Scalable Domain Decomposition Algorithms for Uncertainty Quantification: Three-Dimensional and Time-Dependent SPDEs

Sudhi Sharma[a], Ajit Desai [a,1], Mohammed Khalil[b], Chris Pettit[c], Dominique Poirel [d], Abhijit Sarkar[a]

[a] *Department of Civil and Environmental Engineering, Carleton University, Ottawa, Ontario, Canada;* [b] *Sandia National Laboratories* [2], *Livermore, California, United States;* [c] *Aerospace Engineering Department, United States Naval Academy, Annapolis, Maryland, United States* [d] *Royal Military College of Canada, Kingston, Ontario, Canada*

[a]Currently at Bank of Canada, Ottawa, Ontario, Canada (The opinions here are of the author and do not necessarily reflect the ones of the Bank of Canada)

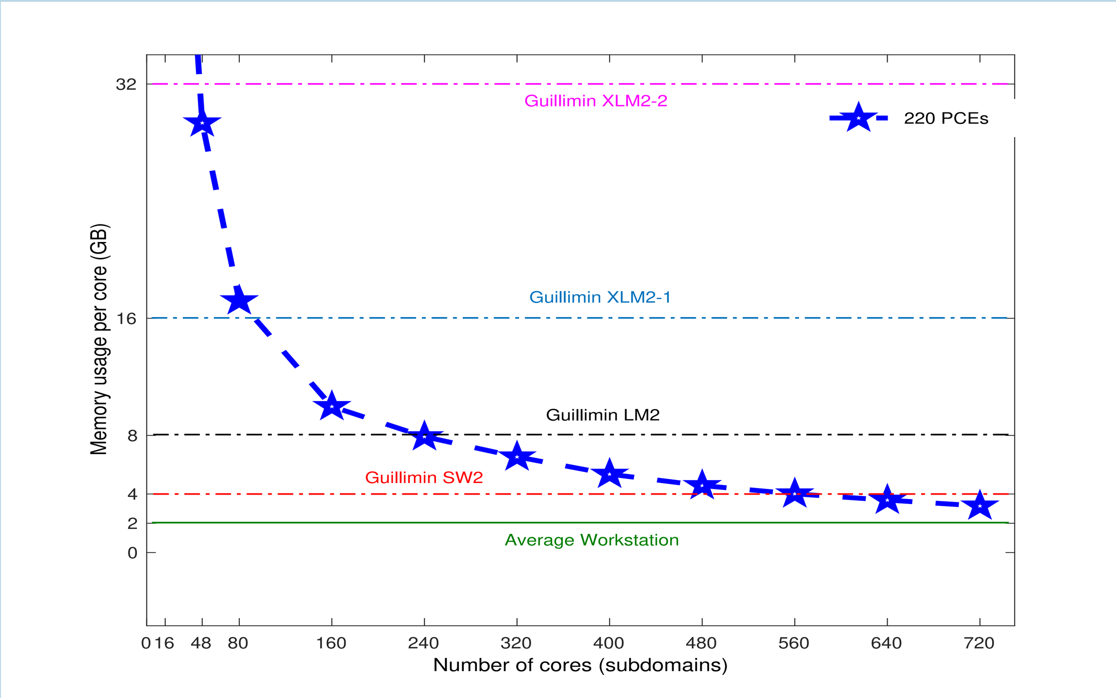
[b]Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Problem

- Stochastic Poisson Problem:

