Greedy algorithms and model reduction

T. Lelièvre

Joint work with
E. Cancès, V. Ehrlacher, J. Infante Acevedo, C. Le Bris and Y. Maday

CERMICS, Ecole des Ponts ParisTech & MicMac project-team, INRIA.

Plan

1. Motivation and the algorithm
2. Cross norms
3. The linear case
4. The nonlinear case
5. Implementation of the algorithm
6. Non-symmetric problems

Plan

Motivation

High dimensional PDEs are ubiquitous: kinetic models, molecular dynamics, quantum mechanics, uncertainty quantification (UQ) using polynomial chaos expansions, finance, etc.

In the context of the estimation of parameters in PDEs, the high-dimensionality comes from the number of parameters. Typical example: $\forall t \in \mathcal{T}$,

$$\begin{cases} -\mathrm{div}_x(a(t,x)\nabla_x u(t,x)) = f(t,x) \quad \forall x \in \mathcal{X}, \\ u(t,x) = 0 \quad \forall x \in \partial\mathcal{X}. \end{cases}$$

Any optimization loop will require to solve the PDE for many values of the parameter t. This is why a reduced model may be useful.

Motivation

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

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

If the number of terms in the expansion remains small (this is not the case for full tensor product expansion), this enables to approximate high-dimensional functions.

Motivation

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, Lozynski, Pommier): if u is sufficiently regular, one does not need to use fine discretizations in each directions:

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

This can be used in Galerkin-like discretizations.

Main difficulties: regularity of the solution, mesh adaptation, implementation.

The greedy algorithm

Here, we consider another approach proposed recently by: (i) Chinesta *et al.* to solve high-dimensional Fokker-Planck equations in the context of kinetic models for polymers and (ii) Nouy *et al* in the context of UQ. (See also Ladevèze *et al.* for time-space variable separation.)

These are related to so-called Greedy Algorithms introduced in nonlinear approximation theory: [Temlyakov, Acta Numerica 2008] (Cohen, DeVore, Mallat, Avellaneda, ...).

Other related works: looking for the best n-term approximation of operators: [Kolda, Bader, SIAM Review 2009] (Hackbusch, Beylkin, Mohlenkamp, ...).

Here, we concentrate on:

- Approximation of the solution u by a sum of tensor products,
- Greedy algorithms: look iteratively for the best tensor product,

and applications to high-dimensional PDEs.

The greedy algorithm

Let us consider for simplicity the case of tensor products of only two spaces: $u(t,x) \in V$. The algorithm and (almost) all the results below generalize to the case of tensor products of more than two functions.

Let us introduce a functional $\mathcal{E} : V \to \mathbb{R}$ with a unique global minimizer:

$$u = \operatorname*{argmin}_{v \in V} \mathcal{E}(v).$$

The so-called greedy algorithm writes:

$$(r_n, s_n) \in \operatorname*{argmin}_{r \in V_t,\, s \in V_x} \mathcal{E}\left(\sum_{k=1}^{n-1} r_k \otimes s_k + r \otimes s\right).$$

Here, V, V_t and V_x are Hilbert spaces such that

(H1) $\mathrm{Vect}\{r \otimes s, r \in V_t, s \in V_x\} \subset V$ is dense.

The greedy algorithm

Let us denote

$$u_n = \sum_{k=1}^{n} r_k \otimes s_k.$$

Question: does u_n converge to u ?

Three frameworks:

- The case of cross norms:

$$\mathcal{E}(v) = \|v - u\|_V^2 \text{ and } \|r \otimes s\|_V = \|r\|_{V_t} \|s\|_{V_x}.$$

- The linear case (quadratic functionals):

$$\mathcal{E}(v) = \|v - u\|_V^2$$

(but $\|r \otimes s\|_V \neq \|r\|_{V_t} \|s\|_{V_x}$).

- The nonlinear case (convex functionals):

(H2) $\mathcal{E}$ is α-convex.

$$\exists \alpha > 0, \forall v, w \in V, \; \mathcal{E}(v) \geq \mathcal{E}(w) + \langle \mathcal{E}'(w), v - w \rangle_V + \frac{\alpha}{2} \|v - w\|_V^2.$$

Plan

Cross norms: examples

The case of cross norms: $\mathcal{E}(v) = \|v - u\|_V^2$ and $\|r \otimes s\|_V = \|r\|_{V_t} \|s\|_{V_x}$.

Example: $V = L^2(\mathcal{T} \times \mathcal{X})$, $V_t = L^2(\mathcal{T})$ and $V_x = L^2(\mathcal{X})$. One then looks for an approximation of a function $u(t, x)$ as a sum of tensor products.

