Greedy algorithms for high-dimensional non-symmetric problems

V. Ehrlacher
Joint work with E. Cancès et T. Lelièvre

Financial support from Michelin is acknowledged.

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

Outline

Outline

1. Greedy algorithms for symmetric coercive problems
2. Problem in the non-symmetric case
3. Other algorithms

Curse of dimensionality

We wish to approximate a function $u(x, \xi_1, \cdots, \xi_d)$ where $\xi_1, \cdots, \xi_d$ are random variables with d very large.

$$x \in \mathcal{X}, \ \xi_1 \in \Xi_1, \ \cdots, \xi_d \in \Xi_d$$

Standard (Galerkin methods):
a priori fixed basis functions $(\phi_i(x))_{1 \leq i \leq N}$, $(\psi_{j_1}(\xi_1))_{1 \leq j_1 \leq N}$, ... , $(\psi_{j_d}(\xi_d))_{1 \leq j_d \leq N}$

$$u(x, \xi_1, \cdots, \xi_d) \approx \sum_{1 \leq i, j_1, \cdots, j_d \leq N} \lambda_{i, j_1, \cdots, j_d} \phi_i(x) \psi_{j_1}(\xi_1) \cdots \psi_{j_d}(\xi_d).$$

$$\boxed{DIM = N^d}$$

Separated variable representation

Separated variable representation (also called canonical format)

The solution is represented as linear combinations of tensor products of small-dimensional functions to avoid the curse of dimensionality [Bellman, 1957]:

$$
\begin{aligned}
u(x, \xi_1, \cdots, x_d) &\approx \sum_{k=1}^{n} s_k(x) r_k^1(\xi_1) \dots r_k^d(\xi_d) \\
&= \sum_{k=1}^{n} \left(s_k \otimes r_k^1 \dots \otimes r_k^d \right) (x, \xi_1, \dots, \xi_d).
\end{aligned}
$$

$$\boxed{DIM = nNd}$$

Symmetric problem

$u(x, \xi_1, \cdots, \xi_d) \in V$ where $V = V_x \otimes V_{\xi_1} \otimes \cdots V_{\xi_d}$.

$V_x \subset L^2(\mathcal{X}),\ V_{\xi_1} \subset L^2(\Xi_1),\ \cdots,\ V_{\xi_d} \subset L^2(\Xi_d),\ V \subset L^2(\mathcal{X} \times \mathcal{X}_1 \times \cdots \times \Xi_d)$

$$\forall v \in V,\ a(u,v) = l(v) \tag{1}$$

where

- a is a *symmetric*, *coercive* continuous bilinear form on $V \times V$;
- l is a continuous linear form on V.

Typical example:

$$\left\{ \begin{array}{l} \text{Find } u \in V = H^1(\mathcal{X}) \otimes L^2_p(\Xi), \\ \mathbb{E}\left[\int_{\mathcal{X}} \nabla_x u(x,\xi) \cdot \nabla_x v(x,\xi) + u(x,\xi)v(x,\xi)\, dx\right] = \mathbb{E}\left[\int_{\mathcal{X}} f(x,\xi)v(x,\xi)\, dx \right. \end{array} \right. \tag{2}$$

with $f \in L^2(\mathcal{X}) \otimes L^2_p(\Xi)$.

$$V_x = H^1(\mathcal{X}),\ V_\xi = L^2_p(\Xi).$$

Greedy algorithm - Progressive Generalized Decomposition

We consider an approach proposed by:

- Ladevèze *et al.* to do time-space variable separation
- Chinesta *et al.* to solve high-dimensional Fokker-Planck equations in the context of kinetic models for polymers [Ammar et al., 2006]
- Nouy *et al* in the context of UQ. [Nouy, 2009]

In the *symmetric coercive* setting, problem (1) can be rewritten as an optimization problem

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

where $\mathcal{E}(v) = \frac{1}{2}a(v,v) - l(v)$.

Definition of the greedy algorithm

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

$(x, \xi) \in \mathcal{X} \times \Xi$ $\qquad$ $u(x,\xi) = \sum_{k \geq 1} s_k(x) r_k(\xi)$.

V $\qquad$ V_x $\qquad$ V_ξ

where the solution u is the unique global minimizer of the functional $\mathcal{E} : V \to \mathbb{R}$.