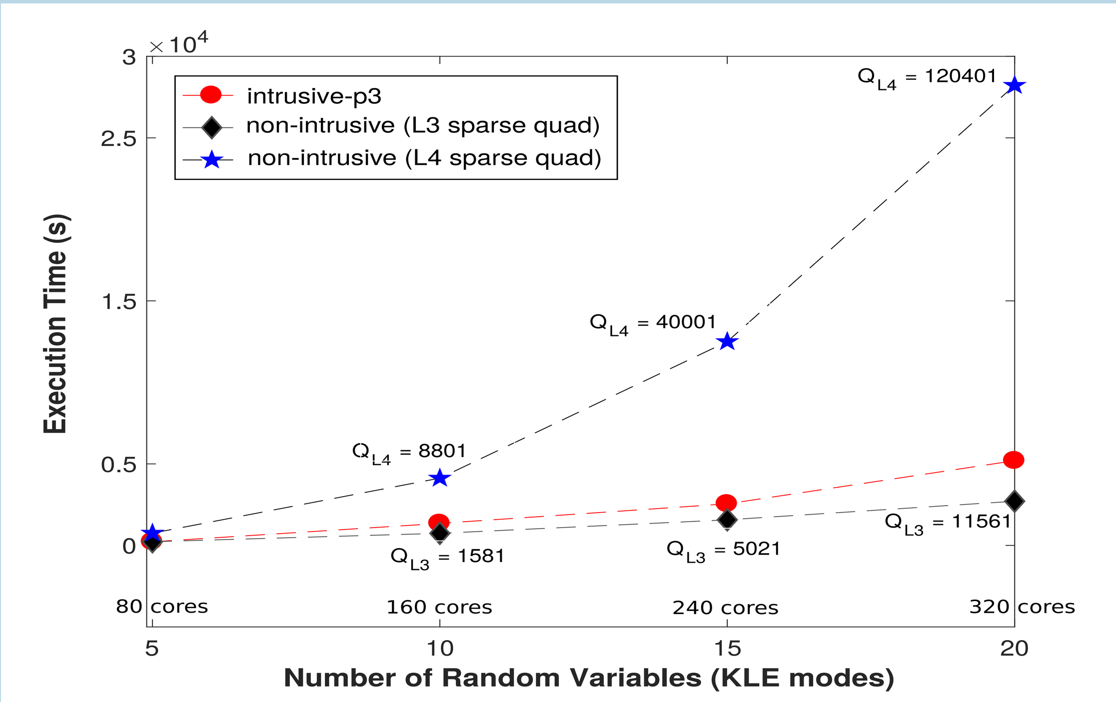
$$\begin{aligned} -\nabla \cdot \left(c_d(\boldsymbol{x},\theta) \;\; \nabla u(\boldsymbol{x},\theta) \right) &= F(\boldsymbol{x}),\ \Omega \times \mathcal{W}, \\ u(\boldsymbol{x},\theta) &= 0,\ \delta\Omega \times \mathcal{W}, \end{aligned}$$

- Diffusion Coefficient c_d modelled as log normal process
- Memory consumption for intrusive PCE(Polynomial Chaos Expansion) based method increases drastically with increase in stochastic dimension and/or order.



Intrusive Vs Non Intrusive

- With a fixed accuracy and amount of resources, non intrusive method with sparse quadrature takes longer time than intrusive for large number of random variables.



Models

- Model Problem: Equations of Linear Elasticity in 3D with Young's modulus E modeled as a log normal stochastic process.

$$\begin{aligned} -\nabla \cdot \sigma\big(\mathcal{U}(\boldsymbol{x},\theta)\big) &= F(\boldsymbol{x}) \quad in \quad \mathcal{D}, \\ \sigma\big(\mathcal{U}(\boldsymbol{x},\theta)\big)\cdot \hat{\mathbf{n}} &= b_T \quad on \quad \Gamma_1 = \delta\mathcal{D}\backslash\Gamma_0, \\ \mathcal{U}(\boldsymbol{x},\theta) &= 0 \quad on \quad \Gamma_0. \end{aligned}$$

- Acoustic Wave propagation in 2D with wave speed c_0 modeled as a log normal stochastic process.

$$\frac{\partial^2 u}{\partial t^2}(\mathbf{x},t,\theta) + \eta\frac{\partial u}{\partial t}(\mathbf{x},t,\theta) - \nabla.(c_0(\mathbf{x},\theta)\nabla \mathbf{u}(\mathbf{x},\mathbf{t},\theta) = \mathbf{f}(\mathbf{x},\mathbf{t})$$

$$u(\mathbf{x},0,\theta) = u_0(\mathbf{x})$$

References

[1] Desai, Ajit and Khalil, Mohammad and Pettit, Chris and Poirel, Dominique and Sarkar, Abhijit: *Scalable domain decomposition solvers for stochastic PDEs in high performance computing*, Computer Methods in Applied Mechanics and Engineering (2018)

