions $(r, s) \in H_0^1(\Omega_x) \times H_0^1(\Omega_y)$,

$$\int_\Omega (a \cdot \nabla(r_n \otimes s_n))(r_n \otimes s + r \otimes s_n) + \nabla(r_n \otimes s_n) \cdot \nabla(r_n \otimes s + r \otimes s_n)$$
$$= \int_\Omega f_{n-1}(r_n \otimes s + r \otimes s_n).$$

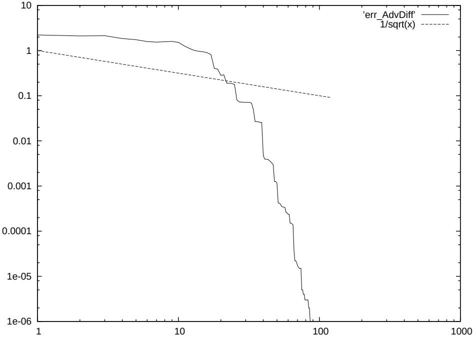
2. Find $u_n \in \mathrm{Vect}(r_1 \otimes s_1, \ldots, r_n \otimes s_n)$ such that for all $v \in \mathrm{Vect}(r_1 \otimes s_1, \ldots, r_n \otimes s_n)$

$$\int_\Omega (a \cdot \nabla u_n) v + \nabla u_n \cdot \nabla v = \int_\Omega f v.$$

3. Set $f_n = f_{n-1} - (a \cdot \nabla u_n - \Delta u_n)$.
4. If $\|f_n\|_{H^{-1}(\Omega)} \geq \varepsilon$, proceed to iteration $n+1$. Otherwise, stop.

Discussion: the non symmetric case

Non symmetric case: D is tridiagonal with $(1, -2-0.2, 1+0.2)$, $d = 20$



$\longrightarrow$ Similar behaviour as for Poisson ! But we observe difficulties of convergence for the fixed point procedure.

Another open question...

References

- A. Ammar, B. Mokdad, F. Chinesta, and R. Keunings. A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids. J. Non-Newtonian Fluid Mech., 139:153-176, 2006.
- C. Le Bris, TL and Y. Maday, Results and questions on a nonlinear approximation approach for solving high-dimensional partial differential equations, http://hal.inria.fr/inria-00336911
- V.N. Temlyakov. Greedy approximation. Acta Numerica, 17:235-409, 2008.