SAND2017-7324C

A Scalable Sampling-Free Nonlinear State Estimation Algorithm for Large-Scale Models and Data Sets using High Performance Computing

Ajit Desai [1], Philippe Bisaillon [1], Mohammad Khalil [2], Chris Pettit [3], Dominique Poirel [4] and Abhijit Sarkar [1]

[1]Department of Civil and Environmental Engineering
Carleton University, Canada
[2]Quantitative Modeling & Analysis
Sandia National Laboratories, Livermore, California 94551, USA
[3]Aerospace Engineering Department
United States Naval Academy, Annapolis, Maryland, USA
[4]Department of Mechanical and Aerospace Engineering
Royal Military College of Canada

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

June 19, 2016

- Motivation
  - Data assimilation for high resolution numerical models.
- Objective
  - Develop scalable parallel algorithms for sequential data assimilation.

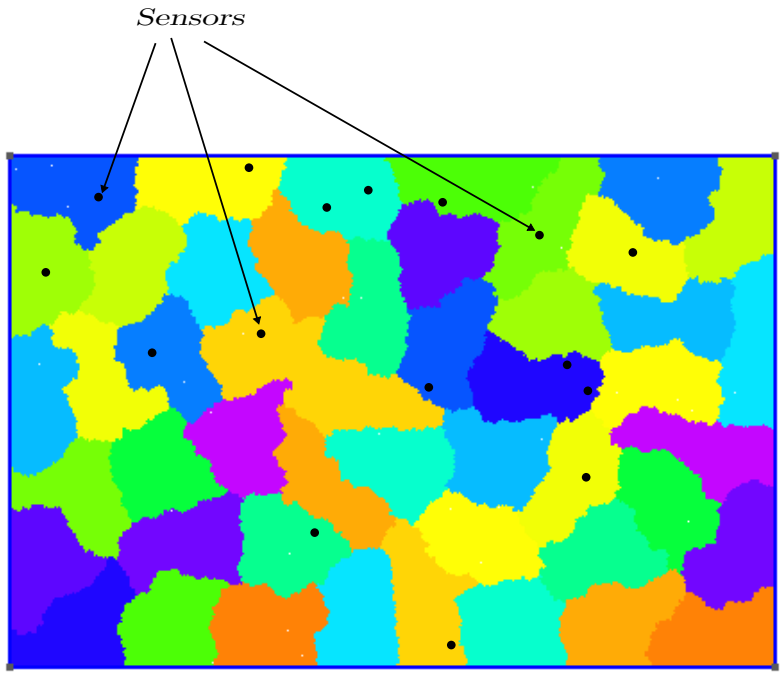
    Scalability: Solve n-times larger problem using n-times more processors/cores without substantially increasing the execution time.
- Methodology
  - Exploit scalable intrusive polynomial chaos expansion-based non-overlapping domain decomposition for distributed implementation of data assimilation algorithms.

- Model Equation

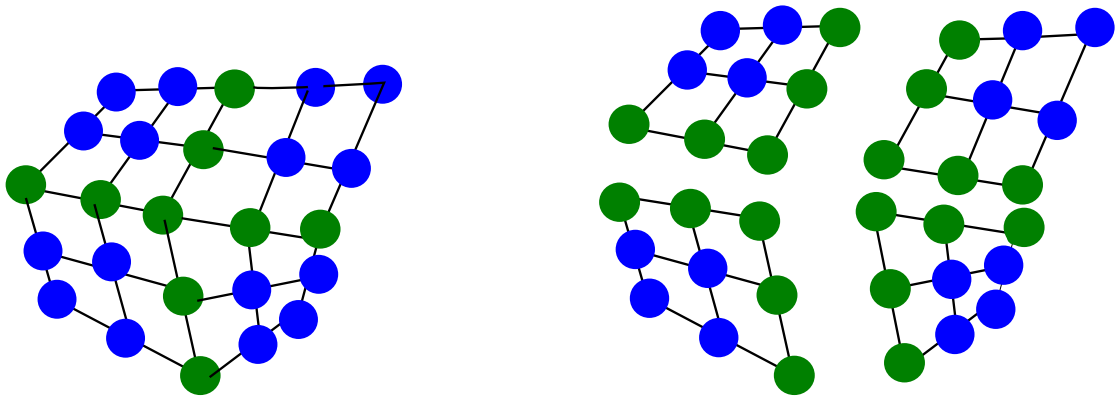
$$\mathbf{u}_{k+1} = \boldsymbol{\psi}_k\left(\mathbf{u}_k, \mathbf{f}_k, \mathbf{q}_k\right) \quad -- \text{ Forecast Step}$$

- Measurement Equation

$$\mathbf{d}_k = \mathbf{h}_k\left(\mathbf{u}_k, \boldsymbol{\epsilon}_k\right) \qquad -- \text{ Assimilation Step}$$



(*Forecast Step*)



- Spatial decomposition

$$\begin{bmatrix} \mathbf{A}^s_{II}(\theta) & \mathbf{A}^s_{I\Gamma}(\theta) \\ \mathbf{A}^s_{\Gamma I}(\theta) & \mathbf{A}^s_{\Gamma\Gamma}(\theta) \end{bmatrix} \begin{Bmatrix} \mathbf{u}^s_I(\theta) \\ \mathbf{u}^s_\Gamma(\theta) \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}^s_I \\ \mathbf{f}^s_\Gamma \end{Bmatrix} .$$

- Polynomial Chaos expansion

$$\sum_{i=0}^{L} \Psi_i \begin{bmatrix} \mathbf{A}^s_{II,i} & \mathbf{A}^s_{I\Gamma,i} \\ \mathbf{A}^s_{\Gamma I,i} & \mathbf{A}^s_{\Gamma\Gamma,i} \end{bmatrix} \begin{Bmatrix} \mathbf{u}^s_I(\theta) \\ \mathbf{u}^s_\Gamma(\theta) \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}^s_I \\ \mathbf{f}^s_\Gamma \end{Bmatrix} .$$

- Galerkin projection

$$\begin{bmatrix} \mathcal{A}_{II}^1 & \dots & 0 & \mathcal{A}_{I\Gamma}^1 \mathcal{R}_1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & \mathcal{A}_{II}^{n_s} & \mathcal{A}_{I\Gamma}^{n_s} \mathcal{R}_{n_s} \\ \mathcal{R}_1^T \mathcal{A}_{\Gamma I}^1 & \dots & \mathcal{R}_{n_s}^T \mathcal{A}_{\Gamma I}^{n_s} & \sum_{s=1}^{n_s} \mathcal{R}_s^T \mathcal{A}_{\Gamma\Gamma}^s \mathcal{R}_s \end{bmatrix} \begin{Bmatrix} \mathcal{U}_I^1 \\ \vdots \\ \mathcal{U}_I^{n_s} \\ \mathcal{U}_\Gamma \end{Bmatrix} = \begin{Bmatrix} \mathcal{F}_I^1 \\ \vdots \\ \mathcal{F}_I^{n_s} \\ \sum_{s=1}^{n_s} \mathcal{R}_s^T \mathcal{F}_\Gamma^s \end{Bmatrix},$$

where

$$[\mathcal{A}_{\alpha\beta}^s]_{jk} = \sum_{i=0}^{L} \langle \Psi_i \Psi_j \Psi_k \rangle \mathbf{A}_{\alpha\beta,i}^s \quad , \quad \mathcal{F}_{\alpha,k}^s = \langle \Psi_k \mathbf{f}_\alpha^s \rangle.$$

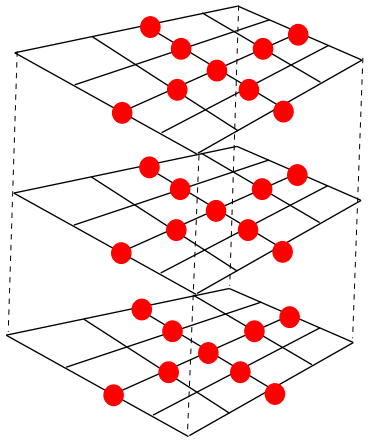
Sarkar, A. Benabbou, N. and Ghanem, R., IJNME, 2009.

- The Extended Schur Complement System

$$\mathcal{S}\mathcal{U}_\Gamma = \mathcal{G}_\Gamma.$$

$$\mathcal{S} = \sum_{s=1}^{n_s} \mathcal{R}_s^T [\mathcal{A}_{\Gamma\Gamma}^s - \mathcal{A}_{\Gamma I}^s (\mathcal{A}_{II}^s)^{-1} \mathcal{A}_{I\Gamma}^s] \mathcal{R}_s.$$

  - Develope parallel iterative algorithms.
  - Formulate scalable preconditioners.
  - Application to 2D and 3D Stochastic PDEs with non-Gaussian coefficients.



$$\mathcal{S}\,\mathcal{U}_\Gamma = \mathcal{G}_\Gamma.$$

$$\mathcal{M}^{-1}\mathcal{S}\,\mathcal{U}_\Gamma = \mathcal{M}^{-1}\mathcal{G}_\Gamma.$$

- The preconditioner $(\mathcal{M}^{-1})$ is a good approximation to $(\mathcal{S}^{-1})$.
- The condition number of $(\mathcal{M}^{-1}\mathcal{S})$ is much smaller than $(\mathcal{S})$.

Parallel Implementation

1. $\mathcal{U}_{\Gamma_0 := 0}$
2. *Gather* $\mathbf{r}_0 := \sum_{s=1}^{n_s} \mathcal{R}_s^T \mathcal{G}_\Gamma^s$
3. *Scatter* $\mathbf{r}_0$
4. *Gather* $\mathcal{Z}_0 := \sum_{s=1}^{n_s} \mathcal{R}_s^T \mathcal{M}_s^{-1} \mathcal{R}_s \mathbf{r}_0$
5. $\mathcal{P}_0 := \mathcal{Z}_0$
6. $\rho_0 := (\mathbf{r}_0, \mathcal{Z}_0)$
7. *For* $j = 0, 1, \cdots$, *until convergence Do*
8. *Scatter* $\mathcal{P}_j$
9. *Gather*

$$\mathcal{Q}_j := \sum_{s=1}^{n_s} \mathcal{R}_s^T \mathcal{S}_s \mathcal{R}_s \mathcal{P}_j$$

10. $\rho_{tmp} := (\mathcal{P}_j, \mathcal{Q}_j)$
11. $\alpha := \rho_j / \rho_{tmp}$
12. $\mathcal{U}_{\Gamma_{j+1}} := \mathcal{U}_{\Gamma_j} + \alpha \mathcal{P}_j$
13. $\mathbf{r}_{j+1} := \mathbf{r}_j - \alpha \mathcal{Q}_j$
14. *Scatter* $\mathbf{r}_{j+1}$
15. *Gather*

$$\mathcal{Z}_{j+1} := \sum_{s=1}^{n_s} \mathcal{R}_s^T \mathcal{M}_s^{-1} \mathcal{R}_s \mathbf{r}_{j+1}$$

16. $\rho_{j+1} := (\mathbf{r}_{j+1}, \mathcal{Z}_{j+1})$
17. $\beta_j := \rho_{j+1} / \rho_j$
18. $\mathcal{P}_{j+1} = \mathcal{P}_j + \beta_j \mathcal{Z}_{j+1}$
19. *EndDo*

Parallel Implementation

1. $\mathcal{U}_{\Gamma_0 := 0}$
2. *Gather* $\mathbf{r}_0 := \sum_{s=1}^{n_s} \mathcal{R}_s^T \mathcal{G}_\Gamma^s$
3. *Scatter* $\mathbf{r}_0$
4. *Gather* $\mathcal{Z}_0 := \sum_{s=1}^{n_s} \mathcal{R}_s^T \mathcal{M}_s^{-1} \mathcal{R}_s \mathbf{r}_0$
5. $\mathcal{P}_0 := \mathcal{Z}_0$
6. $\rho_0 := (\mathbf{r}_0, \mathcal{Z}_0)$
7. *For* $j = 0, 1, \cdots$, *until convergence Do*
8. *Scatter* $\mathcal{P}_j$
9. *Gather*

$$\mathcal{Q}_j := \sum_{s=1}^{n_s} \mathcal{R}_s^T \mathcal{S}_s \mathcal{R}_s \mathcal{P}_j$$

10. $\rho_{tmp} := (\mathcal{P}_j, \mathcal{Q}_j)$
11. $\alpha := \rho_j / \rho_{tmp}$
12. $\mathcal{U}_{\Gamma_{j+1}} := \mathcal{U}_{\Gamma_j} + \alpha \mathcal{P}_j$
13. $\mathbf{r}_{j+1} := \mathbf{r}_j - \alpha \mathcal{Q}_j$
14. *Scatter* $\mathbf{r}_{j+1}$
15. *Gather*

$$\mathcal{Z}_{j+1} := \sum_{s=1}^{n_s} \mathcal{R}_s^T \mathcal{M}_s^{-1} \mathcal{R}_s \mathbf{r}_{j+1}$$

16. $\rho_{j+1} := (\mathbf{r}_{j+1}, \mathcal{Z}_{j+1})$
17. $\beta_j := \rho_{j+1} / \rho_j$
18. $\mathcal{P}_{j+1} = \mathcal{P}_j + \beta_j \mathcal{Z}_{j+1}$
19. *EndDo*

- Extended Lumped Preconditioner

$$\mathcal{M}_L^{-1} = \sum_{s=1}^{n_s} \mathcal{R}_s^T [\mathcal{A}_{\Gamma\Gamma}^s]^{-1} \mathcal{R}_s.$$

- Extended Weighted Lumped Preconditioner

$$\mathcal{M}_{WL}^{-1} = \sum_{s=1}^{n_s} \mathcal{R}_s^T \mathcal{D}_s^T [\mathcal{A}_{\Gamma\Gamma}^s]^{-1} \mathcal{D}_s \mathcal{R}_s.$$

