Adaptive multi-fidelity polynomial chaos approaches to Bayesian inference in inverse problems

Tao ZHOU

NCMIS & Institute of Computational Mathematics, Chinese Academy of Sciences

Joint with Liang YAN (SEU)
Adaptive multi-fidelity polynomial chaos approaches to Bayesian inference in inverse problems

Tao ZHOU

NCMIS & Institute of Computational Mathematics, Chinese Academy of Sciences

Joint with Liang YAN (SEU)
Outline

1 Bayesian approach to inverse problems

2 Surrogate acceleration for Bayesian inference

3 Multi-fidelity surrogate modelling

4 Conclusions and references
Inverse problems

Consider a forward problem defined as follows:

\[ G(m) \approx d \]

- **m**: model parameters
- **G**: forward operator; **computationally intensive**!
- **d**: observed data

**Inverse problem**: Give a set of data **d**, estimate **m**.

- Problems typically **ill-posed** (existence, uniqueness, stability)
- Observations inevitably **noisy**, limited in number or resolution
Deterministic approaches

- Deterministic approach: regularization + optimization
- Regularization: impose smoothness; positivity; sparsity; etc
- Example: Tikhonov-type regularization [Tikhonov '1963]

\[
\text{Minimize} \quad J = \|G(m) - d\|^2 + \alpha \|\Omega(m)\|^2
\]

- Key issues:
  - How to choose $\Omega, \alpha$?
  - Yields a single value of $m$; how about the uncertainty/confidence?
Statistical approach

- **Key idea**: modeling $\mathbf{m}$ as random variables

Bayes’ rule

$$p(\mathbf{m}|\mathbf{d}) = \frac{p(\mathbf{d} | \mathbf{m}) p(\mathbf{m})}{p(\mathbf{d})} \propto L(\mathbf{d}, G(\mathbf{m})) p(\mathbf{m})$$

Notation

- parameters: $\mathbf{m} \in \mathbb{R}^d$; data $\mathbf{d} \in \mathbb{R}^n$
- $p(\mathbf{m})$ is the *prior* probability density
- $p(\mathbf{m} | \mathbf{d})$ is the *posterior* probability density
- $p(\mathbf{d})$ is the *evidence*, or equivalently, the *marginal likelihood*
- $L(\mathbf{d}, G(\mathbf{m})) \equiv p(\mathbf{d} | \mathbf{m})$ is the *likelihood function*
  - Each evaluation of $L$ requires an evaluation of $G$
Markov chain Monte Carlo (MCMC)

- **MCMC = the "computational engine" of Bayesian inference**
- Construct a Markov chain to sampling directly from **posterior**

**Features:**
- No normalization
- Automatic marginalization
- Posterior expectations = ergodic averages

\[
\mathbb{E}[f] \approx f_{n-b} \equiv \frac{1}{n-b} \sum_{t=b+1}^{n} f(m^t)
\]
Computational challenges

- **Computational issues:**
  - Forward models appear in the likelihood
  - Need to extract information from the posterior
  - Difficult to manipulate

- **MCMC** can be very useful, but
  - Posteriors are often **high-dimensional** (e.g., a discretized function)
  - Posterior evaluations are **expensive** (forward model evaluations)
  - Posteriors may contain **complex structure**

- **Accelerating** Bayesian inference for inverse problems:
  - Develop more efficient and robust alternatives to MCMC
  - Build a **cheap** surrogate model  \( \tilde{G}(m) \approx G(m) \) (this talk)
Surrogate-based Bayesian inference

- **Surrogate:** \( G(m) = \tilde{G}(m) \)

- **Approximate Bayesian rule:**

\[
\tilde{p}^d(m) \equiv \tilde{p}(m|d) \propto \tilde{L}(m)p(m)
\]

where \( \tilde{L}(m) = L(d, \tilde{G}(m)) \)

- **Advantages:**
  - Build offline, use online
  - No additional forward model simulations online
  - Allows direct sampling with arbitrarily large samples
Convergence Analysis

- **Basic assumption:** i.i.d. Gaussian observations, i.e. $\eta \sim N(0, \sigma^2I)$.

- **Kullback-Leibler divergence:** $D_{KL}(p_1\|p_2) = \int p_1(z) \log \frac{p_1(z)}{p_2(z)} dz$.

