A Fully Parallelized and Budgeted Multi-level Monte Carlo Framework for Partial Differential Equations

From Mathematical Theory to Automated Large-Scale Computations

Niklas Baumgarten, Christian Wieners, Sebastian Krumscheid | 28.02.2023

CRC 1173 Wave phenomena
Model Problem and Investigation Goals

**Acoustic wave:** Search \((v, p) : \Omega \times D \times [0, T] \rightarrow \mathbb{R}^{d+1}\),

such that

\[
\begin{align*}
\rho(\omega) \partial_t v(\omega) - \nabla p(\omega) &= f \quad D \times (0, T) \\
\partial_t p(\omega) - \text{div} (v(\omega)) &= g \quad D \times (0, T) \\
v \cdot n &= 0 \quad \Gamma \times (0, T) \\
v(0) &= v_0 \quad D \\
p(0) &= p_0 \quad D
\end{align*}
\]

**Determine:** \(E[Q] := \int_{\Omega} Q(\omega) d\mathbb{P} \approx M^{-1} \sum_{m=1}^{M} Q(\mathbf{y}^{(m)}) =: \hat{Q}_{\ell,M}^{MC}\)

**Goal:** Find combination of methods to minimize total error

\[ \text{err}_{\text{total}} = \text{err}_{\text{model}} + \text{err}_{\text{disc}} + \text{err}_{\text{input}} + \text{err}_{\text{solve}} + \text{err}_{\text{float}} + \text{err}_{\text{bug}} + \ldots \]

**Constraint:** Finite computational capacities (CPUs, time, memory)

⇒ Introduce budget for error minimization and utilize effective parallelization
Problem: Approximate $M$-times a PDE on discretization level $\ell$ on a fixed set of CPUs $\mathcal{P}$

Case: $M = 1$ & $|\mathcal{P}| = 1$

Minimize Communication: Search $k \in \mathbb{N}_0$, such that

$$2^k \leq \frac{\left| \mathcal{P} \right|}{M} < 2^{k+1} \Rightarrow \mathcal{P} = \bigsqcup_{m=1}^{M} \mathcal{P}_k^{(m)}$$

Define: Set of parallelised FE-Meshes

$$M_{\mathcal{P}} := \left\{ M_{\mathcal{P}_k^{(m)}} \right\}_{m=1}^{M}$$

Approximation: Search $\left( u_{\ell} \right)_{m=1}^{M} \in \prod_{m=1}^{M} V_{\ell}^{(m)}$ in basis build on $M_{\mathcal{P}}$ via multi-sample systems (MSS)
Introduction to Multi-level Monte Carlo

**Assumptions:** Let $\alpha, \beta, \gamma > 0$ and

\[
|E[Q_\ell - Q]| \lesssim h_\ell^\alpha \\
\forall[Q_\ell - Q_{\ell-1}] \lesssim h_\ell^\beta \\
C(Q_\ell(y^{(m)})) \lesssim h_\ell^{-\gamma}
\]

**Idea:** Telescoping sum with $Y_0 := Q_0$, $Y_\ell := Q_\ell - Q_{\ell-1}$

\[
E[Q_L] = E[Q_0] + \sum_{\ell=1}^{L} E[Q_\ell - Q_{\ell-1}] = \sum_{\ell=0}^{L} E[Y_\ell]
\]

**Multi-level Estimator:**

\[
\hat{Q}_L^{MLMC} = \sum_{\ell=0}^{L} \hat{Y}_{\ell,M_\ell} = \sum_{\ell=0}^{L} M_\ell^{-1} \sum_{m=1}^{M_\ell} Y_\ell(y^{(m)})
\]

**Epsilon-Cost Theorem:** $\exists \{M_\ell\}_{\ell=0}^{L}$, such that

\[
\text{err}_{\text{MSE}} = \sum_{\ell=0}^{L} M_\ell^{-1} V[Y_\ell] + (E[Q_L - Q])^2 < \epsilon^2
\]

and

\[
C_\epsilon \lesssim \begin{cases} 
\epsilon^{-2} & \beta > \gamma \\
\epsilon^{-2-(\gamma-\beta)/\alpha} & \beta < \gamma
\end{cases}
\]

Budgeted Multi-level Monte Carlo (BMLMC) Method

Goal: Replace accuracy $\epsilon$ by budget $B > 0$ measured in $[B] = \#\text{CPU} \cdot \text{hours}$

Motivation:
- Often no a priori knowledge about $\alpha$, $\beta$ and $\gamma$
- $B = |\mathcal{P}| \cdot T$ has to be reserved for HPC
- Empirical study of algorithms

Approximated Knapsack Problem:

$$\min_{(L,\{M_\ell\}_{\ell=0}^L)} \sum_{\ell=0}^L M_\ell^{-1} s_{Y_\ell}^2 + \left(\text{err}_{\text{disc}}(\hat{Y}_\ell, \hat{\alpha})\right)^2$$

such that $\sum_{\ell=0}^L M_\ell \hat{C}_\ell \leq B$

Convergence: For a feasible execution it holds

$$\epsilon \lesssim \begin{cases} B^{-1/2} & \beta > \gamma \\ B^{-1/2} \log(B)^{1/2} & \beta = \gamma \\ B^{-\alpha/(2\alpha+(\gamma-\beta))} & \beta < \gamma \end{cases}$$

Dynamic Programming (DP):
- Decomposition in overlapping subproblems
- Solve subproblems with optimal strategy
- Reutilization of preexisting results