- Extended Neumann-Neumann Preconditioner

$$\mathcal{M}_{NN}^{-1} = \sum_{s=1}^{n_s} \mathcal{R}_s^T \mathcal{D}_s^T \mathcal{S}_s^{-1} \mathcal{D}_s \mathcal{R}_s.$$

  - Preconditioner effect

$$\begin{bmatrix} \mathcal{A}_{II}^s & \mathcal{A}_{I\Gamma}^s \\ \mathcal{A}_{\Gamma I}^s & \mathcal{A}_{\Gamma\Gamma}^s \end{bmatrix} \begin{Bmatrix} \mathcal{X}^s \\ \mathcal{U}_\Gamma^s \end{Bmatrix} = \begin{Bmatrix} 0 \\ r_\Gamma^s \end{Bmatrix}.$$

- Condition Number Bound of Deterministic System
  - Stiffness matrix of elliptic PDE

$$\kappa(A) \leq C\frac{1}{h^2}$$

  - Schur complement matrix

$$\kappa(S) \leq C\frac{1}{hH}$$

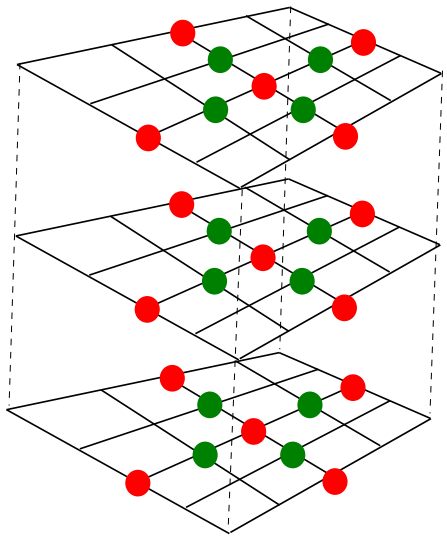
  - One-level preconditioner

$$\kappa(M^{-1}S) \leq C\frac{1}{H^2}(1+\log\frac{H}{h})^2$$

  - Two-level preconditioner

$$\kappa(M^{-1}S) \leq C(1+\log\frac{H}{h})^2$$

- Partitioning the interface nodes into remaining (■) and corner(●) nodes



$$u_\Gamma^s = \begin{Bmatrix} u_r^s \\ u_c^s \end{Bmatrix}$$

$$\begin{bmatrix} \mathcal{A}_{ii}^s & \mathcal{A}_{ir}^s & \mathcal{A}_{ic}^s\mathcal{B}_c^s \\ \mathcal{A}_{ri}^s & \mathcal{A}_{rr}^s & \mathcal{A}_{rc}^s\mathcal{B}_c^s \\ \sum_{s=1}^{n_s}\mathcal{B}_c^{s\,T}\mathcal{A}_{ci}^s & \sum_{s=1}^{n_s}\mathcal{B}_c^{s\,T}\mathcal{A}_{cr}^s & \sum_{s=1}^{n_s}\mathcal{B}_c^{s\,T}\mathcal{A}_{cc}^s\mathcal{B}_c^s \end{bmatrix} \begin{Bmatrix} \mathcal{X}^s \\ \mathcal{U}_r^s \\ \mathcal{U}_c \end{Bmatrix} = \begin{Bmatrix} \mathbf{0} \\ \mathcal{F}_r^s \\ \sum_{s=1}^{n_s}\mathcal{B}_c^{s\,T}\mathcal{F}_c^s \end{Bmatrix}$$

$$\boxed{\mathcal{M}_{NNC}^{-1} = \sum_{s=1}^{n_s}\mathcal{R}_s^T\mathcal{D}_s\mathcal{R}_s^{r\,T}[\mathcal{S}_{rr}^s]^{-1}\mathcal{R}_s^r\mathcal{D}_s\mathcal{R}_s + R_0^T[F_{cc}]^{-1}R_0,}$$

$$R_0 = \sum_{s=1}^{n_s}\mathcal{B}_c^{s\,T}(\mathcal{R}_s^c - \mathcal{S}_{cr}^s[\mathcal{S}_{rr}^s]^{-1}\mathcal{R}_s^r)\mathcal{D}_s\mathcal{R}_s.$$

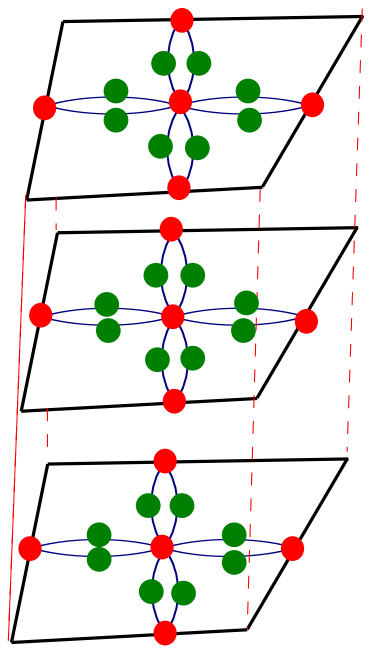
$$F_{cc} = \sum_{s=1}^{n_s}\mathcal{B}_c^{s\,T}(\mathcal{S}_{cc}^s - \mathcal{S}_{cr}^s[\mathcal{S}_{rr}^s]^{-1}\mathcal{S}_{rc}^s)\mathcal{B}_c^s,$$

$$\mathcal{S}_{\alpha\beta}^s = \mathcal{A}_{\alpha\beta}^s - \mathcal{A}_{\alpha i}^s[\mathcal{A}_{ii}^s]^{-1}\mathcal{A}_{i\beta}^s,\ [\mathcal{A}_{\alpha\beta}^s]_{jk} = \sum_{i=0}^{L}\langle\Psi_i\Psi_j\Psi_k\rangle\mathbf{A}_{\alpha\beta,i}^s$$

Probabilistic Dual Primal Domain Decomposition

$$\begin{bmatrix} \mathcal{A}_{ii}^s & \mathcal{A}_{ir}^s & \mathcal{A}_{ic}^s \mathcal{B}_c^s & 0 \\ \mathcal{A}_{ri}^s & \mathcal{A}_{rr}^s & \mathcal{A}_{rc}^s \mathcal{B}_c^s & \mathcal{B}_r^{s\,T} \\ \sum_{s=1}^{n_s} \mathcal{B}_c^{s\,T} \mathcal{A}_{ci}^s & \sum_{s=1}^{n_s} \mathcal{B}_c^{s\,T} \mathcal{A}_{cr}^s & \sum_{s=1}^{n_s} \mathcal{B}_c^{s\,T} \mathcal{A}_{cc}^s \mathcal{B}_c^s & 0 \\ 0 & \sum_{s=1}^{n_s} \mathcal{B}_r^s & 0 & 0 \end{bmatrix} \begin{Bmatrix} \mathcal{U}_i^s \\ \mathcal{U}_r^s \\ \mathcal{U}_c \\ \Lambda \end{Bmatrix} = \begin{Bmatrix} \mathcal{F}_i^s \\ \mathcal{F}_r^s \\ \sum_{s=1}^{n_s} \mathcal{B}_c^{s\,T} \mathcal{F}_c^s \\ 0 \end{Bmatrix},$$

$$(\bar{F}_{rr} + \bar{F}_{rc}[\bar{F}_{cc}]^{-1}\bar{F}_{cr})\Lambda = \bar{d}_r - \bar{F}_{rc}[\bar{F}_{cc}]^{-1}\bar{d}_c,$$



$$\mathcal{M}^{-1} = \sum_{s=1}^{n_s} \mathcal{H}_f^{s\,T} [\mathcal{S}_f^s]^{-1} \mathcal{H}_f^s + \mathcal{H}_0^T [\mathcal{S}_c]^{-1} \mathcal{H}_0,$$

	$\mathcal{H}_f^s$	$[\mathcal{S}_f^s]^{-1}$	$\mathcal{H}_0$	$[\mathcal{S}_c]^{-1}$
a	$\mathcal{R}_s^r \mathcal{D}_s \mathcal{R}_s$	$[\mathcal{S}_{rr}^s]^{-1}$	$\sum_{s=1}^{n_s} \mathcal{B}_c^{s\,T} (\mathcal{R}_s^c - \mathcal{S}_{cr}^s [\mathcal{S}_{rr}^s]^{-1} \mathcal{R}_s^r) \mathcal{D}_s \mathcal{R}_s$	$\sum_{s=1}^{n_s} \mathcal{B}_c^{s\,T} (\mathcal{S}_{cc}^s - \mathcal{S}_{cr}^s \left[\mathcal{S}_{rr}^s\right]^{-1} \mathcal{S}_{rc}^s) \mathcal{B}_c^s$
b	$\mathcal{D}_r^s \mathcal{B}_r^s$	$\left[\mathcal{S}_{rr}^s\right]^{-1}$	$\sum_{s=1}^{n_s} \mathcal{B}_c^{s\,T} \mathcal{S}_{cr} \left[\mathcal{S}_{rr}^s\right]^{-1} \mathcal{D}_r^s \mathcal{B}_r^s$	$\sum_{s=1}^{n_s} \mathcal{B}_c^{s\,T} (\mathcal{S}_{cc}^s - \mathcal{S}_{cr}^s \left[\mathcal{S}_{rr}^s\right]^{-1} \mathcal{S}_{rc}^s) \mathcal{B}_c^s$
c	$\mathcal{B}_r^{s\,T}$	$[\mathcal{S}_{rr}^s]^{-1}$	$\sum_{s=1}^{n_s} \mathcal{B}_c^{s\,T} \mathcal{S}_{cr}^s [\mathcal{S}_{rr}^s]^{-1} \mathcal{B}_r^{s\,T}$	$\sum_{s=1}^{n_s} \mathcal{B}_c^{s\,T} (\mathcal{S}_{cc}^s - \mathcal{S}_{cr}^s [\mathcal{S}_{rr}^s]^{-1} \mathcal{S}_{rc}^s) \mathcal{B}_c^s$

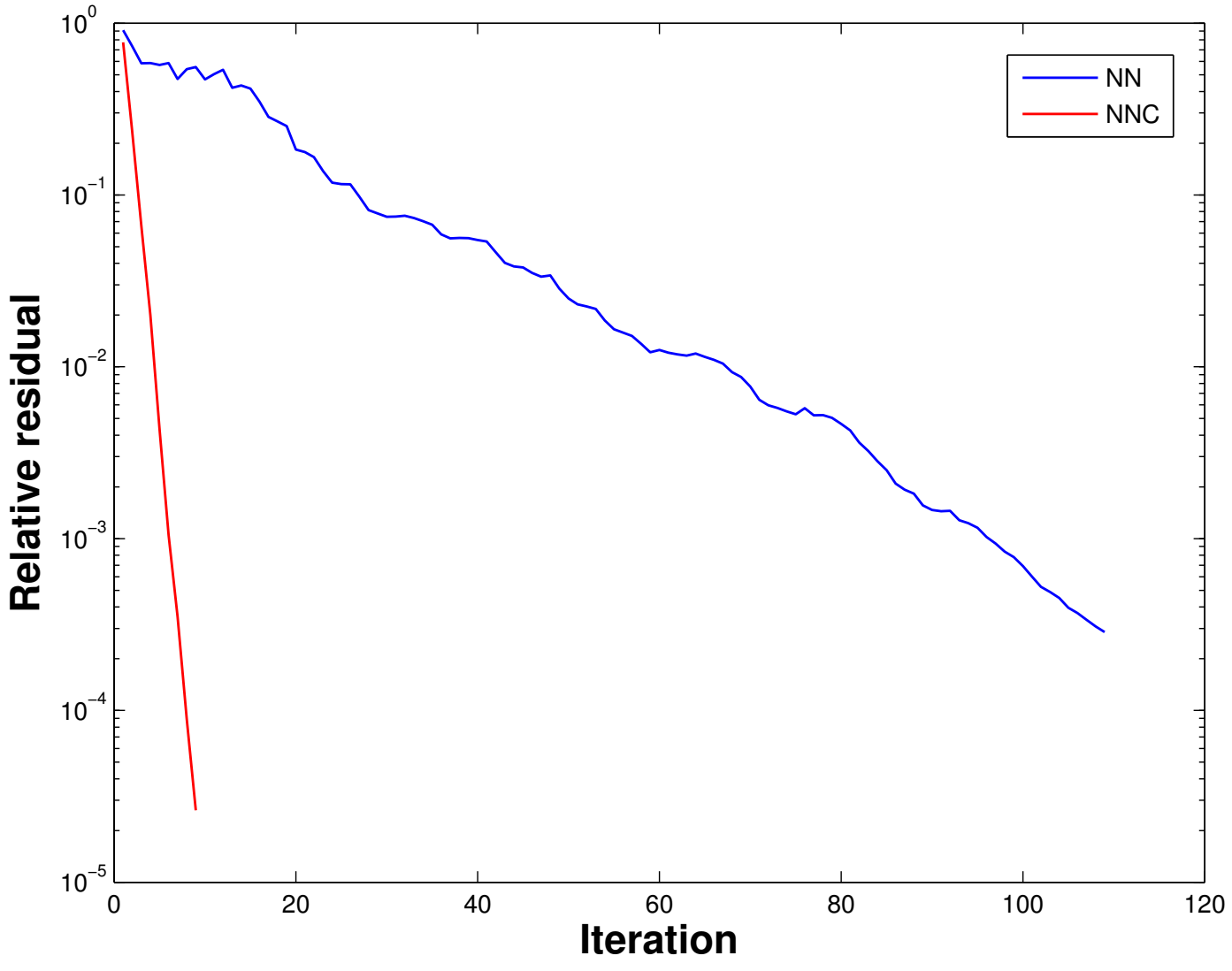
a) Neumann-Neumann with Coarse grid, b) Primal-Primal,c) Dual-Primal Operator.

Investigated numerical and parallel scalabilities:

Subber, W. and Sarkar, A., JCP, 2014

Subber, W. and Sarkar, A., CMAME, 2013

PCGM Iterations



- Stationary diffusion equation with a Dirichlet boundary condition.