Domain Decomposition for SPDEs

The domain is decomposed in to several subdomains with their corresponding interface and interior nodes. Each subdomains with their stochastic coefficient matrix and output vector satisfy equilibrium. Input $A(\theta)$ and output $\mathbf{u}(\theta)$ are expanded using PCE and combined to get a large coupled set of equations for the coefficients. This intractable global stochastic system is solved using a two level non-overlapping domain decomposition technique.

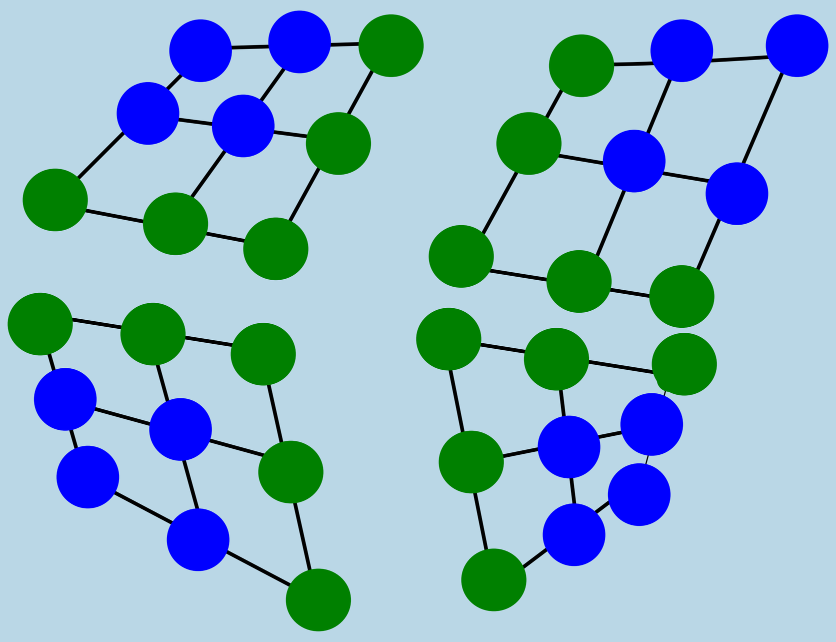


Figure 1: Four subdomains with corresponding interface and interior nodes

- 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

$$\begin{bmatrix} \mathcal{A}^1_{II} & \dots & 0 & \mathcal{A}^1_{I\Gamma}\mathcal{R}_1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & \mathcal{A}^{n_s}_{II} & \mathcal{A}^{n_s}_{I\Gamma}\mathcal{R}_{n_s} \\ \mathcal{R}^T_1\mathcal{A}^1_{\Gamma I} & \dots & \mathcal{R}^T_{n_s}\mathcal{A}^{n_s}_{\Gamma I} & \sum\limits_{s=1}^{n_s}\mathcal{R}^T_s\mathcal{A}^s_{\Gamma\Gamma}\mathcal{R}_s \end{bmatrix} \begin{Bmatrix} \mathcal{U}^1_I \\ \vdots \\ \mathcal{U}^{n_s}_I \\ \mathcal{U}_\Gamma \end{Bmatrix} = \begin{Bmatrix} \mathcal{F}^1_I \\ \vdots \\ \mathcal{F}^{n_s}_I \\ \sum\limits_{s=1}^{n_s}\mathcal{R}^T_s\mathcal{F}^s_\Gamma \end{Bmatrix},$$