- **Uniform prior:**
  - $\Rightarrow D_{KL}(\tilde{p}^d\|p^d) \leq \sum_{i=1}^{n_d} \| G_i - \tilde{G}_i \|_{L^2_p}$. [Marzouk-Xiu ’09]
  - $\Rightarrow D_{KL}(\tilde{p}^d\|p^d) \leq \sum_{i=1}^{n_d} \| G_i - \tilde{G}_i \|_{L^2_p}^2$. [Yan-Guo ’15]

- **Generic prior:**
  - $\Rightarrow D_{KL}(\tilde{p}^d\|p^d) \leq \sum_{i=1}^{n_d} \| G_i - \tilde{G}_i \|_{L^2_p}^2$. [Yan-Zhang ’17]

- **Hellinger distance:** $D_{Hell}(p_1\|p_2) = \left(\frac{1}{2} \int \left( \sqrt{p_1(z)} - \sqrt{p_2(z)} \right)^2 dz \right)^{1/2}$

- **Generic prior:**
  - $\Rightarrow D_{Hell}(\tilde{p}^d\|p^d) \leq \sum_{i=1}^{n_d} \| G_i - \tilde{G}_i \|_{L^2_p}$. [Stuart-Teckentrup ’17]
Prior-based surrogate modelling

- Methods:
  - Projection-type reduced order models
    - proper orthogonal decomposition
    - reduced bases
  - Gaussian process regression
  - Polynomial chaos expansions
  - Deep learning
  - ...
Polynomial chaos (PC)

- Focus on dependence on $m$: $G(m) : \mathbb{R}^d \rightarrow \mathbb{R}$

- $P^\text{th}$-order PC expansion:
  \[ \tilde{G}(m) \in \mathbb{P}_d^P = \{ \text{d-dimensional polynomials of degree up to } P \} \]

- Orthogonal basis: $i = (i_1, \ldots, i_d), \|i\| = i_1 + \ldots + i_d$
  \[ \mathbb{E}[\phi_i(m)\phi_j(m)] = \int \phi_i(m)\phi_j(m)\rho(m)dm = \delta_{ij} \]

- Key issue: Efficiency & Curse-of-dimensionality

\[ \tilde{G}(m) = \sum_{k=1}^{M} c_k \phi_k(m), \quad M = \dim(\mathbb{P}_d^P) = \binom{d + P}{d} \]
Stochastic collocation methods

- **Algorithms:**
  - Choose a set of nodes: \( \{ m^{(k)} \}_{k=1}^{Q} \).
  - Deterministic system on grid points: \( G(m^{(k)}) = y^{(k)} \).
  - Construct the PC approximations using sample evaluations.

- **Matrix form:**
  \[
  Ac \approx y, \quad A \in \mathbb{R}^{Q \times M}, \quad A_{ij} = \phi_j(m^{(i)}).
  \]

- **Key issues:**
  - Nodal selection is critical to efficiency and accuracy
  - Approximation properties
• **Polynomial interpolation** \(\mathbf{Q} = \mathbf{M}: \mathbf{A}c = y\)
  - Tensor Lagrange interpolation: Babuska-Nobile-Tempone '07
  - Sparse grid interpolation: Xiu-Hasthaven '05, Nobile-Tempone-Webster '08
  - ...

• **Compressed sampling** \(\mathbf{Q} < \mathbf{M}: \min \|c\|_1\) subject to \(\mathbf{A}c = y\)
  - Random sampling: Doostan-Owhadi '11, Yan-Guo-Xiu '12
  - Structured random points: Tang-Iaccarino '14, Guo-Narayan-Zhou '17
  - Christoffel sampling: Narayan-Jackman-Zhou, '17
  - ...

• **Least-square projection** \(\mathbf{Q} > \mathbf{M}: c = \min \|\mathbf{A}c - y\|_2\)
  - Random sampling: Cohen-Davenport-Leviatan '13
  - Deterministic sampling: Zhou-Narayan-Xu '14
  - Christoffel Least-squares: Narayan-Jackman-Zhou '17
  - Approximated Fekete points: Guo-Narayan-Yan-Zhou '17
  - ...
Stochastic surrogate model in Bayesian inference

- Construct a PC approach for $m \sim \text{prior}$
- Propagate prior uncertainty through the forward problem

$$\tilde{G}(m) \approx \sum_{k=1}^{M} c_k \phi_k(m)$$

Result: $\tilde{G}(m)$, a stochastic spectral representation of forward model predictions. (compute $\{c_k\}$ only once)

- Use this as a surrogate for the forward model in the likelihood function – no repeated forward problem simulations!
- MCMC + surrogate can be very efficient, but how about the accuracy?
An illustrated example

- Elliptic PDE in two spatial dimensions:
  \[-\nabla \cdot (p(x) \nabla u(x)) = f(x)\]

- \( p(x) = \sum_{i=1}^{9} \kappa_i \exp(-0.5 \frac{||x-x_{0,i}||^2}{0.15^2}); \quad f(x) = 100 \sin(\pi x_1) \sin(\pi x_2). \)

- Log-normal prior on \( \log(\kappa_i) \sim N(0, 1); \)

- The true weight is drawn from \( \log(\kappa_i) \sim U(-5, 5) \)
Left: Direct (1492.2s). Middle: PC ($N = 7; 241.2s$). Right: PC ($N = 2; 11.4s$). From top to bottom, the relative noise level $\delta$ is 0.01 and 0.05, respectively.
## Simulation results

Computational times, in seconds, by two different methods.