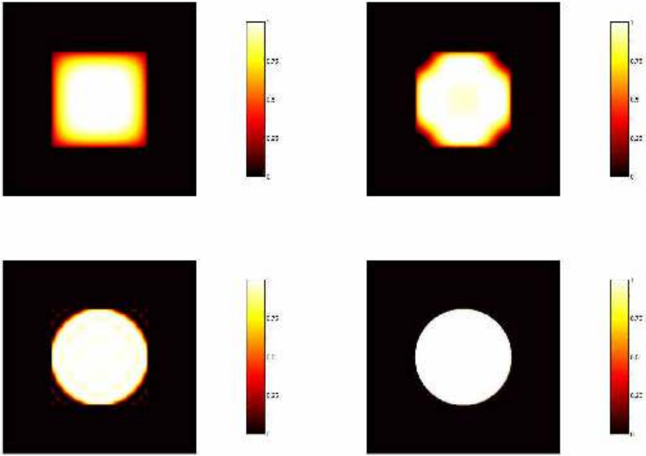
Example 1 (A toy example): Approximation of the characteristic function of a ball in dimension 2.

Example 2: An example in finance (in dimension N): decomposition of the payoff function for a put option

$$\mathcal{E}(v) = \int_{(\mathbb{R}_+)^N} \left| v(x_1, \dots, x_N) - \left(K - \sum_{i=1}^N x_i \right)_+ \right|^2 dx_1 \dots dx_N.$$

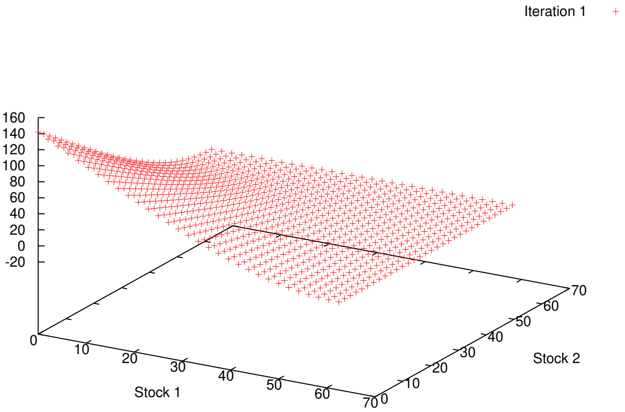
Cross norms: examples

A toy example: approximate the characteristic function of a ball in dimension 2. Approximations obtained after 1, 2, 5, 60 iterations.



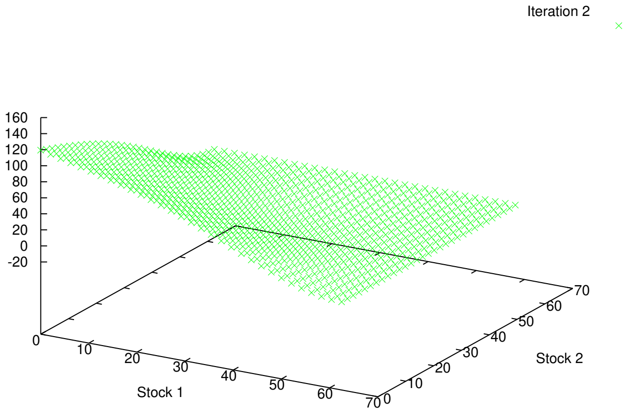
Cross norms: examples

Put option in dimension $N = 2$, with 11 points per dimension.



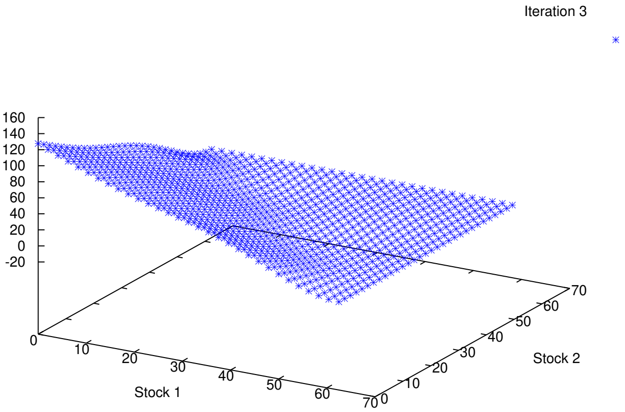
Cross norms: examples

Put option in dimension $N = 2$, with 11 points per dimension.