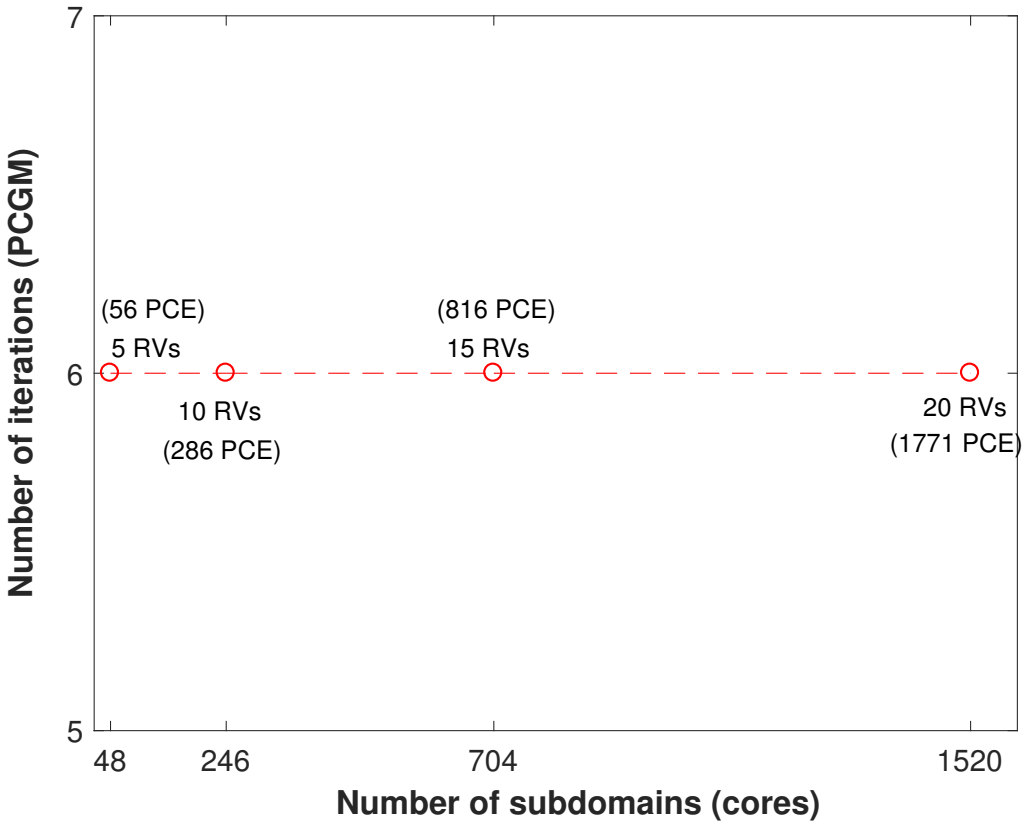
$$\begin{aligned} -\nabla \cdot \left(\, c_d(\mathbf{x},\theta) \;\; \nabla U(\mathbf{x},\theta) \, \right) &= F(\mathbf{x}), & \Omega \times \mathcal{W}, \\ U(\mathbf{x},\theta) &= 0, & \delta\Omega \times \mathcal{W}, \end{aligned}$$

- Random diffusion coefficient c_d modelled as a non-Gaussian stochastic process (here lognormal), obtained from a Gaussian process with mean μ, variance σ^2 and exponential covariance function C (on a 2D domain).

$$C(x_1, y_1; x_2, y_2) = \sigma^2 \; e^{-|x_2 - x_1|/b_1 - |y_2 - y_1|/b_2}, \qquad (2)$$

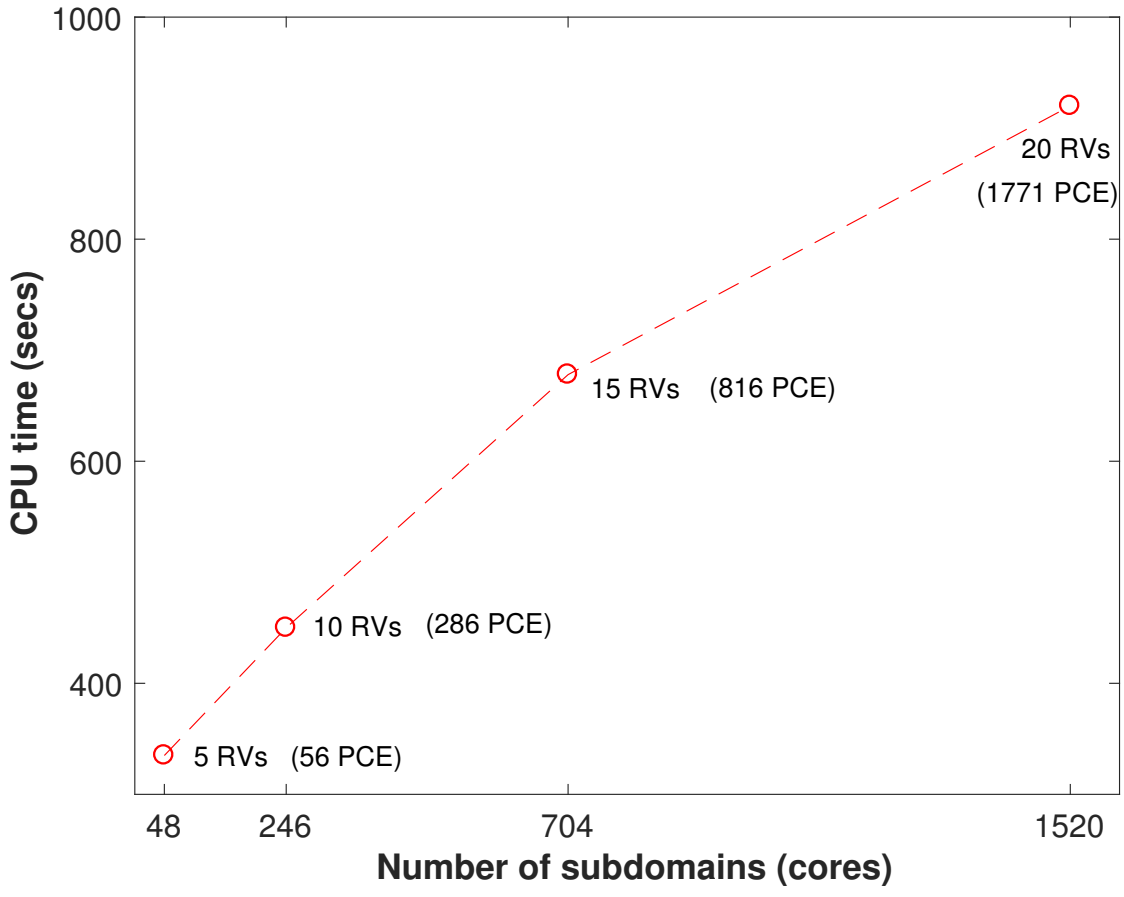
For all simulations we use, $\sigma = 0.3$ and $b_1 = b_2 = 1.0$

Numerical Scalability With Respect to Number of Random Variables: NNC/BDDC



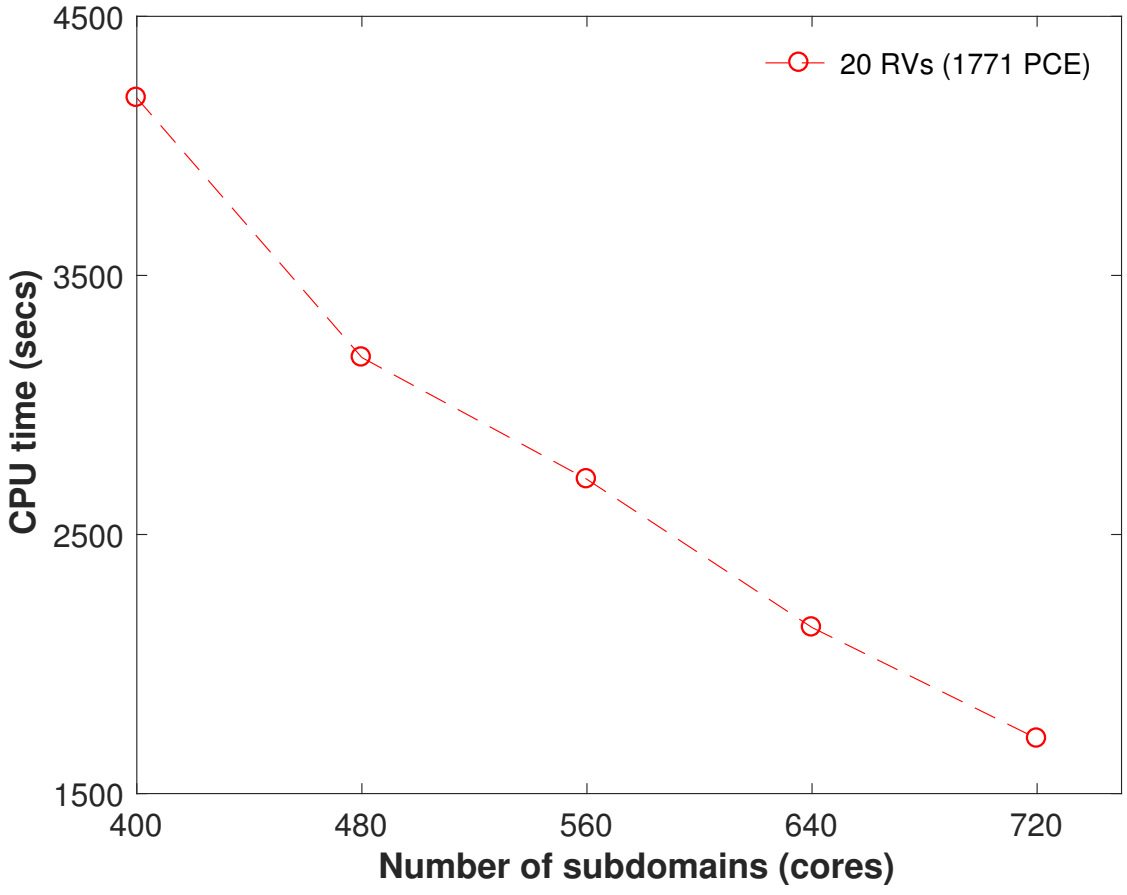
With the **fixed problem size per subdomain** ($N * P \approx 60000$). $N \approx 52,000$, fixed mesh resolution and P is number of PCE terms using **third** order expansion.

Scalability With Respect To Number of Random Variables: NNC/BDDC



With the **fixed problem size per subdomain** ($N * P \approx 60000$).
$N \approx 52,000$, fixed mesh resolution and P is number of PCE terms using **third** order expansion.

Parallel Scalability (Strong): NNC/BDDC



With the **fixed total problem size** (i.e. fixed mesh resolution $N \approx 52,000$ and the fixed number of PCE terms $P = 1771$) using **third** order expansion.

- Independent evaluation of the PCE coefficients of solution process,

$$\hat{\mathbf{u}}_k = \frac{1}{< \Psi_k^2 >} \int_\Omega \mathbf{u}(\theta)\Psi_k(\xi)d\xi.$$

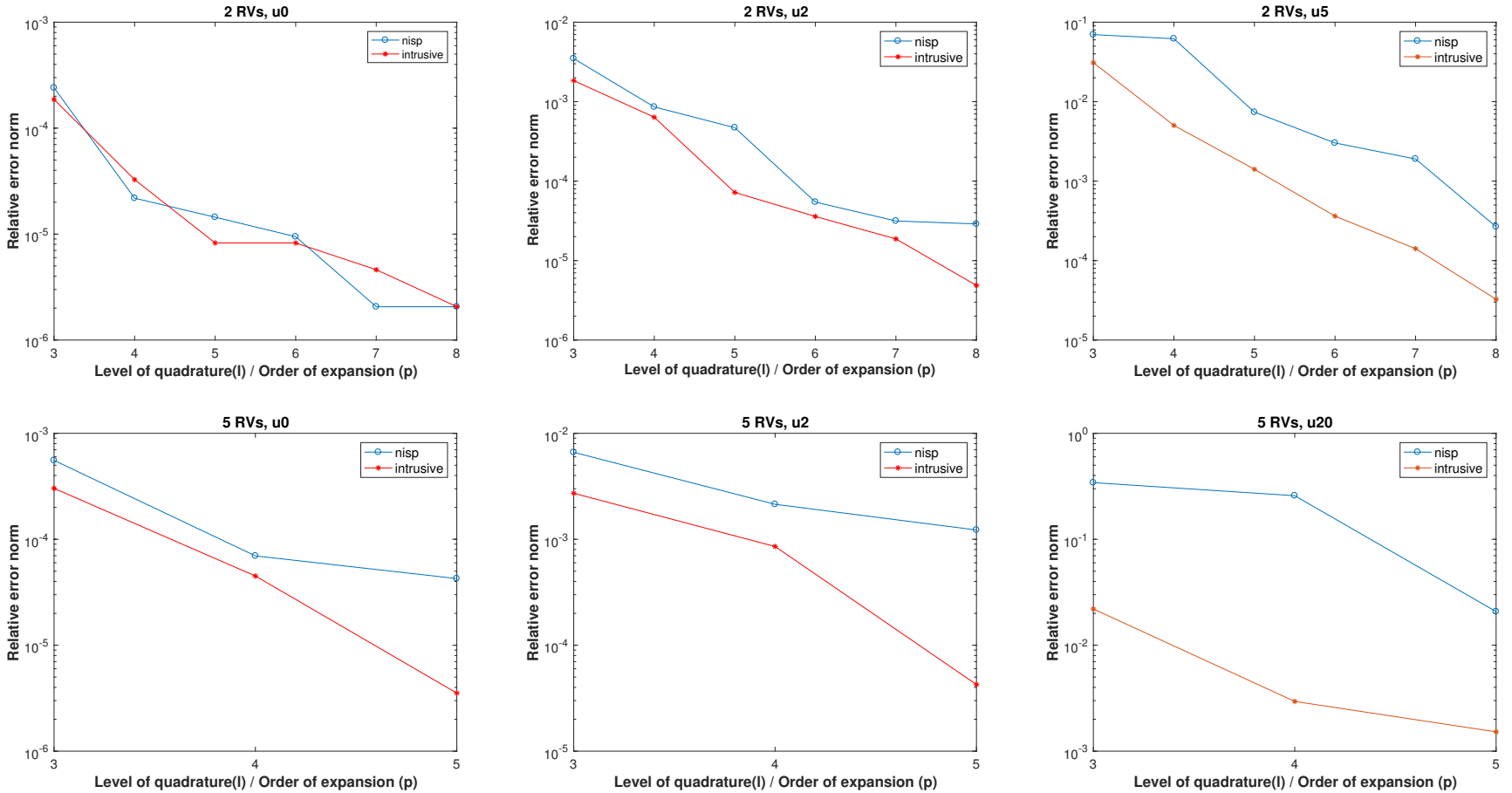
  The integral in the numerator can be solved by using sparse grid quadrature rule.