<table>
<thead>
<tr>
<th>Method</th>
<th># of model evaluation</th>
<th>CPU(s)</th>
<th>Offline</th>
<th># of model evaluation</th>
<th>CPU(s)</th>
<th>Online</th>
<th>Total time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
<td></td>
<td>5×10⁴</td>
<td>1492.2</td>
</tr>
<tr>
<td>PC, (N = 7)</td>
<td>28,880</td>
<td>1246.7</td>
<td></td>
<td></td>
<td></td>
<td>–</td>
<td>241.2</td>
</tr>
<tr>
<td>PC, (N = 6)</td>
<td>10,010</td>
<td>340.4</td>
<td></td>
<td></td>
<td></td>
<td>–</td>
<td>121.7</td>
</tr>
<tr>
<td>PC, (N = 4)</td>
<td>1,430</td>
<td>37.5</td>
<td></td>
<td></td>
<td></td>
<td>–</td>
<td>39.1</td>
</tr>
<tr>
<td>PC, (N = 2)</td>
<td>110</td>
<td>2.9</td>
<td></td>
<td></td>
<td></td>
<td>–</td>
<td>11.4</td>
</tr>
</tbody>
</table>
Multi-fidelity? [e.g. Peherstorfer-Willcox-Gunzburger SIREV’18]

- Full model $u^H$ ("truth"): **Computationally expensive**
- Surrogate model $u^L$ ("approximate"): **Computationally cheap**
- Multifidelity model:
  - invoke multiple modes to reduce computational cost
  - enhance the recovery accuracy

**Key questions:**
- how to combine model estimates?
- how to balance evaluations among models?
- how to guarantee the accuracy?
Multi-fidelity PC based surrogate

• Full model ("truth"): \( u^H = G(z) \)

• Surrogate model ("approximate") : \( u^L = \sum_{i=1}^{P} \alpha_i \psi(z) \)

• Correction term: \( C(z) = u^H(z) - u^L(z) \)

\[
  u^H(z) = u^L(z) + C(z)
\]

• Multi-fidelity expansion:

\[
  u^M(z) = u^L(z) + \sum_{j=1}^{Q} \beta_j \psi(z)
\]

▷ Calculate the difference between the full and surrogate model
▷ Solve the linear equations to obtain \( \beta_j \)
Given $z_0$, simulate chain $\{X_t\}$ with (symmetric) translation invariant density $q$.

**Algorithm**

```
for $n = 1, \cdots, l_{\text{max}}$ do
  ▶ Draw $m-1$ samples $\{z_1, \cdots, z_{m-1}\}$ from the approximate probability density posterior based on $u^L$. Propose $z^* \sim q(\cdot|z_{m-1})$.
  ▶ If the surrogate needs refinement near $z^*$ or $z_{m-1}$, then select new points around $z^*$ or $z_{m-1}$ to construct a new multi-fidelity surrogate $u^M$. Set $u^L = u^M$.
  ▶ Compute acceptance probability

$$
\alpha = \min \left\{ 1, \frac{L(d, u^L(z^*)) p(z^*)}{L(d, u^L(z_{m-1})) p(z_{m-1})} \right\}
$$

  ▶ Draw $\theta \sim U(0, 1)$. If $\theta < \alpha$, let $z_m = z^*$, otherwise $z_m = z_{m-1}$.
  ▶ Let $z_0 = z_m$ and $X_n = X_{n-1} \cup \{z_1, \cdots, z_m\}$
endfor
```
Update the multi-fidelity PC surrogate

Algorithm: update the multi-fidelity PC surrogate

- Compute acceptance probability using high-fidelity model $u^H$

$$\beta = \min \left\{ 1, \frac{L(d, u^H(z^*))p(z^*)}{L(d, u^H(z_{m-1}))p(z_{m-1})} \right\}$$

- Draw $\theta \sim U(0, 1)$. If $\theta < \beta$, let $y = z^*$, otherwise $y = z_{m-1}$.