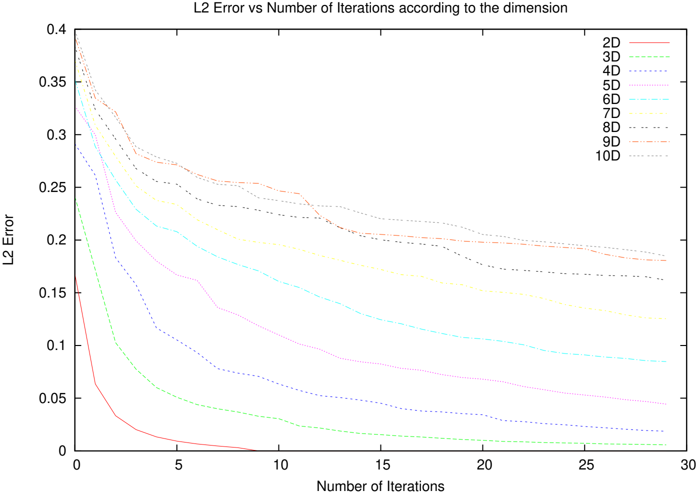
Cross norms: examples

Put option in dimension $N = 2$, with 11 points per dimension.



Cross norms: examples

Put option: L^2 error for dimension $N = 2, \ldots, 10$, with 11 points per dimension.



Cross norms: examples

Put option: Number of iterations needed to obtain a relative error of 10^{-5}, with 11 points per dimension (for comparison $11^8 \simeq 2.10^8$).

Dimension	Number of iterations
1	1
2	2
3	10
4	22
5	101
6	228
7	1077
8	3974

For the same number of terms in the expansion, a full tensor product approximation would have less than 3 dof per dimension ($3^7 = 2187$ $3^8 = 6561$).

Cross norms: relation to Singular Value Decomposition

In the two-dimensional case and for cross-norms, the greedy algorithm is related to the SVD of matrices. Indeed, in the finite dimensional case (u is a matrix and r, s are vectors), the algorithm yields the SVD of u.

A fundamental property of the SVD case: $\forall n \neq m$

$$\langle r_n, r_m \rangle_{V_t} = \langle s_n, s_m \rangle_{V_x} = 0.$$

This yields easily convergence and convergence rates in terms of the spectrum.

This orthogonality property has several consequences:

- the SVD decomposition is unique (up to degeneracies of the singular values),
- at iteration n, $u_n = \sum_{k=1}^n r_k \otimes s_k$ is the minimizer of $\| \sum_{k=1}^n \phi_k \otimes \psi_k - u\|_V^2$ over all possible $(\phi_k, \psi_k)_{1 \leq k \leq n} \in (V_t \times V_x)^n$.

These properties do not hold anymore in dimension larger than 2, or for energies which are not associated to cross norms. Convergence then follows from general convergence results for quadratic functionals $\longrightarrow$

Plan

The linear case: examples

The linear case (quadratic functionals): $\mathcal{E}(v) = \|v - u\|_V^2$ (but $\|r \otimes s\|_V \neq \|r\|_{V_t} \|s\|_{V_x}$).

Example: High-dimensional Poisson equation: $V = H_0^1(\mathcal{T} \times \mathcal{X})$, $V_t = H_0^1(\mathcal{T})$, $V_x = H_0^1(\mathcal{X})$ and $\mathcal{E}(v) = \int_{\mathcal{T} \times \mathcal{X}} a \, |\nabla_{t,x}(v-u)|^2 \, dt \, dx$.

Associated linear problem:

$$-\operatorname{div}_{t,x}(a \nabla_{t,x} v) = f$$

where $f = -\operatorname{div}_{t,x}(a \nabla_{t,x} u)$. Notice that

$$\mathcal{E}(v) = \int_{\mathcal{T} \times \mathcal{X}} a \, |\nabla_{t,x} v|^2 - 2 \int_{\mathcal{T} \times \mathcal{X}} f \, v + C.$$

Only f is needed in practice to implement the algorithm.

The linear case: examples

In higher dimension,

$$\mathcal{E}(v) = \int_{(0,1)^N} a\,|\nabla v|^2 - 2\int_{\mathcal{T}\times\mathcal{X}} f\,v$$

and the greedy algorithm produces an approximate solution as a sum of tensor products:

$$u \approx \sum_{k=1}^{n} r_k^1 \otimes r_k^2 \ldots \otimes r_k^N .$$

Three situations where such elliptic (or parabolic) PDEs appear:
(i) Fokker-Planck equations in kinetic theory (Chinesta *et al*),
(ii) Computation of the commitor function in molecular dynamics,
(iii) Valuation of options in finance.