- Smolyak's quadrature rule for a d dimensional function $\mathbf{f}$ at l level using univariate quadrature $\mathbf{Q}_l^{(1)}$ in recursive form

$$\mathbf{Q}_l^{(d)}\mathbf{f} = \left(\sum_{i=0}^{l}(\mathbf{Q}_l^{(1)} - \mathbf{Q}_{l-1}^{(1)}) \otimes \mathbf{Q}_{l-i+1}^{d-1}\right)\mathbf{f}$$

Errors Analysis of PCE Coefficients of Solution Process:

Intrusive Vs Non-Intrusive

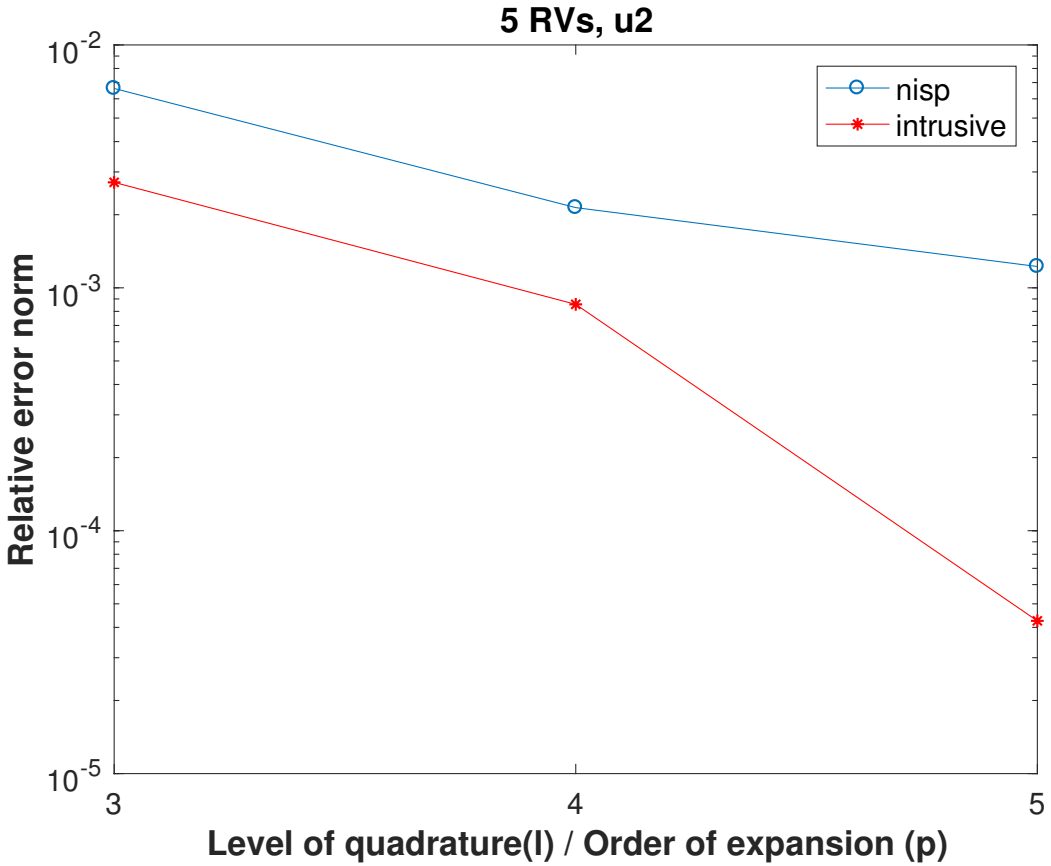


Matlab simulations with coarse mesh ($N \approx 150$) for RVs, 2 and 5.

Relative error $= \frac{\|u_{iH} - u_{ih}\|}{\|u_{iH}\|}$, where $h = 3, 4, ..., H$ and $i = 1, 2, ..., P$

In intrusive $h = p$ and in non-intrusive $h = l$.

Errors Analysis of PCE Coefficients of Solution Process: Intrusive Vs Non-Intrusive



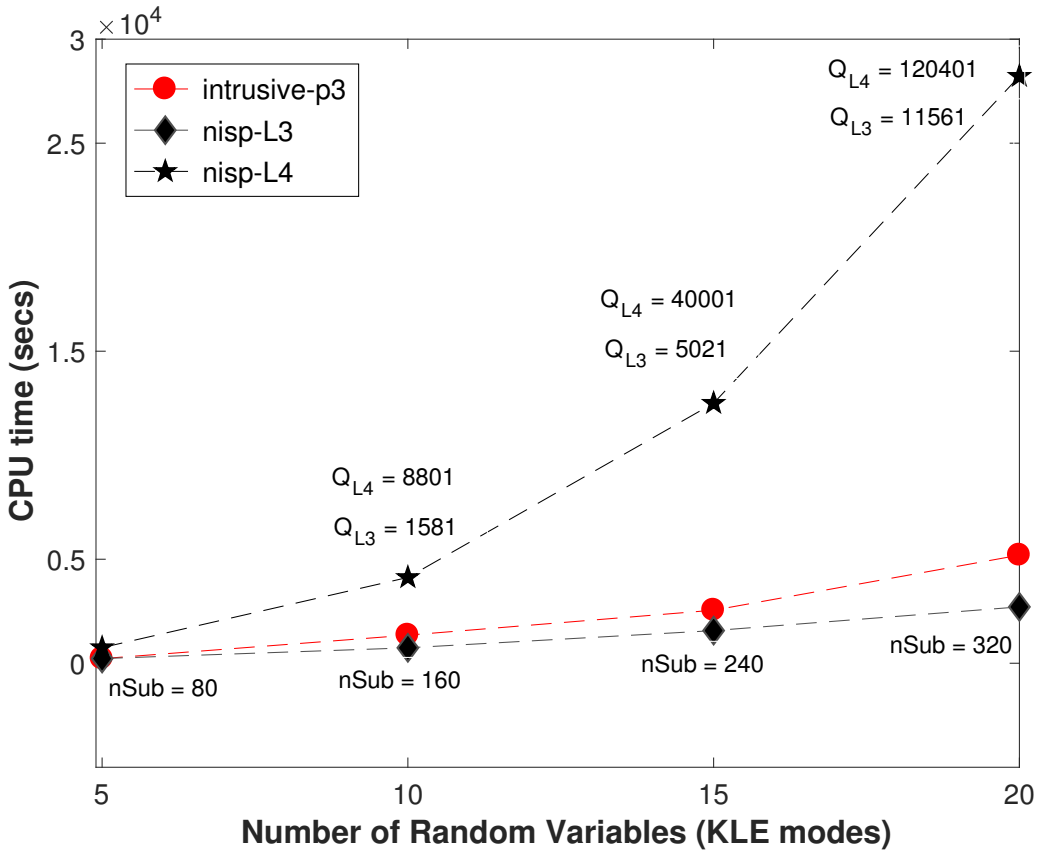
For the case of 5 random variables, error in PCE coefficient ($\hat{\mathbf{u}}_2$)

Relative error $= \frac{\|u_{iH} - u_{ih}\|}{\|u_{iH}\|}$, where $h = 3, 4, ..., H$ and $i = 1, 2, ..., P$

In intrusive $h = p$ and in non-intrusive $h = l$.

Scalability with Number of Stochastic Dimensions:

Intrusive Vs Non-Intrusive (Sparse Grid)



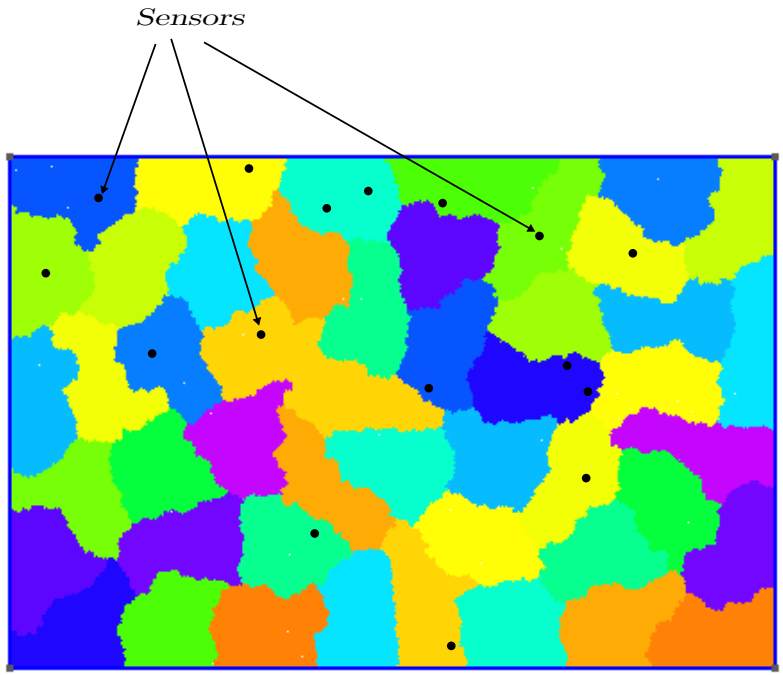
Intrusive with the order of expansion ($p = 3$) and non-intrusive with level of sparse-grid ($L = 3$ and $L = 4$).
For a fixed mesh resolution ($N \approx 52,000$).

- Model Equation

$$\mathbf{u}_{k+1} = \boldsymbol{\psi}_k \left(\mathbf{u}_k, \mathbf{f}_k, \mathbf{q}_k \right) \quad -- \quad \text{Forecast Step}$$

- Measurement Equation

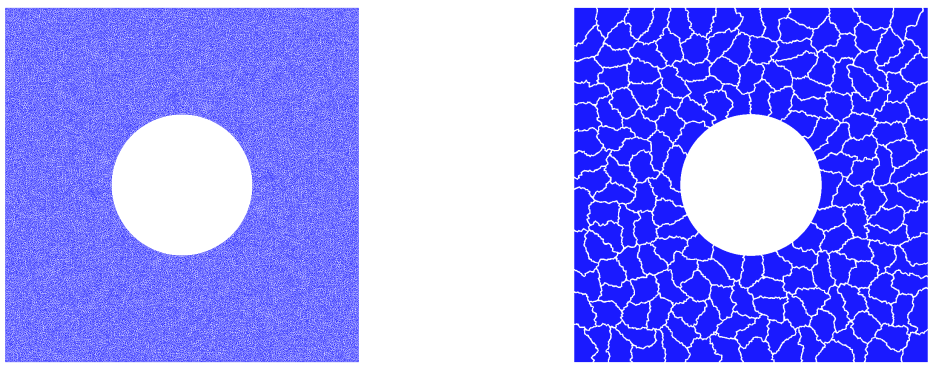
$$\mathbf{d}_k = \mathbf{h}_k \left(\mathbf{u}_k, \boldsymbol{\epsilon}_k \right) \qquad -- \text{ Assimilation Step}$$



Stationary Stochastic Diffusion Problem

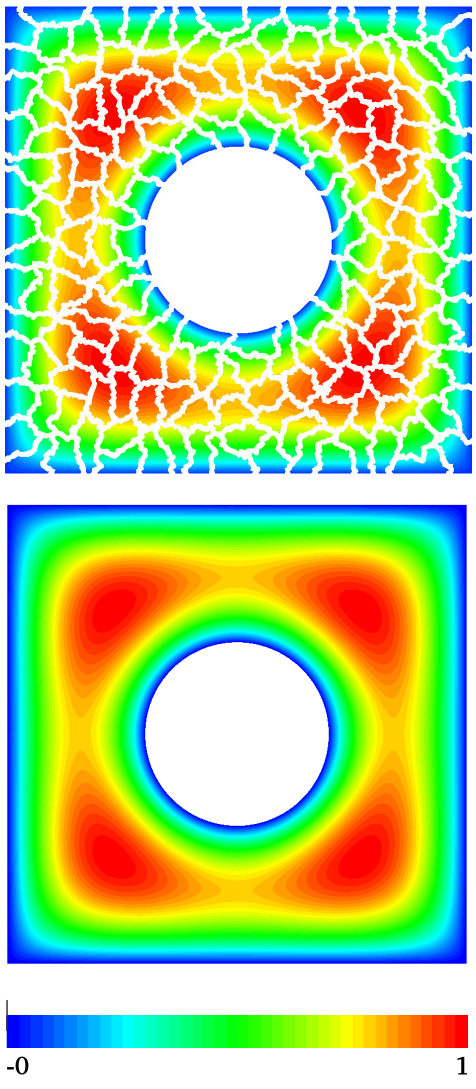
$$-\nabla \cdot (c(\mathbf{x},\theta)\ \nabla u(\mathbf{x},\theta)) \quad = \quad f(\mathbf{x}) \qquad \text{in} \quad D \times \Omega\ ,$$

- FEM mesh: 100,241 elements and 50,118 nodes
- Domain Decomposition

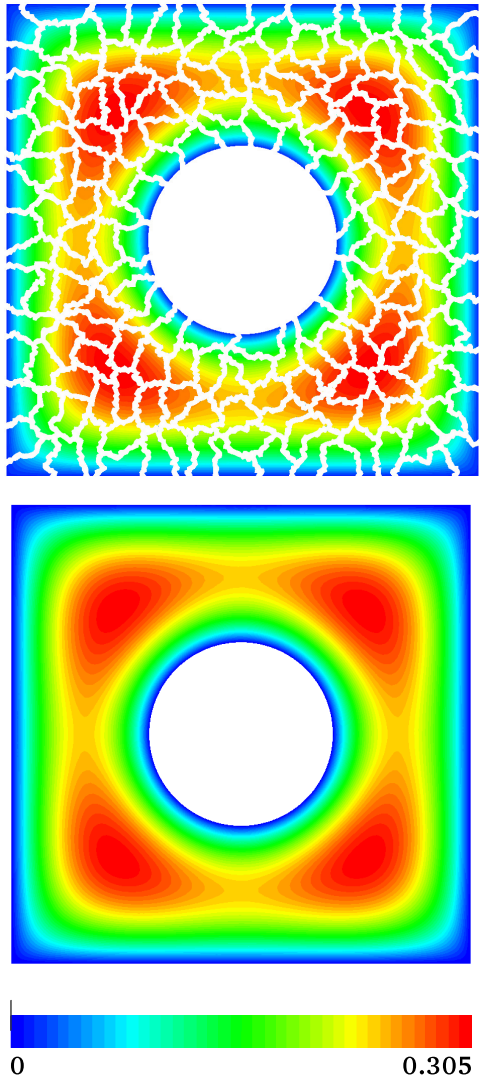


(a) (b)

- Diffusion coefficient c modeled as a log-normal stochastic process and 20-term PC expansions for both c and u.
- Linear system of order 1,002,360.