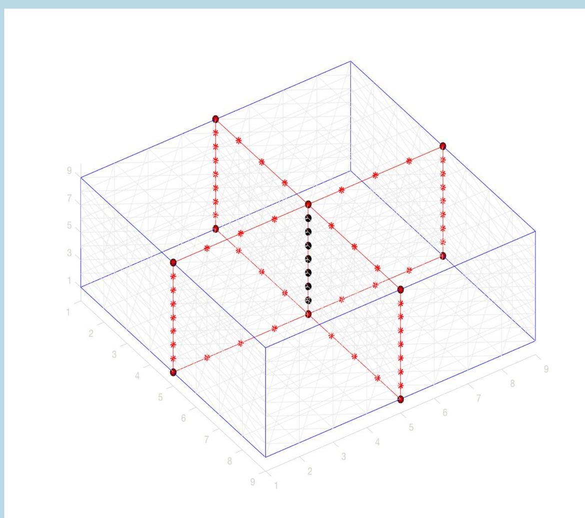
where

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

Wire Basket Grid for 3D Poisson and Elasticity

Figure 2: Wire basket grid with interface edge nodes (red dots) and vertex nodes (black dots). All grey dots are face nodes

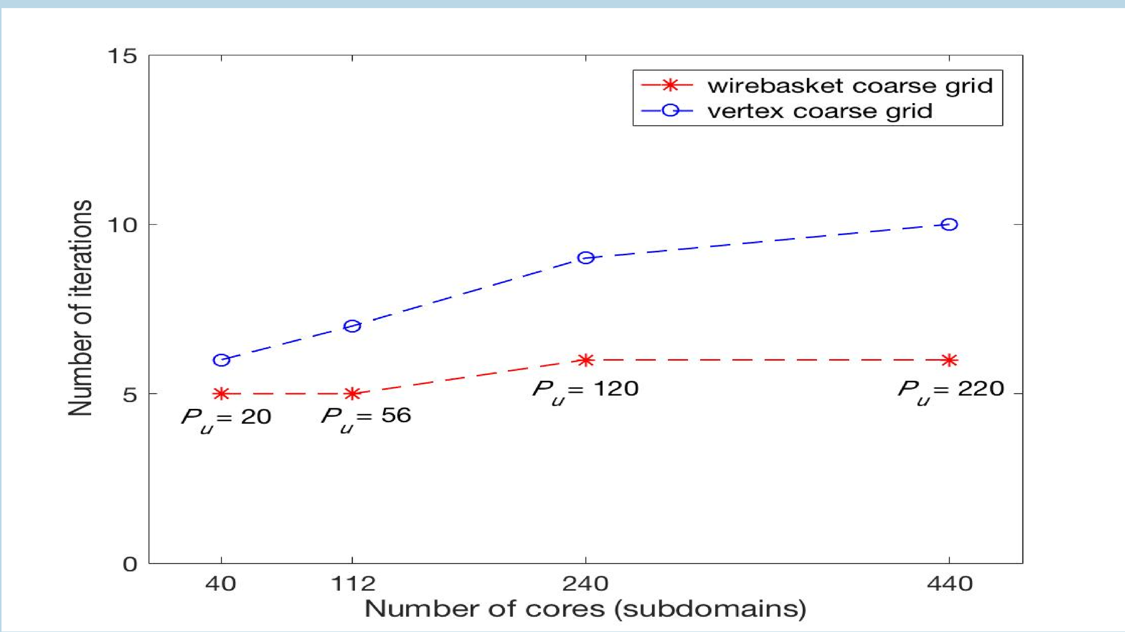
Wire basket coarse grid consists of all the vertex nodes and interface edges nodes (W). All the remaining nodes on the interface are grouped as face nodes (F). This coarse grid allows for better global error propagation in three dimension compared to vertex based coarse grid reducing the number of iterations. The preconditioner is set up consisting of a fine grid with face nodes and coarse grid with wire basket as below.