The linear case: convergence

Convergence analysis:

This case falls into the general theory of Greedy Algorithms developed in approximation theory, for which convergence results have been proven for general dictionaries (and not only tensor products).

Let us mention two results ([De Vore, Temlyakov, 1996] [Le Bris, TL, Maday, 2009]):

- Strong convergence: $\lim_{n\to\infty} u_n = u$ in V.

The linear case: convergence

- Rate of convergence: For $u \in \mathcal{L}^1$, we have

$$\|u - u_n\|_V \le \|u\|_V^{2/3} \|u\|_{\mathcal{L}^1}^{1/3} n^{-1/6}.$$

where

$$\mathcal{L}^1 = \Big\{ u = \sum_{k \ge 0} c_k r_k \otimes s_k, \text{ s.t. } r_k \in V_t,\, s_k \in V_x,\, \|r_k \otimes s_k\|_V = 1 \text{ and } \sum_{k \ge 0} |c_k| < \infty \Big\},$$

and we define the $\mathcal{L}^1$-norm (projective norm) as: for $u \in \mathcal{L}^1$,

$$\|u\|_{\mathcal{L}^1} = \inf \left\{ \sum_{k \ge 0} |c_k|, u = \sum_{k \ge 0} c_k r_k \otimes s_k, \text{ where } \|r_k \otimes s_k\|_V = 1 \right\}.$$

Remark: The rate can be enhanced to $n^{-1/2}$ with an orthogonalized version of the algorithm.

Plan

The nonlinear case: examples

The nonlinear case: $\mathcal{E}$ is α-convex.

Example: Uncertainty Quantification for nonlinear problems:
Let us consider, as an example of UQ in a nonlinear problem, the obstacle problem:

$$\begin{cases} -\operatorname{div}_x(a\nabla_x\overline{u}) \geq f, \\ \overline{u} \geq g, \\ (\operatorname{div}_x(a\nabla_x\overline{u}) + f)\,(\overline{u} - g) = 0, \end{cases}$$

with homogeneous Dirichlet boundary conditions. All the functions ($\overline{u}$, a, f, g) depend on the space variable $x \in \mathcal{X} \subset \mathbb{R}^3$ and on the value $t \in \mathcal{T} \subset \mathbb{R}^p$ of a random variable T.

Aim: Compute how uncertainties on the data ($a(T,x)$, $f(T,x)$ and $g(T,x)$) are propagated on the result ($\overline{u}(T,x)$).

The brute force Monte Carlo algorithm is typically too costly. We are thus interested in computing a reduced model for $\overline{u}$.

The nonlinear case: examples

We again apply the greedy algorithm (see Nouy *et al*, generalized spectral decomposition):

$$\overline{u}(t,x) \approx \sum_{k=1}^{n} r_k(t) s_k(x).$$

Remarks:

- Compared to a full tensor product discretization approach (Galerkin procedure), the complexity goes from KL to $n(K+L)$, where K and L are the number of d.o.f. for functions of t and x respectively.
- If p is large, it is possible to apply the same algorithm to obtain a decomposition:

$$\overline{u}(t,x) \approx \sum_{k=1}^{n} r_k^1(t_1) \dots r_k^p(t_p)\, s_k(x).$$

The nonlinear case: examples

To apply the algorithm, we need to recast the problem as a minimization problem over a Hilbert space. We will therefore consider the solution to the associated penalized formulation (ρ is a large positive parameter):

$$u = \underset{v \in L^2_T(\mathcal{T}, H^1_0(\mathcal{X}))}{\operatorname{argmin}} \mathbb{E}_T \left(\frac{1}{2} \int_{\mathcal{X}} a \, |\nabla_x v|^2 \, dx - \int_{\mathcal{X}} f \, v \, dx + \frac{\rho}{2} \int_{\mathcal{X}} [g - v]_+^2 \, dx \right)$$

where $\mathbb{E}_T$ means that integration on t is wrt to the law of T.

In the limit $\rho \to \infty$, the solution u to the penalized problem converges to the solution $\overline{u}$ to the original obstacle problem.

Thus, in this case, $V_t = L^2_T(\mathcal{T})$, $V_x = H^1_0(\mathcal{X})$, $V = L^2_T(\mathcal{T}, H^1_0(\mathcal{X}))$ and $\mathcal{E}(v) = \mathbb{E}_T \left(\frac{1}{2} \int_{\mathcal{X}} a \, |\nabla_x v|^2 \, dx - \int_{\mathcal{X}} f \, v \, dx + \frac{\rho}{2} \int_{\mathcal{X}} [g - v]_+^2 \, dx \right)$.

