Implementing hybrid PDE solvers

George Sarailidis and Manolis Vavalis

ECE Department, University of Thessaly

November 14, 2015

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Outline

Objectives

Overall Objective

Effectively combine

- conventional deterministic PDE solving methods and
- traditional probabilistic Monte Carlo approaches

for solving linear Elliptic Partial Differential Equations.

Objectives

Overall Objective

Effectively combine

- conventional deterministic PDE solving methods and
- traditional probabilistic Monte Carlo approaches

for solving linear Elliptic Partial Differential Equations.

Short term goal

Design, implement and evaluate a robust and easy to use prototype system that allows further experimentation in order to elucidate the capabilities and computational characteristics of the resulting PDE solvers.

Objectives

Overall Objective

Effectively combine

- conventional deterministic PDE solving methods and
- traditional probabilistic Monte Carlo approaches

for solving linear Elliptic Partial Differential Equations.

Short term goal

Design, implement and evaluate a robust and easy to use prototype system that allows further experimentation in order to elucidate the capabilities and computational characteristics of the resulting PDE solvers.

Long term goal

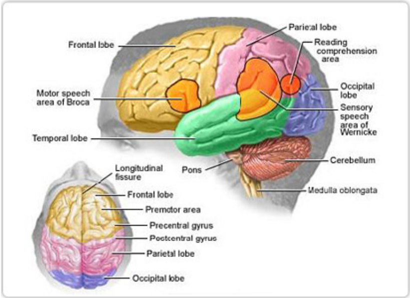
Provide practical (production?) computational tools that add value to existing high performance PDE solvers.

Basic idea

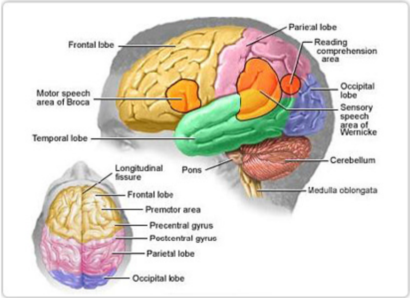
Stochastic pre-processing A Monte Carlo-based walk on spheres approach is utilized to decouple the original PDE problem into a set of independent PDE sub-problems.

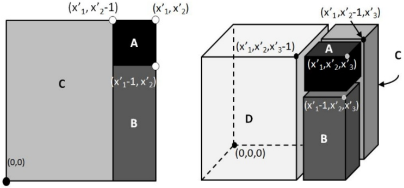
Deterministic solving Each of the resulting sub-problems is solved independently by an appropriately selected PDE solver.

Long and short term: PDE problems

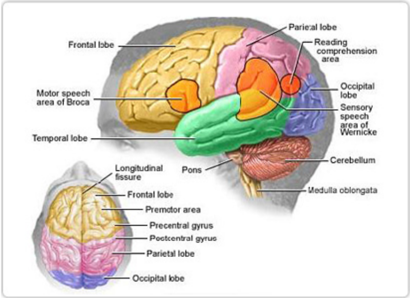


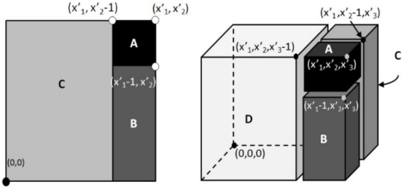
Long and short term: PDE problems





Long and short term: PDE problems





BEWARE:

Specific Algorithm (1/2)

PDE Problem

$$
\begin{aligned}
Lu(x) &= f(x) \quad x \in \mathcal{D} \subset \mathbb{R}^d, \\
Bu(x) &= g(x) \quad x \in \partial\mathcal{D},
\end{aligned}
$$

Specific Algorithm (1/2)

PDE Problem

$$Lu(x) = f(x) \quad x \in \mathcal{D} \subset \mathbb{R}^d,$$
$$Bu(x) = g(x) \quad x \in \partial\mathcal{D},$$

Subdomains and Interfaces

$$\mathcal{D} = \cup_{\mu=1}^{\mathcal{N}_{\mathcal{D}}} \mathcal{D}_\mu$$
$$\mathcal{I}_{\mu,\nu} = \partial\mathcal{D}_\mu \cap (\partial\mathcal{D}_\nu \cup \mathcal{D}_\nu) \subset \mathbb{R}^{d-1}, \quad \mu \neq \nu, \quad \mu, \nu = 1, \ldots, \mathcal{N}_{\mathcal{D}}.$$