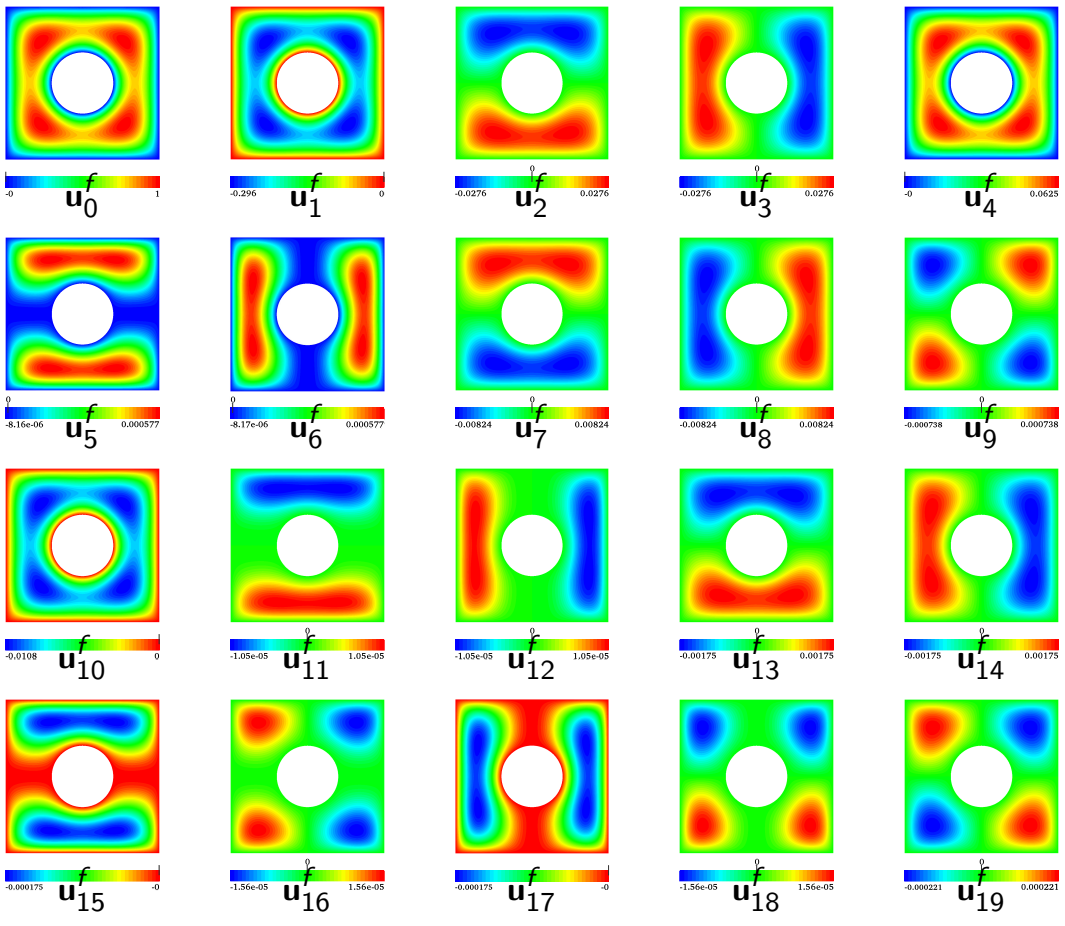


Mean concentration

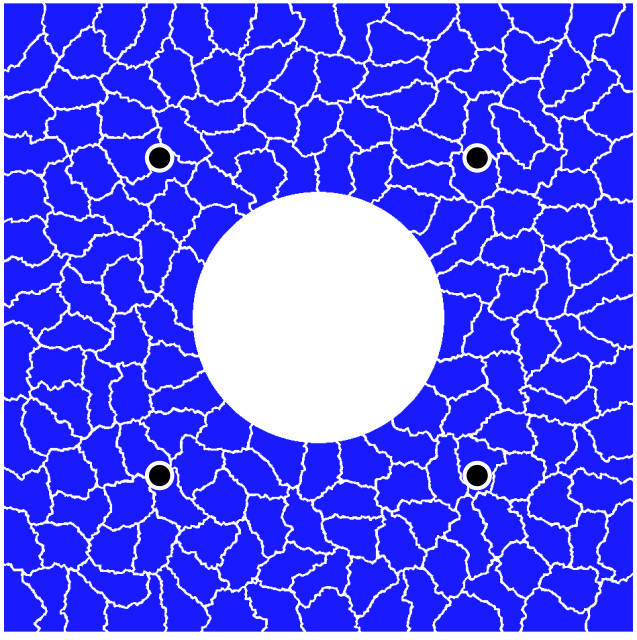


Standard deviation

Diffusion: Prior PCE Coefficients



Diffusion: Experiment with Four Sensors



Parallel Data Assimilation: Polynomial Chaos-Based Kalman Filter (Saad and Ghanem, 2009)

- State Vector Representation

$$\mathbf{u}(\theta) = \sum_{i=0}^{N} \mathbf{u}_i \Psi_i(\theta) \; ; \quad \mathbf{P}^f = \sum_{i=1}^{N} \mathbf{u}_i \mathbf{u}_i' \langle \Psi_i^2 \rangle \; .$$

$$\mathbb{A}^f = [\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_N]$$

- Measurement Vector Representation

$$\mathbf{d} = \sum_{j=0}^{N} \mathbf{d}_j \Psi_j(\theta) \; ; \quad \mathbf{\Gamma} = \sum_{i=1}^{N} \mathbf{d}_i \mathbf{d}_i' \langle \Psi_i^2 \rangle$$

$$\mathbb{D} = [\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_N]$$

- Analysis/Update Step (Saad and Ghanem, 2009)

$$\mathbb{A}^a = \mathbb{A}^f + \mathbf{P}^f \mathbf{H}' \left(\mathbf{H} \mathbf{P}^f \mathbf{H}' + \mathbf{\Gamma} \right)^{-1} \left(\mathbb{D} - \mathbf{H} \mathbb{A}^f \right)$$

Parallel Data Assimilation: Analysis Step - Distributed Implementation

- Spatial decomposition of the state vector

$$\mathbf{u}(\theta) = \sum_{i=0}^{N} \begin{bmatrix} \mathbf{u}_{l,i}^{1} \\ \vdots \\ \mathbf{u}_{l,i}^{n_s} \\ \mathbf{u}_{\Gamma,i} \end{bmatrix} \Psi_i(\theta) \ .$$

- Decomposition of the state ensemble matrix

$$\mathbb{A}^f = \begin{bmatrix} \mathbf{u}_{l,0}^{1} & \mathbf{u}_{l,1}^{1} & \cdots & \mathbf{u}_{l,N}^{1} \\ \mathbf{u}_{l,0}^{2} & \mathbf{u}_{l,1}^{2} & \cdots & \mathbf{u}_{l,N}^{2} \\ \vdots & \vdots & & \vdots \\ \mathbf{u}_{l,0}^{n_s} & \mathbf{u}_{l,1}^{n_s} & \cdots & \mathbf{u}_{l,N}^{n_s} \\ \mathbf{u}_{\Gamma,0} & \mathbf{u}_{\Gamma,1} & \cdots & \mathbf{u}_{\Gamma,N} \end{bmatrix} = \begin{bmatrix} \left[\mathbb{A}^f\right]_l^1 \\ \left[\mathbb{A}^f\right]_l^2 \\ \vdots \\ \left[\mathbb{A}^f\right]_l^{n_s} \\ \left[\mathbb{A}^f\right]_\Gamma \end{bmatrix},$$

Parallel Data Assimilation

- Analysis Step - Distributed Implementation

$$\begin{aligned}\mathbb{A}^a &= \mathbb{A}^f + \mathbf{P}^f\mathbf{H}'\left(\mathbf{H}\mathbf{P}^f\mathbf{H}' + \mathbf{\Gamma}\right)^{-1}\left(\mathbb{D} - \mathbf{H}\mathbb{A}^f\right)\\ &= \mathbb{A}^f\left[\mathbf{I} + \mathbf{B}\mathbb{D}^{f\prime}\left(\mathbb{D}^f\mathbf{B}\mathbb{D}^{f\prime} + \mathbb{D}\mathbf{B}\mathbb{D}\right)^{-1}\left(\mathbb{D} - \mathbb{D}^f\right)\right]\\ &= \mathbb{A}^f\mathbf{C}\end{aligned}$$

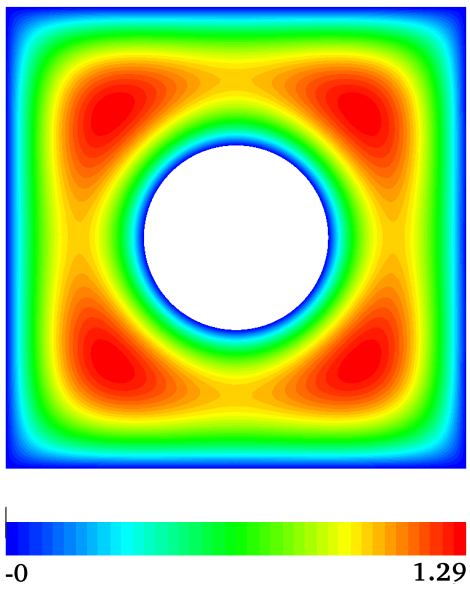
- Compute the forecast observation matrix $\mathbb{D}^f = \mathbf{H}\mathbb{A}^f$ in parallel

$$\begin{aligned}\mathbb{D}^f &= \mathbf{H}\mathbb{A}^f\\ &= \mathbf{H}_I^1\left[\mathbb{A}^f\right]_I^1 + \mathbf{H}_I^2\left[\mathbb{A}^f\right]_I^2 + \cdots \mathbf{H}_I^{n_s}\left[\mathbb{A}^f\right]_I^{n_s} + \mathbf{H}_\Gamma\left[\mathbb{A}^f\right]_\Gamma\\ &= \left[\mathbb{D}^f\right]_I^1 + \left[\mathbb{D}^f\right]_I^2 + \cdots + \left[\mathbb{D}^f\right]_I^{n_s} + \left[\mathbb{D}^f\right]_\Gamma = \sum_{s=1}^{n_s}\left[\mathbb{D}^f\right]_I^s + \left[\mathbb{D}^f\right]_\Gamma\ .\end{aligned}$$

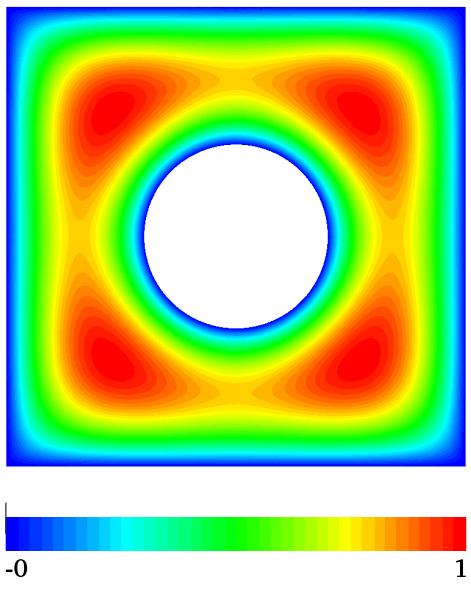
- Perform ensemble update in parallel

$$\begin{array}{rl}\textit{Interior nodes, Subdomain 1} \longrightarrow & \\ \vdots & \\ \textit{Interior nodes, Subdomain } n_s \longrightarrow & \\ \textit{Global boundary nodes} \longrightarrow & \end{array}\begin{bmatrix}[\mathbb{A}^a]_I^1\\ \vdots\\ [\mathbb{A}^a]_I^{n_s}\\ [\mathbb{A}^a]_\Gamma\end{bmatrix} = \begin{bmatrix}\left[\mathbb{A}^f\right]_I^1\\ \vdots\\ \left[\mathbb{A}^f\right]_I^{n_s}\\ \left[\mathbb{A}^f\right]_\Gamma\end{bmatrix}\mathbf{C} = \begin{bmatrix}\left[\mathbb{A}^f\right]_I^1\mathbf{C}\\ \vdots\\ \left[\mathbb{A}^f\right]_I^{n_s}\mathbf{C}\\ \left[\mathbb{A}^f\right]_\Gamma\mathbf{C}\end{bmatrix}$$