The nonlinear case: convergence

Convergence analysis:

Let us assume (H1), (H2), and two additional hypothesis:

(H3) For any sequence $r_n \otimes s_n$ in Σ which is bounded in V, there exists a subsequence which weakly converges in V to an element of Σ.

(H4) $\mathcal{E}$ is differentiable, and its gradient is Lipschitz on bouded sets of V.

$$\forall K \text{ bdd } \subset V, \exists L_K > 0, \forall v, w \in V, \|\mathcal{E}'(v) - \mathcal{E}'(w)\|_V \leq L_K \|v - w\|_V.$$

Remark: (H1)–(H4) are satisfied for the examples above.

Then ([Cancès, Ehrlacher, TL, 2010]), the iterations are well-defined ((r_n, s_n) exists and is non-zero iff $u_{n-1} \neq u$) and: **(i)** Strong convergence still holds:

$$\lim_{n \to \infty} u_n = u \text{ in } V.$$

(ii) In the finite dimensional case, the convergence is exponentially fast: $\exists C > 0, \sigma \in (0,1)$,

$$\|u - u_n\|_V \leq C\sigma^n.$$

The nonlinear case: convergence

(iii) In the case of tensor products of only two functions, these two results (i) and (ii) can be generalized to the case (r_n, s_n) is only a local minimum which ensures the decrease of energy:

$$(r_n, s_n) \in \operatorname*{local\,argmin}_{r \in V_t, s \in V_x} \mathcal{E}\left(\sum_{k=1}^{n-1} r_k \otimes s_k + r \otimes s\right) \text{ and } \mathcal{E}(u_n) < \mathcal{E}(u_{n-1}),$$

under the additional assumption:

(H5) $\exists \beta, \gamma > 0, \forall (r, s) \in V_t \times V_x,$

$$\beta \|r\|_{V_t} \|s\|_{V_x} \leq \|r \otimes s\|_V \leq \gamma \|r\|_{V_t} \|s\|_{V_x}.$$

This assumption is satisfied in the UQ case, and more generally if V_t and V_x are finite dimensional.

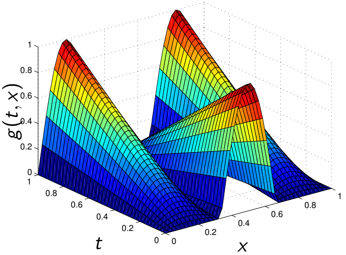
This last result is important since, in practice, only local minima can be computed.

The nonlinear case: UQ for a 1d obstacle problem

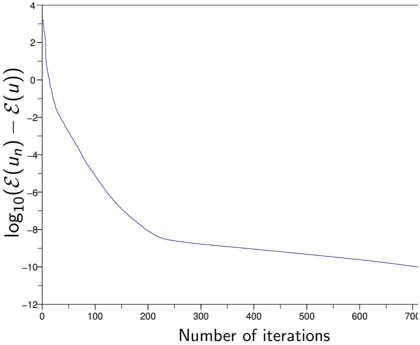
$\mathcal{X} = \mathcal{T} = (0,1)$. T has uniform law on $(0,1)$. The functions f and g are: $\forall (t,x) \in (0,1)^2$,

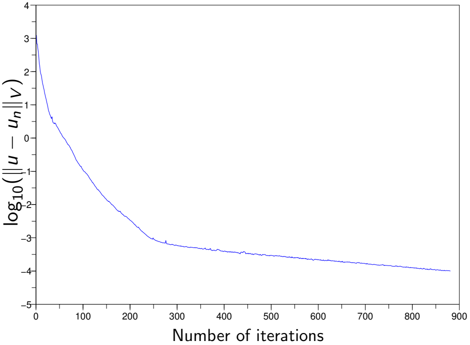
$$f(t,x) = -1 \text{ and } g(t,x) = t[sin(3\pi x)]_+ + (t-1)[sin(3\pi x)]_-.$$

Other parameters: $\rho = 2500$, continuous piecewise linear approximation, with $k = l = 40$ d.o.f.



The nonlinear case: UQ for a 1d obstacle problem





The convergence seems indeed to be exponentially fast.

For really high-dimensional cases, see the works by F. Chinesta, A. Lozynski or A. Nouy, and collaborators.

Plan

Implementation of the algorithm: the Euler equations

How to implement the greedy algorithm in practice ?

Method 1: solve the associated Euler equations.