Greedy algorithm [Temlyakov, 2008]: We define recursively

$$(r_n, s_n) \in \underset{(r,s) \in V_t \times V_x}{\operatorname{argmin}} \ \mathcal{E}\left(\sum_{k=1}^{n-1} r_k \otimes s_k + r \otimes s \right) \qquad (3)$$

Let us denote $u_n = \sum_{k=1}^{n} r_k \otimes s_k$.
Question: Does u_n converge towards u? Yes!

Convergence results

[Le Bris, Lelièvre, Maday, 2009], [Cancès, VE, Lelièvre, 2011] or [Nouy, Falco, 2011]

$$\Sigma = \{r \otimes s, (r,s) \in V_x \times V_\xi\}$$

- $\mathrm{Span}(\Sigma)$ dense in V;
- Σ is weakly closed in V;
- $\mathcal{E}$ is differentiable and $\mathcal{E}'$ is Lipschitz continuous on bounded sets;
- $\mathcal{E}$ is elliptic, i.e. there exists $\alpha > 0$ and $s > 1$ such that

$$\forall v, w \in V, \ \langle \mathcal{E}'(v) - \mathcal{E}'(w), v - w \rangle_V \geq \alpha \|v - w\|_V^s;$$

$$\boxed{u_n \underset{n\to\infty}{\longrightarrow} u}$$

Remark: $\mathcal{E}$ does not necessarily need to be a quadratic functional for the algorithm to converge.

Euler equations

How are computed the $(s_n, r_n) \in V_x \times V_\xi$ in practice? By solving the Euler equations:

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

For problem (5),

$$\mathbb{E}\left[r_n(\xi)^2\right](-\Delta_x s_n(x) + s_n(x)) = \mathbb{E}\left[(f(x,\xi) + \Delta_x u_{n-1}(x,\xi)) r_n(\xi)\right],$$
$$\left(\int_{\mathcal{X}} |\nabla_x s_n(x)|^2 + |s_n(x)|^2 \, dx\right) r_n(\xi) = \int_{\mathcal{X}} (f(x,\xi) + \Delta_x u_{n-1}(x,\xi)) s_n(x) \, dx.$$

Outline

Non-symmetric problem

$$\forall v \in V, \; a(u,v) = l(v) \tag{4}$$

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 minization problem formulation of the problem as in the symmetric case!! How can we define the PGD/greedy algorithm?

Naive idea: By solving the Euler equations:

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

Convection-diffusion example

Let us denote by A the operator on $L^2(\mathcal{X} \times \Xi)$, with domain $D(A)$ such that

$$\forall u, v \in D(A),\ \langle Au, v\rangle_{L^2(\mathcal{X}\times\Xi)} = a(u, v).$$

Typical example: $A = -\Delta_x + b \cdot \nabla_x + 1 = A_x \otimes I_\xi$, $D(A) = H^2(\mathcal{X}) \otimes L^2_p(\Xi)$, with $A_x = -\Delta_x + b \cdot \nabla_x + 1$ on $L^2(\mathcal{X})$ and I is the identity operator on $L^2_p(\xi)$.

$$\left\{ \begin{array}{l} \text{Find } u \in V = H^1(\mathcal{X}) \otimes L^2_p(\Xi), \\ \mathbb{E}\left[\int_{\mathcal{X}} \nabla_x u(x,\xi) \cdot \nabla_x v(x,\xi) + b \cdot \nabla u(x,\xi) v(x,\xi) + u(x,\xi) v(x,\xi)\, dx\right] \\ = \mathbb{E}\left[\int_{\mathcal{X}} f(x,\xi) v(x,\xi)\, dx\right], \end{array} \right. \tag{5}$$

with $f \in L^2(\mathcal{X}) \otimes L^2_p(\Xi)$.

$$V_x = H^1(\mathcal{X}),\ V_\xi = L^2_p(\Xi).$$

"Euler" algorithm

For problem (5),

$$\mathbb{E}\left[r_n(\xi)^2\right] A_x s_n(x) = \mathbb{E}\left[(f(x,\xi) + Au_{n-1}(x,\xi))r_n(\xi)\right],$$

$$\left(\int_{\mathcal{X}} (A_x s_n(x)) s_n(x)\, dx\right) r_n(\xi) = \int_{\mathcal{X}} (f(x,\xi) + Au_{n-1}(x,\xi)) s_n(x)\, dx.$$

Problem: There are cases where the only solutions of these Euler equations are $(r_n, s_n) = (0, 0)$ even if $u_{n-1} \neq u_n$!!