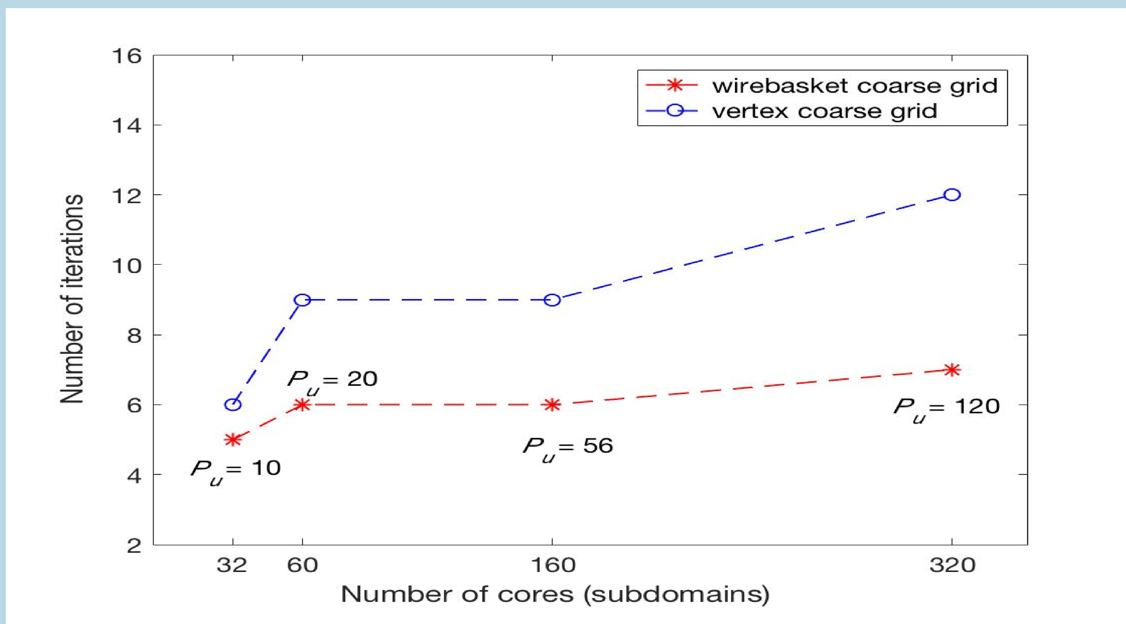


$$\mathcal{M}^{-1}_{NNW} = \sum_{s=1}^{n_s}\mathcal{R}^{\mathrm{T}}_s\mathcal{D}_s(\mathcal{R}^{F}_s{}^{\mathrm{T}}[\mathcal{S}^s_{FF}]^{-1}\mathcal{R}^F_s)\mathcal{D}_s\mathcal{R}_s + \mathcal{R}^{\mathrm{T}}_0[\mathcal{F}_{WW}]^{-1}\mathcal{R}_0$$

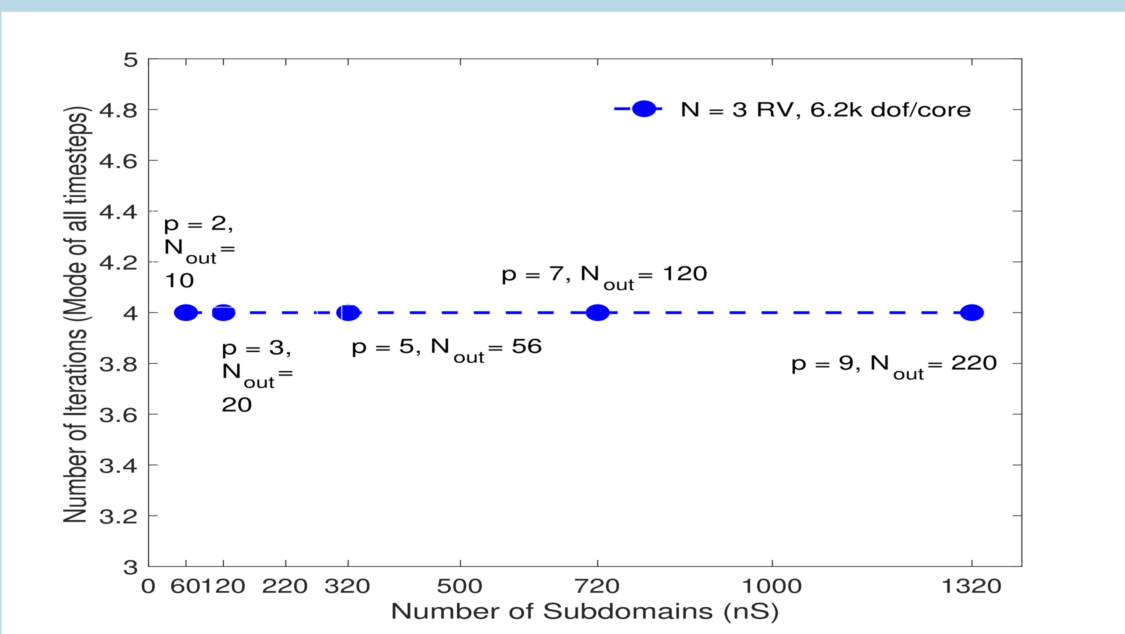
$$\mathcal{R}_0 = \sum_{s=1}^{n_s}\mathcal{B}^s_W{}^{\mathrm{T}}(\mathcal{R}^W_s - \mathcal{S}^s_{WF}[\mathcal{S}^s_{FF}]^{-1}\mathcal{R}^F_s)\mathcal{D}_s\mathcal{R}_s$$