Specific Algorithm (2/2)

Data: $i_1, i_2, \ldots, i_N$: subdomains we wish to compute the solution.
Result: $\tilde{u}_\mu$, $\mu = i_1, \ldots, i_N$: computed solutions in $\mathcal{D}_\mu$

Specific Algorithm (2/2)

Data: $i_1, i_2, \dots, i_N$: subdomains we wish to compute the solution.
Result: $\tilde{u}_\mu$, $\mu = i_1, \dots, i_N$: computed solutions in $\mathcal{D}_\mu$

```
// PHASE I: Estimate solution on interfaces
```

while $\mathcal{I}_{\mu,\nu} \subset \cup_{j=1}^{N} \partial \mathcal{D}_{i_j}$ **do**

  Select control points $x_i \in \mathcal{I}_{\mu,\nu}, \quad i = 1, 2, \dots, M_{\mu,\nu}$;
  Estimate the solution u at x_i by Monte Carlo;
  Calculate the interpolant $u^I_{\mu,\nu}$ of $u_{\mu,\nu}$ using x_i;

end

Specific Algorithm (2/2)

Data: $i_1, i_2, \ldots, i_N$: subdomains we wish to compute the solution.
Result: $\tilde{u}_\mu$, $\mu = i_1, \ldots, i_N$: computed solutions in $\mathcal{D}_\mu$

`// PHASE I: Estimate solution on interfaces`

while $\mathcal{I}_{\mu,\nu} \subset \cup_{j=1}^{N} \partial\mathcal{D}_{i_j}$ **do**

Select control points $x_i \in \mathcal{I}_{\mu,\nu}, \quad i = 1, 2, \ldots, M_{\mu,\nu}$;
Estimate the solution u at x_i by Monte Carlo;
Calculate the interpolant $u^I_{\mu,\nu}$ of $u_{\mu,\nu}$ using x_i;

end

`// PHASE II: Compute solution in subdomains`

for $j = 1, 2, \ldots, N$ **do**

Solve the PDE problem:;

$$L_{i_j} u_{i_j}(x) = f_{i_j}(x) \quad x \in \mathcal{D}_{i_j} \; ;$$
$$B_{i_j} u_{i_j}(x) = g_{i_j}(x) \quad x \in \partial\mathcal{D}_{i_j} \cap \partial\mathcal{D} \; ;$$
$$L^*_{i_j} u_{i_j}(x) = h_{i_j}(x) \quad x \in \mathcal{D}_{i_j}$$

end

State-of-the-art

- M. Muller, *Some Continuous Monte Carlo Methods for the Dirichlet Problem*, Annals Mathem Statistics, 1956.
- J. DeLaurentis and L. Romero, *A Monte Carlo method for Poisson's equation*, J. Comput. Phys., 1990.
- M. Mascagni, A. Karaivanova, and Y. Li, *A quasi-Monte Carlo method for elliptic PDEs*. Monte Carlo Methods and Applications, 2001.
- R. Papancheva, I. Dimov, and T. Gurov. *A new class of grid-free Monte Carlo algorithms for elliptic BVP*, Numerical Methods and Applications, 2003.
- J. Acebron, M. Busico, P. Lanucara, and R. Spigler, *Domain decomposition solution of elliptic boundary-value problems via Monte Carlo and quasi-Monte Carlo methods*. SIAM Journal of Sci. Comp., 2006.
- PDD-HPC Research project (Acebron at al.)

Nevertheless

To the best of our knowledge

- No systematic experimentation as been performed.
- No state-of-the-art PDE solving modulus have been considered.
- No practical issues have been raised.
- No software components are readily available.
- No new computing paradigms have been explored.

Nevertheless

To the best of our knowledge

- No systematic experimentation as been performed.
- No state-of-the-art PDE solving modulus have been considered.
- No practical issues have been raised.
- No software components are readily available.
- No new computing paradigms have been explored.

Isolated and sporadic efforts.