Let us consider again the Poisson problem (with $a = 1$):

$$\mathcal{E}(v) = \frac{1}{2} \int_{\mathcal{T} \times \mathcal{X}} |\nabla v|^2 - \int_{\mathcal{T} \times \mathcal{X}} f\, v.$$

Recall that $(r_n, s_n) \in \mathrm{argmin}_{r \in H^1_0(\mathcal{T}), s \in H^1_0(\mathcal{X})} \mathcal{E}\left(\sum_{k=1}^{n-1} r_k \otimes s_k + r \otimes s\right)$, or equivalently:

$$(r_n, s_n) \in \underset{r \in H^1_0(\mathcal{T}), s \in H^1_0(\mathcal{X})}{\mathrm{argmin}} \frac{1}{2} \int_{\mathcal{T} \otimes \mathcal{X}} |\nabla (r \otimes s)|^2 - \int_{\mathcal{T} \otimes \mathcal{X}} f_{n-1}\, r \otimes s,$$

where $f_{n-1} = f + \Delta\left(\sum_{k=1}^{n-1} r_k \otimes s_k\right)$. The associated Euler equations write: for any functions $(r, s) \in H^1_0(\mathcal{T}) \times H^1_0(\mathcal{X})$

$$\int_{\mathcal{T} \times \mathcal{X}} \nabla(r_n \otimes s_n) \cdot \nabla(r_n \otimes s + r \otimes s_n) = \int_{\mathcal{T} \times \mathcal{X}} f_{n-1}(r_n \otimes s + r \otimes s_n).$$

Implementation of the algorithm: the Euler equations

This can be written equivalently as

$$\begin{cases} -\left(\int_{\mathcal{X}} |s_n|^2\right) \Delta_t r_n + \left(\int_{\mathcal{X}} |\nabla_x s_n|^2\right) r_n = \int_{\mathcal{X}} f_{n-1}\, s_n, \\ -\left(\int_{\mathcal{T}} |r_n|^2\right) \Delta_x s_n + \left(\int_{\mathcal{T}} |\nabla_t r_n|^2\right) s_n = \int_{\mathcal{T}} f_{n-1}\, r_n. \end{cases}$$

This is a nonlinear coupled system of low-dimensional Poisson equations, which may be solved by a simple fixed point procedure.

Remarks:

- The data (f, or a) is typically approximated by a sum of tensor products in a preliminary step (SVD) to avoid high-dimensional integrals.
- In the UQ context for linear problems, this yields non-intrusive type methods.

Implementation of the algorithm: the Euler equations

Remarks (cont'd):

- Starting from a linear problem with exponential complexity wrt N, one ends up with a nonlinear problem with linear complexity (?) wrt N.
- The space discretized version of the algorithm consists in solving the discretized Euler equations: find $(r_n^h, s_n^h) \in V_t^h \times V_x^h$ such that, for any functions $(r^h, s^h) \in V_t^h \times V_x^h$,

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

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

Implementation of the algorithm: the minimization procedure

For nonlinear problems, it seems difficult to solve the Euler equations by a simple procedure.

Method 2: solve the minmization problem.
Another approach that we have followed in [Cancès, Ehrlacher, TL, 2010] is to use a minimization procedure (quasi-Newton method), with an appropriate technique to choose the initial guess. This yields at each step local minima, with a decrease of the energy.

Plan

Non-symmetric problem

$$\forall v \in V,\ a(u,v) = l(v)$$

where

- $a = a_s + a_{as}$ where a_s is a *symmetric*, *coercive* continuous bilinear form on $V \times V$ and a_{as} an *antisymmetric* continuous bilinear form on $V \times V$;
- l is a continuous linear form on V.

There is no minimization problem formulation of the problem as in the symmetric case!! How can we define the greedy algorithm?

Naive idea: By solving the Euler equations:

$$a(u_{n-1} + s_n \otimes r_n, s_n \otimes \delta r + \delta s \otimes r_n) = l(s_n \otimes \delta r + \delta s \otimes r_n),$$
$$\forall (\delta s, \delta r) \in V_x \times V_t.$$

Convection-diffusion example in a periodic setting

Typical example: $\mathcal{T} = \mathcal{X} = (-1, 1)$, $V_t = L^2(\mathcal{T})$, $V_x = H^1_{\rm per}(\mathcal{X})$, $b \in \mathbb{R}$.

$$\left\{ \begin{array}{l} \text{find } u \in V = V_t \otimes V_x \text{ such that} \\ \forall v \in V, \ a(u, v) = l(v), \end{array} \right. \tag{1}$$

where

$$a(u, v) = \int_{\mathcal{T} \times \mathcal{X}} \nabla_x u \cdot \nabla_x v + (b \cdot \nabla_x u)\, v + uv$$

and

$$l(v) = \int_{\mathcal{T} \times \mathcal{X}} fv,$$

with $f \in L^2_{\rm per}(\mathcal{X}) \otimes L^2(\mathcal{T})$.