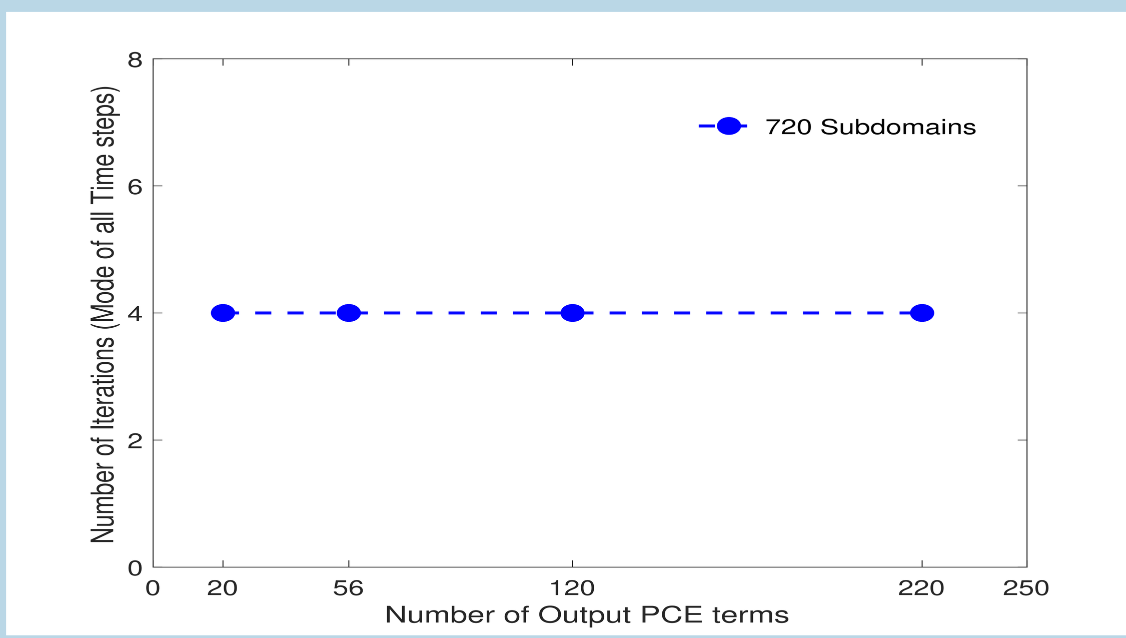
$$\mathcal{F}_{WW} = \sum_{s=1}^{n_s}\mathcal{B}^s_W{}^{\mathrm{T}}\Big(\mathcal{S}^s_{WW} - \mathcal{S}^s_{WF}[\mathcal{S}^s_{FF}]^{-1}\mathcal{S}^s_{FW}\Big)\mathcal{B}^s_W$$





Acoustic Wave Propagation





- Time dependant problem is much more complex in terms of the memory and time consumption but has a better conditioned coefficient matrix than static case.
- Scalability of solver for a fixed amount of workload/core and for fixed amount of resources with increasing output PC terms is shown.