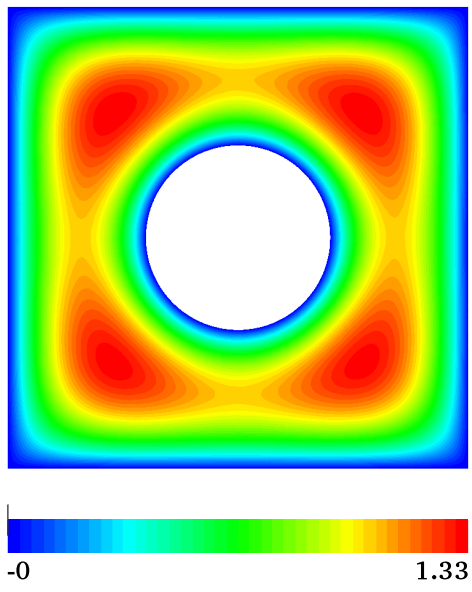
Parallel Data Assimilation: *Data Assimilation*



Actual concentration

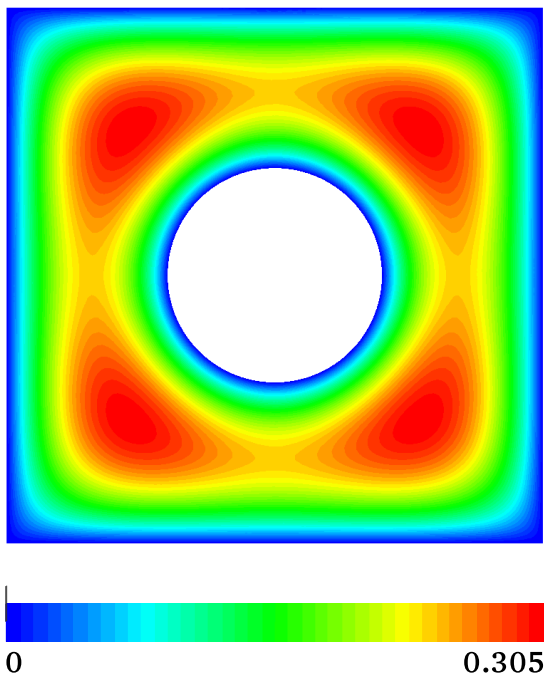


Prior Estimate

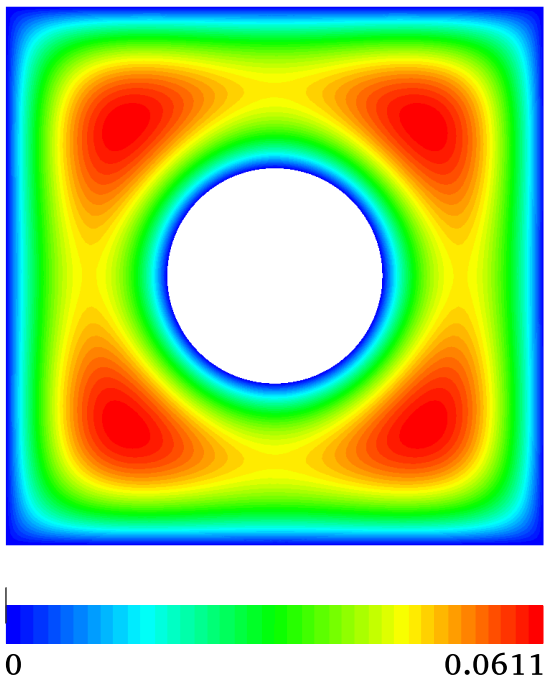


Posterior Estimate

Parallel Data Assimilation: *Data Assimilation*

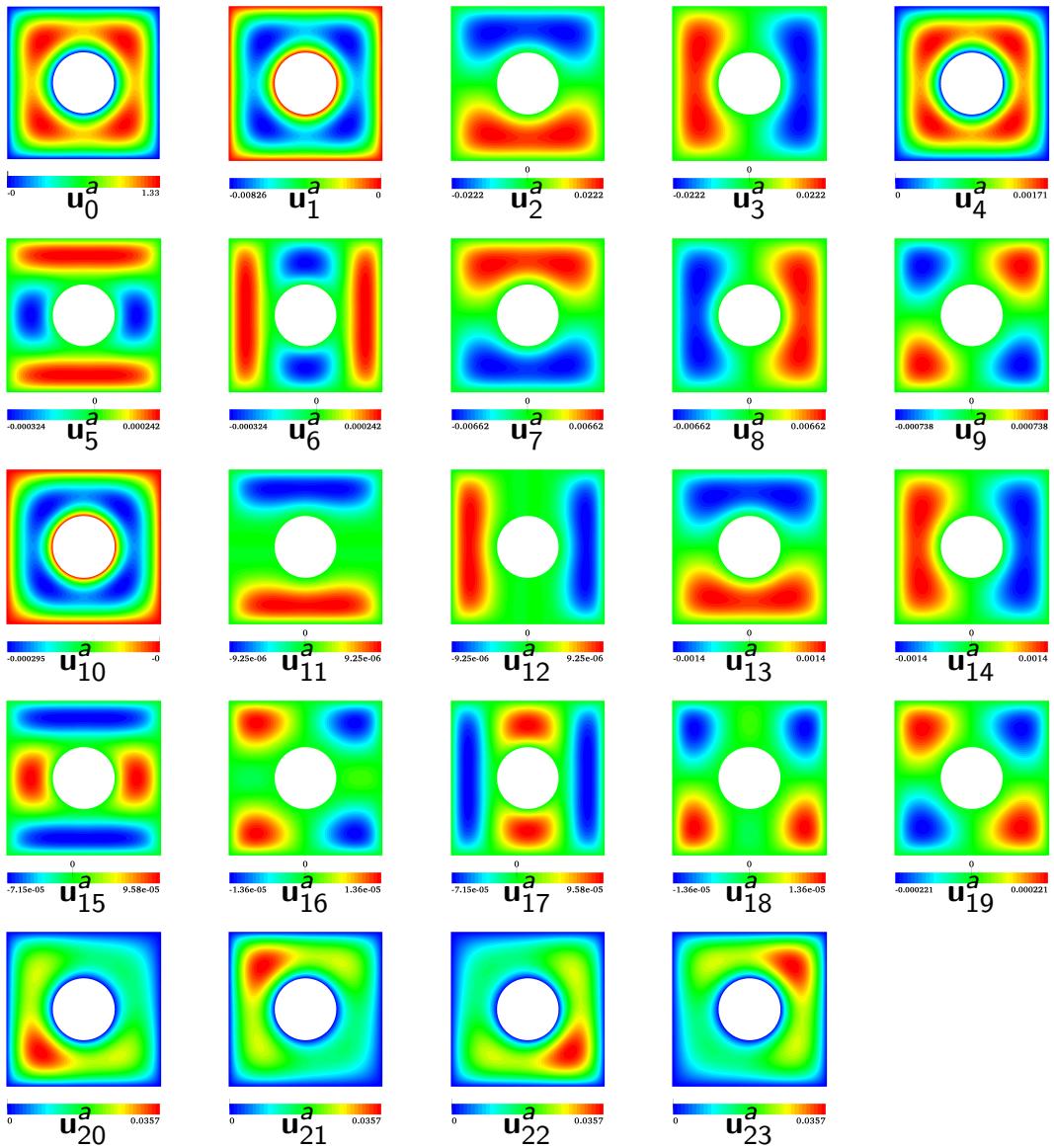


Prior Standard Deviation



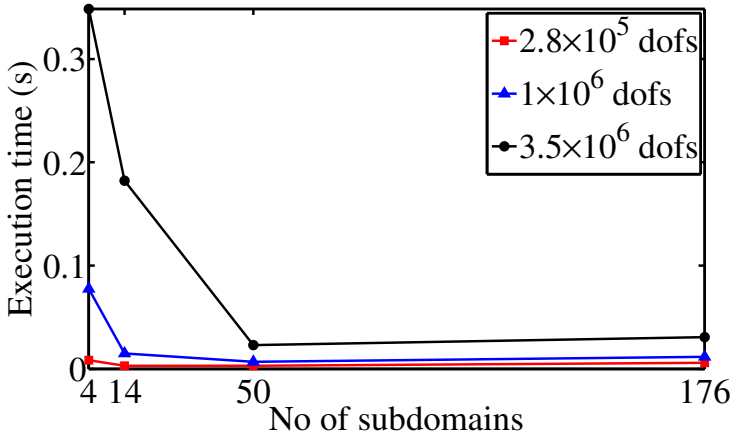
Posterior Standard Deviation

Diffusion: Posterior PCE Coefficients - 4 Sensors

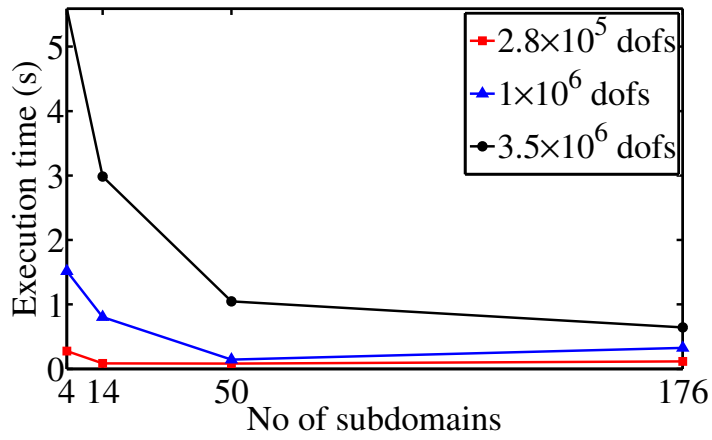


Parallel Scalability

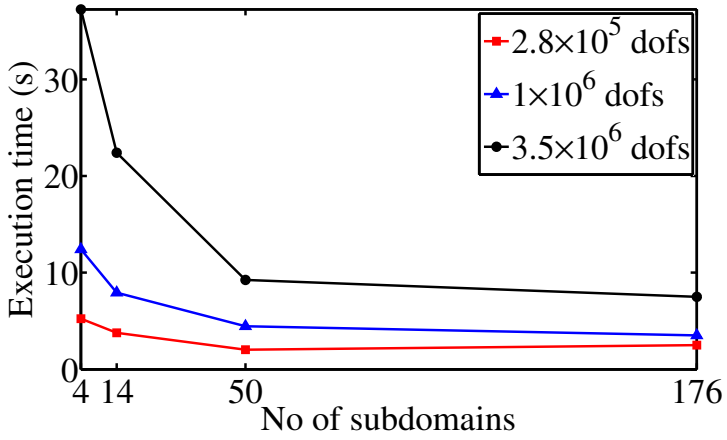
For a fixed problem size and number of measurements:



32 measurements



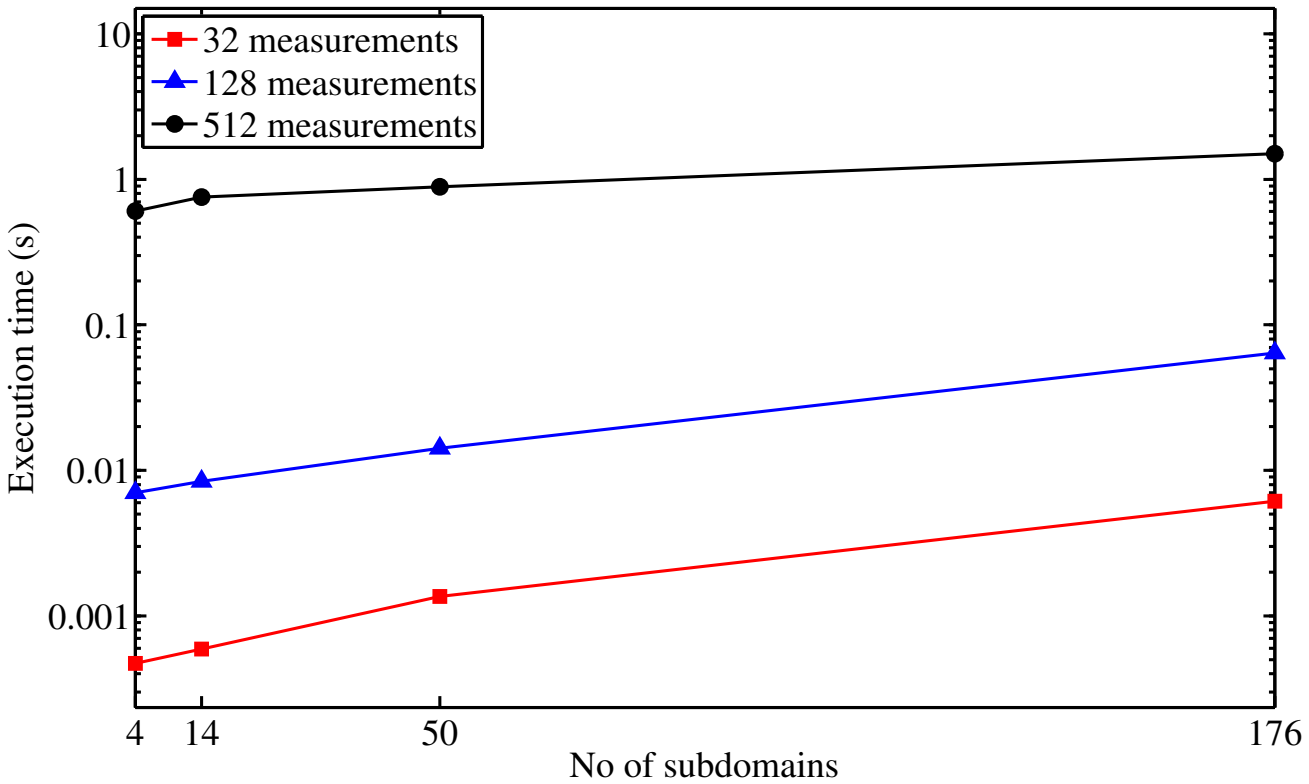
128 measurements



512 measurements

Parallel Scalability

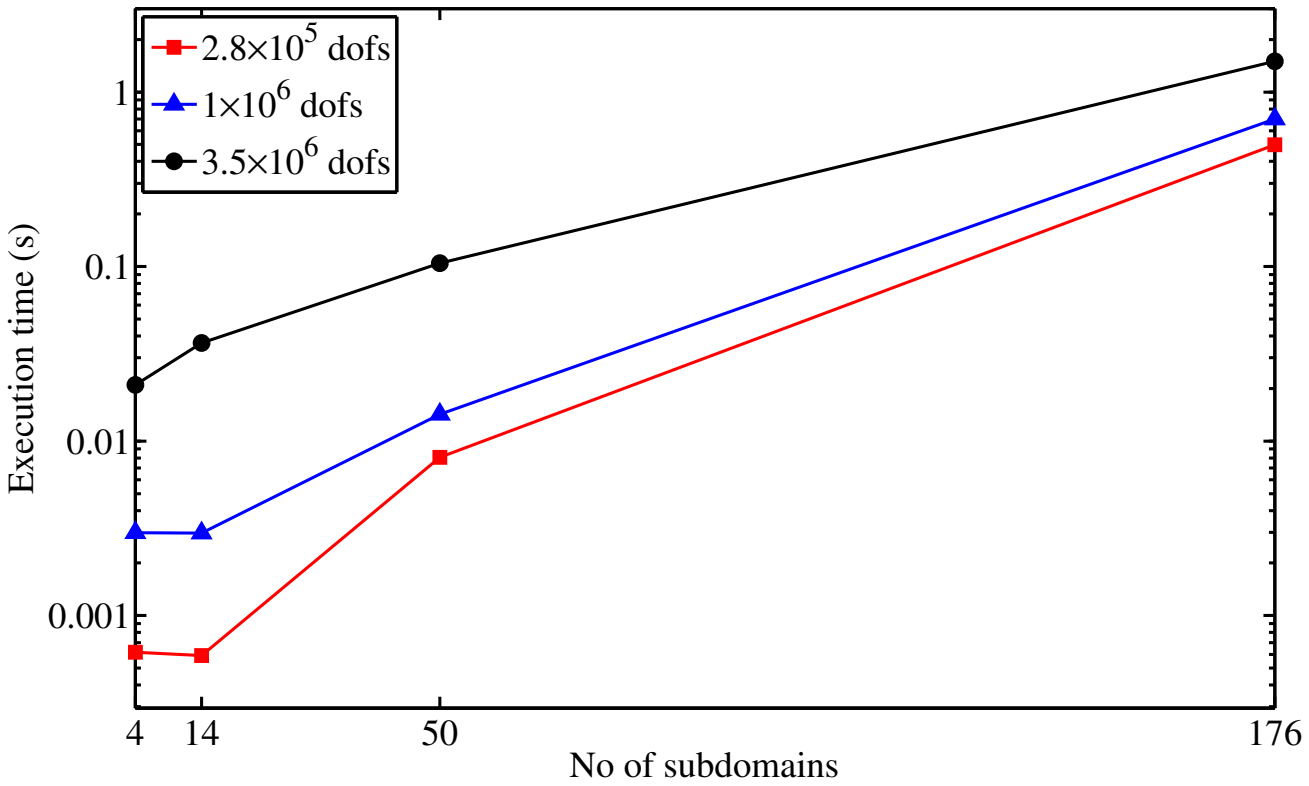
For a fixed problem size per subdomain/core:



20,000 unknowns per subdomain/core

Parallel Scalability

For a fixed number of measurements per subdomain/core:



2 measurements per subdomain/core

$$\mathbf{C} = \mathbf{I} + \mathbf{B}\mathbb{D}^{f'}\left(\mathbb{D}^{f}\mathbf{B}\mathbb{D}^{f'} + \mathbb{D}\mathbf{B}\mathbb{D}\right)^{-1}\left(\mathbb{D} - \mathbb{D}^{f}\right)$$

Two-level DDM solver & Update

Table: Forward Time (No of measurements = 2048)

No of dof's ($\times 10^5$)	NP=76	NP=100	NP=140	NP=176
2.01	117.1	63.0	38.4	30.0
5.02	1390.0	617.2	282.8	172.9
9.92	-	4023.7	1660.8	900.7
15.0	-	-	5031.6	2867.23

Table: N = 5.02×10^5, No of measurements = 1024)

Execution Time	NP=76	NP=100	NP=140	NP=176
FT	1390.0	617.2	282.8	172.9
UT	16.48	16.51	16.73	17.02
TT	1406.5	633.7	299.5	189.92

Two-level DDM solver & Update

Table: N $= 5.02 \times 10^5$, No of measurements $= 2048$)

Execution Time	NP=76	NP=100	NP=140	NP=176
FT	1390.0	617.2	282.8	172.9
UT	122.3	122.98	124.3	125.08
TT	1512.38	740.1	407.7	297.9

Table: N $= 5.02 \times 10^5$, No of measurements $= 4096$)

Execution Time	NP=76	NP=100	NP=140	NP=176
FT	1390.0	617.2	282.8	172.9
UT	970.9	971.0	972.5	974.3
TT	2360.9	1588.2	1258.3	1147.2

PCKF Update

- The PCKF coefficients are updated using

$$\mathbf{A}^a = \mathbf{A}^f + \mathbf{K}\left(\mathbf{D} - \mathbf{H}\mathbf{A}^f\right)$$

- Where the Kalman gain is

$$\mathbf{K} = \mathbf{A}^f\mathbf{B}\left[\mathbf{H}\mathbf{A}^f\right]^{\mathrm{T}}\left(\left[\mathbf{H}\mathbf{A}^f\right]\mathbf{B}\left[\mathbf{H}\mathbf{A}^f\right]^{\mathrm{T}} + \mathbf{D}\mathbf{B}_m\mathbf{D}^{\mathrm{T}}\right)^{-1}$$

- The costly step of the update step is the inversion of matrix in the Kalman gain. The size of this matrix is $m \times m$, where m is the number of measurements.
- Instead of inverting the matrix, a linear system is solved

$$\mathbf{A}^a = \mathbf{A}^f + \mathbf{A}^f\mathbf{B}\mathbf{D}^{f\mathrm{T}}\underbrace{\left(\mathbf{D}^f\mathbf{B}\mathbf{D}^{f\mathrm{T}} + \mathbf{D}\mathbf{B}_m\mathbf{D}^{\mathrm{T}}\right)^{-1}\left(\mathbf{D} - \mathbf{D}^f\right)}_{\mathbf{Z}}$$

$$\left(\mathbf{D}\mathbf{B}_m\mathbf{D}^{\mathrm{T}} + \mathbf{D}^f\mathbf{B}\mathbf{D}^{f\mathrm{T}}\right)\mathbf{Z} = \mathbf{D} - \mathbf{D}^f$$

Sherman-Morrison Solver

- For independent measurement

$$\mathbf{D}\mathbf{B}_m\mathbf{D}^{\mathrm{T}} = \mathbf{A}_0$$

  is a diagonal matrix. Use of SM is advantageous in this case
- Using the following relations

$$\mathbf{D}^f\mathbf{B} = \mathbf{U}, \quad \mathbf{D}^f = \mathbf{V}$$

  The left hand side matrix can be expressed as

$$\mathbf{A} = \left(\mathbf{D}\mathbf{B}_m\mathbf{D}^{\mathrm{T}} + \mathbf{D}^f\mathbf{B}\mathbf{D}^{f\,\mathrm{T}}\right) = \left(\mathbf{A}_0 + \sum_{i=1}^{N}\mathbf{u}_i\mathbf{v}_i^{\mathrm{T}}\right)$$

- Sherman-Morrison formula is recursively used to solve the underlying linear system

$$(\mathbf{A}_0 + \mathbf{u}\mathbf{v}^{\mathrm{T}})^{-1} = \mathbf{A}_0^{-1} - \frac{\mathbf{A}_0^{-1}\mathbf{u}\mathbf{v}^{\mathrm{T}}\mathbf{A}_0^{-1}}{1.0 + \mathbf{v}^{\mathrm{T}}\mathbf{A}_0^{-1}\mathbf{u}}$$

Sherman-Morrison based PCKF Update

Algorithm 1 Parallel Sherman-Morrison based PCKF Update

1: All Gather: $\mathbf{D}^f = \mathbf{H}_\Gamma \mathbf{A}^f{}_\Gamma + \sum_{s=1}^{n_s} \mathbf{H}_I^s \mathbf{A}^{fs}{}_I$

2: $\mathbf{A}_0 = \mathbf{D}\mathbf{B}_m\mathbf{D}^{\mathrm{T}}$

3: **for** i = 1 to N **do** (Process work on different columns)

4: Solve $\left(\mathbf{A}_0 + \sum_{i=1}^{N} \mathbf{u}_i \mathbf{v}_i^{\mathrm{T}}\right) \mathbf{z}_i = \mathbf{d}_i$ where $\mathbf{d}_i$ is the i^{th} column of $\mathbf{D} - \mathbf{D}^f$

5: $\mathbf{c}_i = \mathbf{B}\mathbf{D}^{f\,\mathrm{T}} z_i$

6: **end for**

7: Gather and scatter columns of $\mathbf{C}$ across all processors

8: (Each processor) Update the local PCE coefficients: $\mathbf{A}^{as}{}_I = \mathbf{A}^{fs}{}_I(\mathbf{I} + \mathbf{C})$

9: (Head node) Update the interface PCE coefficients: $\mathbf{A}^a{}_\Gamma = \mathbf{A}^f{}_\Gamma(\mathbf{I} + \mathbf{C})$

Sherman-Morrison based PCKF Update

Algorithm 2 Sherman-Morrison based solver

1: Solve $\left(\mathbf{A}_0 + \sum_{i=1}^{N} \mathbf{u}_i \mathbf{v}_i^{\mathrm{T}}\right) \mathbf{x} = \mathbf{b}$
2: Solve $\mathbf{A}_0 \mathbf{x}_0 = \mathbf{b}$
3: Solve $\mathbf{A}_0 \mathbf{y}_{0,k} = \mathbf{u}_k$ for $k = 1, \ldots, N$
4: **for** i=1 to N-1 **do**
5: $\quad \mathbf{x}_i = \mathbf{x}_{i-1} - \dfrac{\mathbf{v}_i^T \mathbf{x}_{i-1}}{1.0 + \mathbf{v}_i^T \mathbf{y}_{i-1,i}} \mathbf{y}_{i-1,i}$
6: $\quad$ **for** k=i+1 to N **do**
7: $\quad\quad \mathbf{y}_{i,k} = \mathbf{y}_{i-1,k} - \dfrac{\mathbf{v}_i^T y_{i-1,k}}{1.0 + \mathbf{v}_i^T \mathbf{y}_{i-1,i}} \mathbf{y}_{i-1,i}$
8: $\quad$ **end for**
9: **end for**
10: $\mathbf{x}_N = \mathbf{x}_{N-1} - \dfrac{\mathbf{v}_N^T \mathbf{x}_{N-1}}{1.0 + \mathbf{v}_N^T \mathbf{y}_{N-1,N}} \mathbf{y}_{N-1,N}$

Sherman-Morrison based PCKF Update

- The computation of the N vectors of $\mathbf{y}_{i-1,i}$ does not depend on the right hand side.
- For subsequent right-hand sides, a efficient version of the solver can be employed provided.

Algorithm 3 Simplified Sherman-Morrison based solver

1: Solve $\left(\mathbf{A}_0 + \sum_{i=1}^{N} \mathbf{u}_i \mathbf{v}_i^{\mathrm{T}}\right) \mathbf{x} = \mathbf{b}_i$