Euler equations

For problem (1), at the first iteration, the Euler equations write

$$\left(\int_{\mathcal{T}} |r_1|^2\right)(-\Delta_x s_1 + b\nabla_x s_1 + s_1) = \int_{\mathcal{T}} f\, r_1,$$
$$\left(\int_{\mathcal{X}} |\nabla_x s_1|^2 + |s_1|^2\right) r_1 = \int_{\mathcal{X}} f\, s_1.$$

Problem: If $f = \phi(x-t)$ with $\phi \in L^2_{\rm per}(-1,1)$ an odd function, then the only solution to the "Euler" equations is $r_1 \otimes s_1 = 0$ even for arbitrary small b.

Proof (1/3)

Let us proceed by contradiction and assume that there exits a solution $(r_1, s_1) \in V_t \times V_x$ such that $r_1 \otimes s_1 \neq 0$. The couple (r_1, s_1) can be chosen such that

$$\int_{-1}^{1} |r_1(t)|^2 \, dt = \int_{-1}^{1} |\nabla_x s_1(x)|^2 + |s_1(x)|^2 \, dx = \lambda > 0.$$

$$\left\{ \begin{array}{rcl} -\Delta_x s_1(x) + b\nabla_x s_1(x) + s_1(x) & = & \frac{1}{\lambda} \int_{-1}^{1} f(t,x) r_1(t) \, dt, \\ r_1(t) & = & \frac{1}{\lambda} \int_{-1}^{1} f(t,x) s_1(x) \, dx. \end{array} \right.$$

Proof (2/3)

Plugging the second equation into the first one yields

$$-\Delta_x s_1(x)+b\nabla_x s_1(x)+s_1(x) = \frac{1}{\lambda^2}\int_{-1}^{1}\left(\int_{-1}^{1} f(t,x)f(t,x')\,dt\right) s_1(x')\,dx'. \tag{2}$$

Let us assume that $f(x,t)=\phi(x-t)$ when $\phi \in L^2_{\rm per}(-1,1)$ is an odd function (typically $f(t,x)=\sin(2\pi(x-t))$),

$$\begin{aligned} g(x,x') &= \int_{-1}^{1} f(t,x)f(t,x')\,dt = \int_{-1}^{1}\phi(x-t)\phi(x'-t)\,dt, \\ &= -\int_{-1}^{1}\phi(x-t)\phi(t-x')\,dt = -\int_{-1-x'}^{1-x'}\phi(x-x'-u)\phi(u)\,du, \\ &= -\int_{-1}^{1}\phi(x-x'-u)\phi(u)\,du, \\ &= -\phi*\phi(x-x'). \end{aligned}$$

Proof (3/3)

Fourier transform of (2) yields that for all $k \in \pi\mathbb{Z}$,

$$|k|^2 \widehat{s}(k) + ikb\widehat{s}(k) + \widehat{s}(k) = -\frac{1}{\lambda^2}\left(\widehat{\phi}(k)\right)^2 \widehat{s}(k). \quad (3)$$

Since ϕ is an odd function, $\widehat{\phi}(k) \in i\mathbb{R}$, $\widehat{\phi}(0) = 0$ and this yields that $\widehat{s}(k) = 0$ for all $k \in \pi\mathbb{Z}$.

Conclusion: There are cases when the solution to the original problem is not zero, while the only solution to the Euler-Lagrange equations associated to one iteration of the greedy algorithm is zero !

Non-symmetric problems: which algorithm ?

Let V' be the dual space of V with respect to the $L^2(\mathcal{T} \times \mathcal{X})$ scalar product and let $\|\cdot\|_{V'}$ be its associated norm. Let $A : V \to V'$ and $L \in V'$ such that

$$\left\{ \begin{array}{l} \text{find } u \in V \text{ such that} \\ \forall v \in V, \ a(u,v) = l(v), \end{array} \right.$$

is equivalent to

$$\left\{ \begin{array}{l} \text{find } u \in V \text{ such that} \\ Au = L \text{ in } V', \end{array} \right.$$

Let also denote by $R_V : V \to V'$ be the linear operator such that for all $v \in V$,

$$\|v\|_V = \|R_V v\|_{V'}.$$

Symmetrize the problem: minimization of the L^2 residual

Algorithm 1: Symmetrize the problem by minimizing the L^2 residual [Falco et al, 2012].