- Compute the relative error

$$err = \| u^H(y) - u^L(y) \|_\infty$$

- if $err > \epsilon$ then Select random points locally in $B(y, R)$ and construct a new multi-fidelity model $u^M$. 

T.Zhou (CAS, China)  Multi-fidelity Bayesian inference  Joint with Liang YAN (SEU)
Theorem [Yan-Zhou ’18]

Suppose we have the true posterior distribution \( p^d \) and its approximation \( \tilde{p}^d \) induced by the adaptive surrogate model. For a given tolerance \( \epsilon > 0 \), there exist constants \( c_1 > 0 \) and \( c_2 > 0 \) such that

\[
D_{KL}(\tilde{p}^d\|p^d) \leq \left( c_1 \epsilon + c_2 \mu(\Omega_N^N(\epsilon)) \right)^2.
\]

- \( \epsilon \)-feasible set: \( \Omega_N^N(\epsilon) = \left\{ y \in \Gamma : \|u^H(y) - u^L(y)\|_{\infty} \leq \epsilon \right\} \)

- posterior measure: \( \mu(\Omega_N^N(\epsilon)) = \int_{\Omega_N^N(\epsilon)} p^d dz \)

- complement of the \( \epsilon \)-feasible set: \( \Omega_N^N(\epsilon) = \Gamma \setminus \Omega_N^N(\epsilon) \)

- **posterior measure**: \( \mu(\Omega_N^N(\epsilon)) = 1 - \mu(\Omega_N^N(\epsilon)) \)
Example: Elliptic PDEs

Left: Direct (1492.2s). Middle: AMPC ($N = 4; 49.2s$). Right: AMPC ($N = 2; 38.7s$). From top to bottom, the relative noise level $\delta$ is 0.01 and 0.05, respectively.
Example: Elliptic PDEs

Computational times, in seconds, given by three different methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Offline # of model evaluation</th>
<th>Offline CPU(s)</th>
<th>Online # of model evaluation</th>
<th>Online CPU(s)</th>
<th>Total time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct</td>
<td>–</td>
<td>–</td>
<td>5×10^4</td>
<td>1492.2</td>
<td>1492.2</td>
</tr>
<tr>
<td>PC, $N = 7$</td>
<td>28,880</td>
<td>1246.7</td>
<td>–</td>
<td>241.2</td>
<td>1487.9</td>
</tr>
<tr>
<td>PC, $N = 6$</td>
<td>10,010</td>
<td>340.4</td>
<td>–</td>
<td>121.7</td>
<td>426.1</td>
</tr>
<tr>
<td>PC, $N = 4$</td>
<td>1,430</td>
<td>37.5</td>
<td>–</td>
<td>39.1</td>
<td>76.6</td>
</tr>
<tr>
<td>PC, $N = 2$</td>
<td>110</td>
<td>2.9</td>
<td>–</td>
<td>11.4</td>
<td>14.3</td>
</tr>
<tr>
<td>AMPC, $N_L = 4, N_C = 2$</td>
<td>1,430</td>
<td>37.5</td>
<td>570</td>
<td>49.2</td>
<td>86.7</td>
</tr>
<tr>
<td>AMPC, $N_L = 2, N_C = 2$</td>
<td>110</td>
<td>2.9</td>
<td>1,010</td>
<td>38.7</td>
<td>41.6</td>
</tr>
</tbody>
</table>
Example: Source Inversion

\[ cD_t^\alpha u - \nabla^2 u = \frac{s}{2\pi\sigma^2} e^{-t} \exp\left(-\frac{|\chi - x|^2}{2\sigma^2}\right) \]

\[ \nabla u \cdot n = 0 \]

\[ u(x, 0) = 0 \]

- \( s = 0.5, \sigma = 0.1; \chi = (Z_1, Z_2) \)
- Measurement error is additive Gaussian
- Priors: \( m = (Z_1, Z_2) \sim U(0, l) \)
Example: Source Inversion: $\sigma = 0.2$

- Model evaluation points for constructing the prior-based (Left) and multi-fidelity surrogate (Right).
Example: Source Inversion: $\sigma = 0.2$

- Computational times: PC: $17.3s(N = 9)$; $12.2s(N = 6)$; $7.7s(N = 3)$. Direct: $4424.6s \approx 1.2h!$
Example: Source Inversion: $\sigma = 0.2$

- Computational times: AMPC: $14.5s(N = 6), 10.3s(N = 3)$. Direct: $4424.6s \approx 1.2h$!
Ensemble Kalman Filter (EnKF)

Sequential signal-observation system

Signal: \( x_{j+1} = G(x_j) + \xi_j, \quad j \in \mathbb{Z}^+ \)