$\Rightarrow$ Use DP for approximated knapsack problem
The Budgeted Multi-level Monte Carlo Algorithm

\[
data = \{ i \mapsto \{ \text{err}_i, \{ M_i, \ell \}_{\ell=0}^{L_i}, \{ \hat{C}_i, \ell \}_{\ell=0}^{L_i}, \{ \hat{Q}_i, \ell \}_{\ell=0}^{L_i}, \{ \hat{Y}_i, \ell \}_{\ell=0}^{L_i}, \ldots \} \}
\]

function BMLMC(B_0, \{ M_0^{\text{init}}, \ell \}_{\ell=0}^{L_0}) :

\[
\begin{array}{l}
\text{for } \ell = L_0, \ldots, 0: \\
\quad C_\ell, Q_\ell, Y_\ell \leftarrow \text{MC}(M_0^{\text{init}}, \ell) \leftarrow \text{MSS}(M_0^{\text{init}}, P) \\
\end{array}
\]

Update data[0]

return BMLMC(B_0 - \sum_{\ell=0}^{L_0} C_\ell, \eta \cdot \text{err}_0)

function BMLMC(B_i, \epsilon_i):

\[
\begin{array}{l}
\text{if } B_i \approx 0: \\
\quad \text{return } \hat{\text{err}}_{\text{RMSE}} \\
\text{if } \hat{\text{err}}_{\text{disc}}(\text{data}) \geq \sqrt{1 - \theta \epsilon_i}: \\
\quad L_i \leftarrow L_i + 1 \\
\text{if } \hat{\text{err}}_{\text{input}}(\text{data}) \geq \theta \epsilon_i^2: \\
\quad \Delta M_i, \ell \leftarrow \max \left\{ \hat{M}_i^{\text{opt}}(\text{data}) - M_{i-1, \ell}, 0 \right\} \\
\text{if } \hat{\text{err}}_{\text{MSE}} < \epsilon_i^2: \\
\quad \text{return } \text{BMLMC}(B_i, \eta \cdot \epsilon_i) \\
\text{for } \ell = L_i, \ldots, 0: \\
\quad C_\ell, Q_\ell, Y_\ell \leftarrow \text{MC}(\Delta M_i, \ell) \leftarrow \text{MSS}(\Delta M_i, \ell, P) \\
\quad \text{return } \text{BMLMC}(B_i - \sum_{\ell=0}^{L_i} C_\ell, \epsilon_i)
\end{array}
\]


Model Problem and Investigation Goals

**Acoustic wave**: Search \((v, p): \Omega \times D \times [0, T] \rightarrow \mathbb{R}^{d+1}\),

\[
\begin{align*}
\rho(\omega)\partial_t v(\omega) - \nabla p(\omega) &= f \quad D \times (0, T] \\
\partial_t p(\omega) - \text{div} (v(\omega)) &= g \quad D \times (0, T] \\
v \cdot n &= 0 \quad \Gamma \times (0, T] \\
v(0) &= v_0 \quad D \\
p(0) &= p_0 \quad D
\end{align*}
\]

such that

**Goal**: Find combination of methods to minimize total error

\[
\text{err}_{\text{total}} = \text{err}_{\text{model}} + \text{err}_{\text{disc}} + \text{err}_{\text{input}} + \text{err}_{\text{solve}} + \text{err}_{\text{float}} + \text{err}_{\text{bug}} + \ldots
\]

**Constraint**: Finite computational capacities (CPUs, time, memory)

⇒ Introduce budget for error minimization and utilize effective parallelization
Parallelization Experiments

Problem & Method Configuration:
- Log-normally distributed material density
- Implicit midpoint-rule (IMPR)
- Discontinuous Galerkin (dG)

1st Experiment:
- Solver parallelization vs. multi-sample systems (MSS) with fixed $B = 2048 \cdot 6h$

2nd Experiment:
- Weak scaling measurement with $T = 6h$ on $|\mathcal{P}| \in \{128, 512, 2048\}$
Empirical Method Comparison

Parallelization & Problem Configuration:
- Log-normally distributed material density
- MSS with \[ B = |\mathcal{P}| \cdot T = 1024 \cdot 6h \]

3rd Experiment:
- Polynomial degree \( p \in \{1, 2, 3\} \) of \( V^\ell_{\ell, p} \)

4th Experiment:
- Diagonal implicit Runge-Kutta (DIRK)
- Implicit midpoint-rule (IMPR)
- Crank-Nicolson (CN)
Conclusion and Outlook

Conclusion:
- Multi-mesh parallelization on distributed memory
- Budgeted multi-level Monte Carlo method
- Acoustic wave simulations in random media

Further Work:
- Other PDEs with various FE&UQ methods
- Automated high-performance computing (HPC) via CI/CD pipeline for good scientific practice

In Progress / Outlook / Interests:
- Publish PhD and Preprint
- Interfaces: Umbridge & Ginkgo
- SGD for Optimal Control
- (B)MLSC, (B)MLQMC, (B)MIMC
- Bayesian Inverse UQ

Man is a decision-making animal and, above all, conscious of this fact. This self-scrutiny has led to continuing efforts to find efficient ways of behaving in the face of complexity and uncertainty. - Richard Bellman, 1966
Load Distribution and Update-Rule

Load Distribution:

Layered Welford Update-Rule:

Properties:

- Heterogeneous sample- und solver parallelization
- Dynamic load distribution at run time
- Integrated SPMD implementation

Continuous Delivery (CD) as Data-Model Cycle

- **Development**
  - Model and Method refinement

- **Verification**
  - Unit, consistency and convergence testing

- **Validation**
  - Data-evaluation, benchmarking

- **Publication**
  - Research data-base, Persistent identifier

- **Simulation Data**
  - Measurement Data

- **Data Science**

- **Verification**
  - Model release, Persistent identifier

- **Validation**
  - Data-evaluation, Hyper-parameter tuning

Computational Science
The FEM-Library M++ with UQ Extension