Outline

Symmetrize the problem: minimization of the L^2 residual

[Falco et al, 2011]

The idea is then to symmetrize the problem by minimizing the L^2 residual.
In the convection-diffusion case, perform the symmetric greedy algorithm on

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

Euler equations: minimization of the L^2 residual

When f is regular enough for the convection diffusion problem

$$\mathbb{E}\left[r_n(\xi)^2\right] A_x^* A_x s_n(x) = \mathbb{E}\left[(A^*(f(x,\xi) + Au_{n-1}(x,\xi)))r_n(\xi)\right],$$

$$\left(\int_{\mathcal{X}} \langle A_x^* A_x s_n, s_n \rangle_{L^2(\mathcal{X})} \, dx\right) r_n(\xi) = \int_{\mathcal{X}} (A^*(f(x,\xi) + Au_{n-1}(x,\xi)))s_n(x) \, dx.$$

The Euler equations are the ones associated to the problem

$$A^* A u = A^* f$$

The conditining of the discretized problems scales quadratically with the conditioning of the original problem!!

Symmetrize the problem: minimization of the H^{-1} residual

The idea is then to symmetrize the problem by minimizing the H^{-1} residual. In the convection-diffusion case, perform the symmetric greedy algorithm on

$$\mathcal{E}(v) = \|Av - f\|_{H^{-1}(\mathcal{X})\otimes L^2(\Xi)}.$$

Euler equations: minimization of the L^2 residual

$$\mathbb{E}\left[r_n(\xi)^2\right] A_x^*(-\Delta_x)^{-1} A_x s_n(x)$$
$$= \mathbb{E}\left[(A^*(-\Delta_x)^{-1}(f(x,\xi) + Au_{n-1}(x,\xi)))r_n(\xi)\right],$$
$$\left(\int_{\mathcal{X}} \langle A_x^*(-\Delta_x)^{-1} A_x s_n, s_n\rangle_{L^2(\mathcal{X})}\, dx\right) r_n(\xi)$$
$$= \int_{\mathcal{X}} (A^*(-\Delta_x)^{-1}(f(x,\xi) + Au_{n-1}(x,\xi)))s_n(x)\, dx.$$

The Euler equations are the ones associated to the problem

$$A^*(-\Delta_x)^{-1} Au = A^*(-\Delta_x)^{-1} f$$

The conditining of the discretized problems scales linearly with the conditioning of the original problem!!

However, this method needs to solve a lot of small-dimensional Poisson problems. It takes more time (although it can be done in parallel) and more memory.

Explicitation of the antisymmetric part

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

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

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.

Proposition

If V_x and V_ξ 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.

Problem: The rate κ depends on the dimension of V_x and V_ξ.

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

Use a symmetric formulation of the antisymmetric problem

Idea: use a symmetric version of the antisymmetric problem [Cohen et al., 2011]

For the convection diffusion problem:

$$\begin{pmatrix} 0 & A^* \\ A & -\Delta \end{pmatrix} \begin{pmatrix} u \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ f \end{pmatrix}$$

This is a symmetric problem whose solution is $(v, y) = (u, 0)$.

But it is not a coercive problem!!!

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.

References

- R.E. Bellman. Dynamic Programming. Princeton University Press, 1957.
- 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.E., Tony Lelièvre. Convergence of a greedy algorithm for high-dimensional convex nonlinear problems, M3AS, 2011.
- A. Nouy and A. Falco. Proper Generalized Decomposition for Nonlinear Convex Problems in Tensor Banach Spaces", Numerische Mathematik,2011.
- C. Le Bris, Tony Lelièvre 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, A priori model reduction through Proper Generalized Decomposition for solving time-dependent partial differential equations, *preprint*, 2010.
- V.N. Temlyakov. Greedy approximation. Acta Numerica, 17:235-409, 2008.
- A. Falco, L. Hilario, N. Montés, M.C. Mora. Numerical Strategies for the Galerkin Proper Generalized Decomposition Methods. 2011.
- A. Cohen, W. Dahmen, G. Welper. Adaptivity and Variational Stabilization for Convection-Diffusion Equations, 2011.

Thank you for your attention!