Observation: \( y_{j+1} = H(x_{j+1}) + \eta_{j+1} \quad j \in \mathbb{Z}^+ \)

- EnKF is widely used in variance applications.
- Origins in sequential data assimilation [Evensen '94]
- Application to inverse problem [Iglesias-Law-Stuart '16]
- Requires only black box forward model \( G \)

T.Zhou (CAS, China)  Multi-fidelity Bayesian inference  Joint with Liang YAN (SEU)
Algorithm 1: Basic EKI for \( y = G(z) + \eta \)

- **Initial Ensemble:** \( z_0^{(j)} \sim \pi(z), \quad j = 1, \cdots, N_e. \)

- **Prediction step:**
  
  **Means:**
  \[
  \bar{z}_n = \frac{1}{N_e} \sum_{j=1}^{N_e} z_n^{(j)}, \quad \bar{\omega}_n = \frac{1}{N_e} \sum_{j=1}^{N_e} G(z_n^{(j)}). 
  \]

  **Covariances:**
  \[
  C_n^{z\omega} = \frac{1}{N_e - 1} \sum_{j=1}^{N_e} (z_n^{(j)} - \bar{z}_n)(G(z_n^{(j)}) - \bar{\omega}_n)^T 
  \]
  \[
  C_n^{\omega\omega} = \frac{1}{N_e - 1} \sum_{j=1}^{N_e} (G(z_n^{(j)}) - \bar{\omega}_n)(G(z_n^{(j)}) - \bar{\omega}_n)^T. 
  \]

- **Update step**
  \[
  z_{n+1}^{(j)} = z_n^{(j)} + C_n^{z\omega} (C_n^{\omega\omega} + \Gamma)^{-1} (y - G(z_n^{(j)}), \quad j = 1, \cdots, N_e. 
  \]
Algorithm

▶ **Step 1: Initialization**: Build the prior-based PC surrogate $f^L$. Generate $N_e$ parameter realizations from the prior distribution as the initial ensemble.

▶ **Step 2: At the $n$-th iteration step**, generate the system outputs for the ensemble realizations with the surrogate $f^L$.

$$
\bar{z}_n = \frac{1}{N_e} \sum_{j=1}^{N_e} z_n^{(j)}, \quad \bar{\omega}_n = \frac{1}{N_e} \sum_{j=1}^{N_e} f^L(z_n^{(j)}).
$$

Update the parameter ensemble with the EKI formula, i.e., Algorithm 1.

▶ **Step 3**: Compute the ensemble mean $\bar{z}_{n+1} = \frac{1}{N_e} \sum_{j=1}^{N_e} z_{n+1}^{(j)}$.

*If approximation needs refinement* near $\bar{z}_{n+1}$, then select new points to construct the multi-fidelity model $f^M$. Refine $f^L = f^M$.

▶ **Step 4**: Repeat Steps 2-3 until one of the stop criteria of EKI is met.
Example: Inverse fractional diffusion problems

\[ cD_t^\alpha u - \nabla \cdot (\rho(x) \nabla u(x, t)) = f(x, t) \quad (x, t) \in \Omega \times (0, 1) \]

- \( \rho(x) = \sum_{i=1}^{9} \kappa_i \exp(-0.5 \frac{\|x-x_{0,i}\|^2}{0.15^2}) \);
- Log-normal prior on \( \log(\kappa_i) \sim N(0, 1) \);
- The true weight is drawn from \( \log(\kappa_i) \sim U(-4, 4) \)

Left: the true permeability. Right: the initial ensemble mean.
Numerical results for the final iteration using $N_e = 100$. (Left) Direct (56.71s); (Middle) PC (N=4: 40.25s+0.82s); (Right) PC (N=6: 336.59s+3.15s).
Numerical results for the final iteration using $N_e = 100$: (Left) Direct (56.71s); (Middle) AMPC ($N=2$, $tol = 1 \times 10^{-2}$: 3.69s+6.28s); (Right) AMPC ($N=2$, $tol = 1 \times 10^{-3}$: 3.69s+10.92s).
Numerical results with different ensemble size. Left: online CPU time. Right: number of required adaptively.
Conclusions

• **Multi-fidelity surrogate for Bayesian inverse problems:**
  ▶ invoke multiple models to reduce computational cost.
  ▶ maintain the recovery accuracy.

• **Extension and ongoing work:**
  ▶ Dimensionality reduction for inference
  ▶ Hierarchical Bayesian Inversion
  ▶ Hierarchical EnKF Inversion

• **References:**

Thank you!