Basic algorithms and software components

- Our QMC implementations based on known random walks on spheres algorithms. GRID FREE!
- Use of standard C++ library
- Common 2D and 3D user interface
- SINTEF's Multilevel B-splines library for interpolation
- State of the art PDE solvers from http://www.dealii.org/ (2007 J. H. Wilkinson Prize for Numerical Software)
- State of the art graphics integration (TecPlot)

Selected experiments in 2D

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y),\ \forall (x,y) \in \Omega \equiv [-1,1] \times [-1,1],$$

$$\left.\begin{aligned} u(\pm 1, y) = &\quad \cosh(\pm 2)\cos(2\pi y) \\ u(x, \pm 1) = &\quad \sin(\pi x)\sinh(\pm 1) + \cosh(2x), \end{aligned}\right. \quad \forall (x,y) \in \partial\Omega.$$

$$u(x,y) = \sin(\pi x)\sinh(y) + \cosh(2x)\cos(2\pi y).$$

Selected experiments in 2D

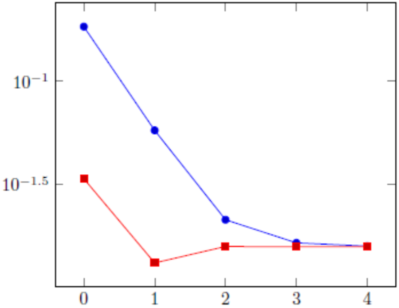
$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y),\ \forall (x,y) \in \Omega \equiv [-1,1] \times [-1,1],$$

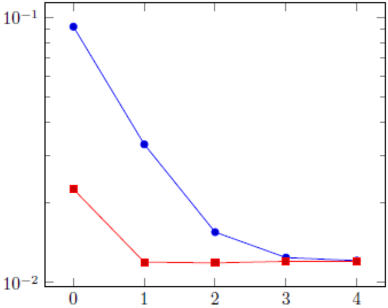
$$\begin{array}{ll} u(\pm 1, y) = & \cosh(\pm 2)\cos(2\pi y) \\ u(x, \pm 1) = & \sin(\pi x)\sinh(\pm 1) + \cosh(2x), \end{array} \quad \forall (x,y) \in \partial\Omega.$$

$$u(x,y) = \sin(\pi x)\sinh(y) + \cosh(2x)\cos(2\pi y).$$

- 8 subdomains with interfaces at $x_1 = 0$, $y_1 = -0.5$, $y_2 = 0$ and $y_3 = 0.75$.
- Seek the solution only in $\Omega_{1,0}$, $\Omega_{0,1}$ and $\Omega_{2,1}$

Error reductions for various configurations





GPU experimentation

- Computing devices:
  - CPU: Intel(R) Core(TM) i7 CPU 870 (2.93GHz)
  - GPU: GeForce GTX 480 (1401MHz)
- Speedups CPU/GPU+CPU
  - Maximum speedup of QMC: 13x
  - Maximum overall speedup: 150x

Trivial implementation!

Web Services experimentation

- Computing servers:
  - Local PCs
  - Tier 3 local computational servers
  - Amazon virtual machines

Low communication/computation ratio

Trivial implementation!

Synopsis and Prospects

- Potential for
  - Software reuse
  - Hardware utilization
  - Utilization of high performance PDE solvers and interpolants
  - Further experimentation
  - Huge speedups and practical use

Synopsis and Prospects

- Potential for
  - Software reuse
  - Hardware utilization
  - Utilization of high performance PDE solvers and interpolants
  - Further experimentation
  - Huge speedups and practical use
- Questions need to be answered
  - Error analysis
  - Balance the errors involved (random walks, interpolation, PDE solving and roundoff) in practice
  - Extent to more general problems
  - ...

Synopsis and Prospects

- Potential for
  - Software reuse
  - Hardware utilization
  - Utilization of high performance PDE solvers and interpolants
  - Further experimentation
  - Huge speedups and practical use
- Questions need to be answered
  - Error analysis
  - Balance the errors involved (random walks, interpolation, PDE solving and roundoff) in practice
  - Extent to more general problems
  - ...

New line of reasoning that provides new intuition about the dynamics of PDE MC simulations.

https://github.com/mvavalis/Hybrid-numerical-PDE-solvers