In other words, perform the symmetric greedy algorithm on

$$\mathcal{E}(v) = \|Av - L\|_{L^2(\mathcal{X}\times\mathcal{T})}.$$

The Euler equations are the ones associated to the problem

$$A^* A u = A^* L$$

Difficulty: the conditioning of the discretized problems scales quadratically with the conditioning of the original problem...

Symmetrize the problem: minimization of the residual in the dual norm

Algorithm 2: Symmetrize the problem by minimizing the residual in the dual norm. In other words, perform the symmetric greedy algorithm on

$$\mathcal{E}(v) = \|Av - L\|_{V'}^2 = \|R_V^{-1}(Av - L)\|_V^2.$$

The Euler equations are the ones associated to the problem

$$A^*(R_V)^{-1}Au = A^*(R_V)^{-1}L$$

The conditioning of the discretized problems scales linearly with the conditioning of the original problem.

Difficulty: how to compute efficiently $(R_V)^{-1}f$?

Algorithm 3: Perform the greedy algorithm with the symmetric part a_s of the bilinear form a and update the right-hand side at each iteration:

$$(r_n, s_n) \in \underset{(r,s)\in V_t \times V_x}{\text{argmin}} \ \mathcal{E}_{n-1}(r \otimes s),$$

where

$$\mathcal{E}_{n-1}(r \otimes s) = \frac{1}{2} a_s(u_{n-1} + r \otimes s, u_{n-1+} r \otimes s) - l(r \otimes s) - a_{as}(u_{n-1}, r \otimes s)$$

with $u_{n-1} = \sum_{k=1}^{n-1} r_k \otimes s_k$. In other words, at each iteration, one performs one greedy iteration on the problem

$$\forall v \in V, \ a_s(u, v) = l(v) - a_{as}(u_{n-1}, v).$$

Partial convergence results

Of course, such an algorithm is expected to converge only if the antisymmetric part is small enough.

Result: If V_x and V_t are finite-dimensional, there exists $\kappa > 0$ such that if $\|a_{as}\|_{\mathcal{L}(V,V)} \leq \kappa \|a_s\|_{\mathcal{L}(V,V)}$, then the algorithm converges strongly in V.

Difficulty: In the proof, the rate κ seems to depend on the dimension of V_x and V_ξ.

Numerically, the rate κ seems not to depend on the dimension...

Other algorithms in the literature

- Minimax algorithm [Nouy, 2010]
- Dual algorithm, X-Greedy algorithm (Lozinski, based on ideas of Temlyakov)

Good numerical results but no theoretical proof of convergence.

Greedy algorithms

Non-linear approximation techniques are very promising to tackle high-dimensional problems.

Related works:

- Application to high-dimensional PDEs in finance (J. Infante Acevedo).
- Application to eigenvalue problems (E. Cancès and V. Ehrlacher) $\longrightarrow$ see the talk by V. Ehrlacher this afternoon (15h15, MENUDI, Salle Pala).

Open problems:

- How to modify the algorithm for non-symmetric problems ?
- How to obtain a rate of convergence in nonlinear infinite dimensional cases ?
- How to reduce the number of terms generated by the algorithm ?
- On which type of problems this technique is efficient ?

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.
- E. Cancès, V. Ehrlacher and TL, Convergence of a greedy algorithm for high-dimensional convex nonlinear problems, M3AS, 21(12), 2433-2467, (2011).
- E. Cancès, V. Ehrlacher and TL, Greedy algorithms for high-dimensional non-symmetric linear problems, `http://arxiv.org/abs/1210.6688` .
- T.G. Kolda and B.W. Bader, Tensor Decompositions and Applications, SIAM Rev., 51(3), 455-500, 2009.
- C. Le Bris, TL and Y. Maday. Results and questions on a nonlinear approximation approach for solving high-dimensional partial differential equations, Constructive Approximation 30(3):621-651, 2009.
- A. Nouy, Recent Developments in Spectral Stochastic Methods for the Numerical Solution of Stochastic Partial Differential Equations, Archives of Computational Methods in Engineering 16:251-285, 2009.
- A. Nouy and O. Le Maitre. Generalized spectral decomposition for stochastic nonlinear problems, J. Comput. Phys. 228(1):202-235, 2009.
- V.N. Temlyakov. Greedy approximation. Acta Numerica, 17:235-409, 2008.