2: Solve $\mathbf{A}_0 \mathbf{x}_0 = \mathbf{b}$

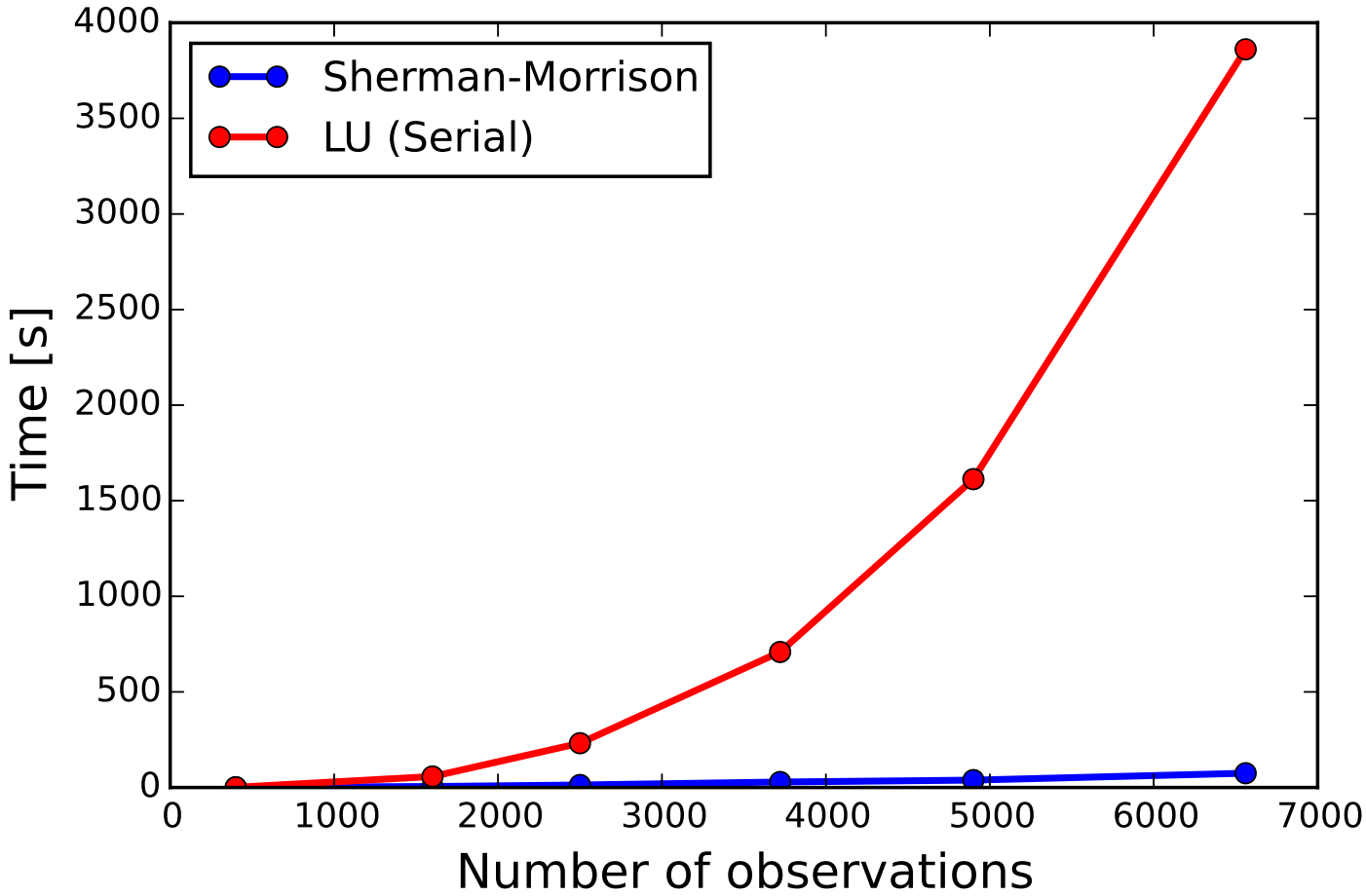
3: **for** i=1 to N **do**

4: $$\mathbf{x}_i = \mathbf{x}_{i-1} - \frac{\mathbf{v}_i^T \mathbf{x}_{i-1}}{1.0 + \mathbf{v}_i^T \mathbf{y}_{i-1,i}} \mathbf{y}_{i-1,i}$$

5: **end for**

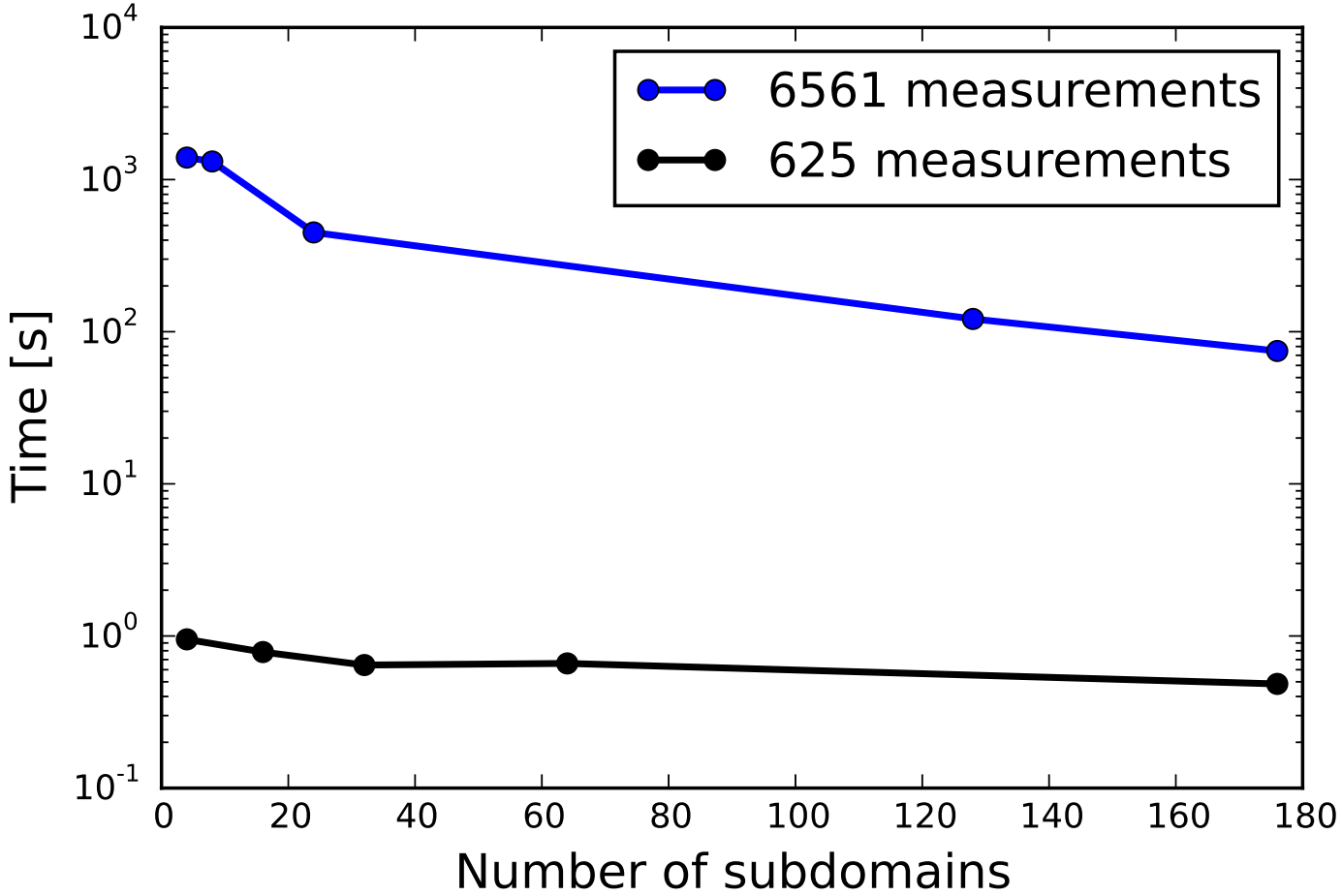
Timing

- 176 subdomains, 19602 nodes, 39209 elements



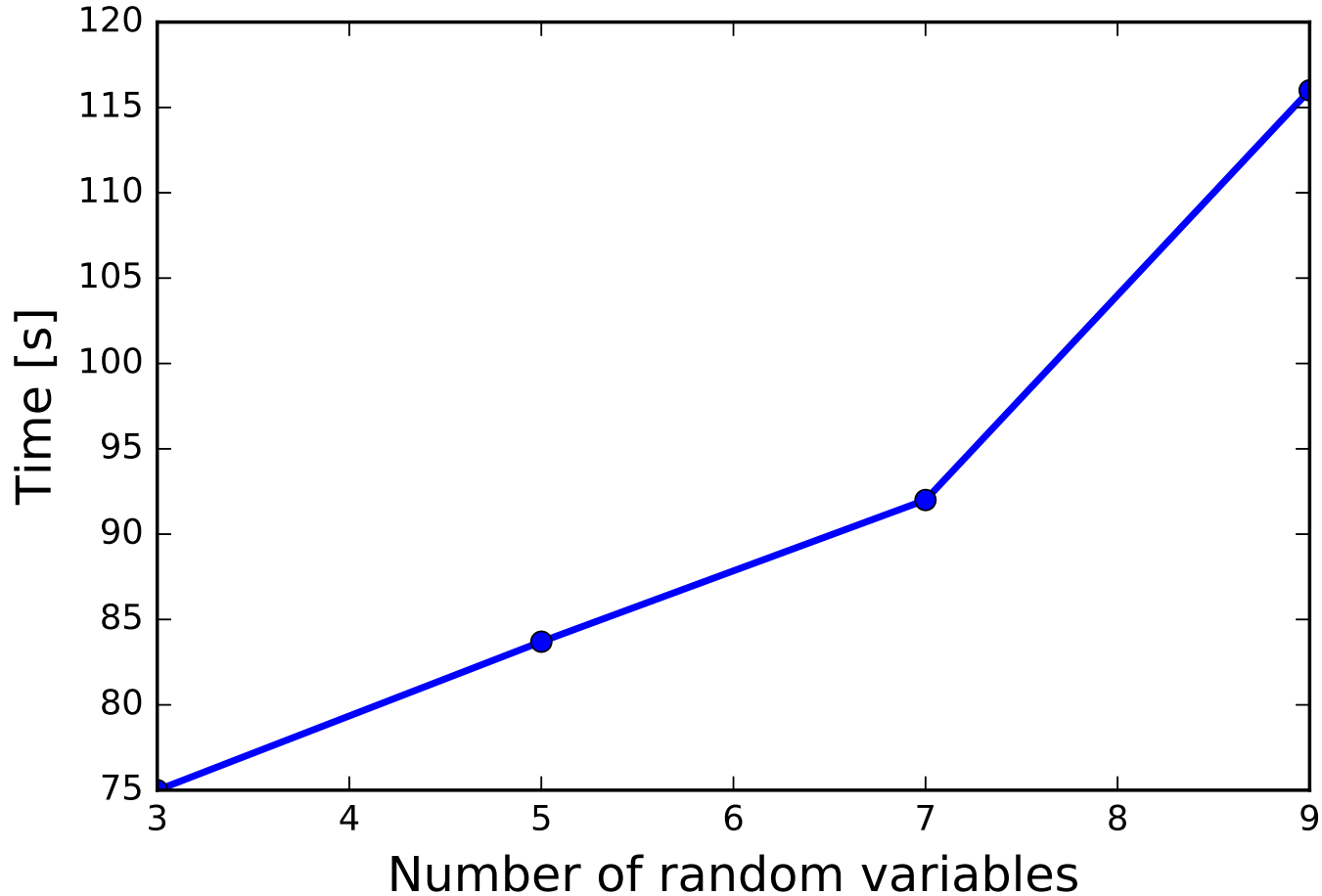
Timing

- Problem size: 19602 nodes, 39209 elements



Timing

- Problem size: 19602 nodes, 39209 elements, 6400 measurements



Sherman-Morrison based PCKF Update

- Doesn't scale well because of communication
- Need another approach to solve the underlying system

Preconditioned Conjugate Gradient Method (PCGM)

- Future work : using PCGM to solve the underlying linear system. Compared to the Sherman-Morrison approach, it will provide better scalability.
- Linear problem can be rewritten as

$$\underbrace{\sum_{s=1}^{n_s} \mathscr{R}_s^{\mathrm{T}} \left(\mathbf{D}_s^f \mathbf{B} {\mathbf{D}_s^f}^{\mathrm{T}} + \mathbf{D}_s \mathbf{B}_m \mathbf{D}_s^{\mathrm{T}} \right) \mathscr{R}_s}_{\mathcal{S}} \mathbf{Z} = \underbrace{\sum_{s=1}^{n_s} \mathscr{R}_s^{\mathrm{T}} \left(\mathbf{D}_s - \mathbf{D}_s^f \right)}_{\boldsymbol{G}}$$

$$\mathcal{M}^{-1} \mathcal{S} \mathbf{Z} = \mathcal{M}^{-1} \boldsymbol{G}$$

- Use of localization to get preconditioner.
- Each column of $\mathbf{Z}$ is solved independently.

$$\mathcal{M}^{-1} \mathcal{S} \mathbf{z}_i = \mathcal{M}^{-1} \boldsymbol{g}_i$$

Preconditioned Conjugate Gradient Method (PCGM)

1: $\mathbf{z}_{i,0} := 0$

2: Gather $\mathbf{r}_0 := \sum_{s=1}^{n_s} \mathscr{R}_s^{\mathrm{T}} \mathbf{g}_i^s$

3: Scatter $\mathbf{r}_0$

4: Gather $\mathcal{Z}_0 := \sum_{s=1}^{n_s} \mathscr{R}_s^{\mathrm{T}} \mathcal{M}_s^{-1} \mathscr{R}_s \mathbf{r}_0$

5: $\mathcal{P}_0 := \mathcal{Z}_0$

6: $p_0 := (\mathbf{r}_0, \mathcal{Z}_0)$

7: **for** j = 0,1,...,until convergence **do**

8: Scatter $\mathcal{P}_j$

9: Gather $\mathcal{Q}_j := \sum_{s=1}^{n_s} \mathscr{R}_s^{\mathrm{T}} \mathcal{S}_s \mathscr{R}_s \mathcal{P}_j$

10: $p_{tmp} := (\mathcal{P}_j, \mathcal{Q}_j)$

11: $\alpha := p_j / p_{tmp}$

12: $\mathbf{z}_{i,j+1} := \mathbf{z}_{i,j} + \alpha \mathcal{P}_j$

13: $\mathbf{r}_{j+1} := \mathbf{r}_j - \alpha \mathcal{Q}_j$

14: Scatter $\mathbf{r}_{j+1}$

15: Gather $\mathcal{Z}_{j+1} := \sum_{s=1}^{n_s} \mathscr{R}_s^{\mathrm{T}} \mathcal{M}_s^{-1} \mathscr{R}_s \mathbf{r}_{j+1}$

16: $p_{j+1} := (\mathbf{r}_{j+1}, \mathcal{Z}_{j+1})$

17: $\beta_j := p_{j+1} / p_j$

18: $\mathcal{P}_{j+1} := \mathcal{P}_j + \beta_j \mathcal{Z}_{j+1}$

19: **end for**

Conclusion

- Development of parallel PCKF that exploits available Two-level domain decomposition algorithms for SPDEs.
- Distributed implementation and scalability studies of the parallel PCKF using a stationary stochastic diffusion problem.

Acknowledgment

- **Financial contributions**
  - **Natural Sciences and Engineering Research Council of Canada**
  - **Canada Research Chair Program**
  - **Canada Foundation of Innovation**
  - **Ontario Innovation Trust**