An alternative to coarse spaces? Piecewise Krylov Methods

Kévin Santugini

22nd International Conference on Domain Decomposition Methods
Lugano, September 16th–20th, 2013

Importance of scalable algorithms

Massive parallelism

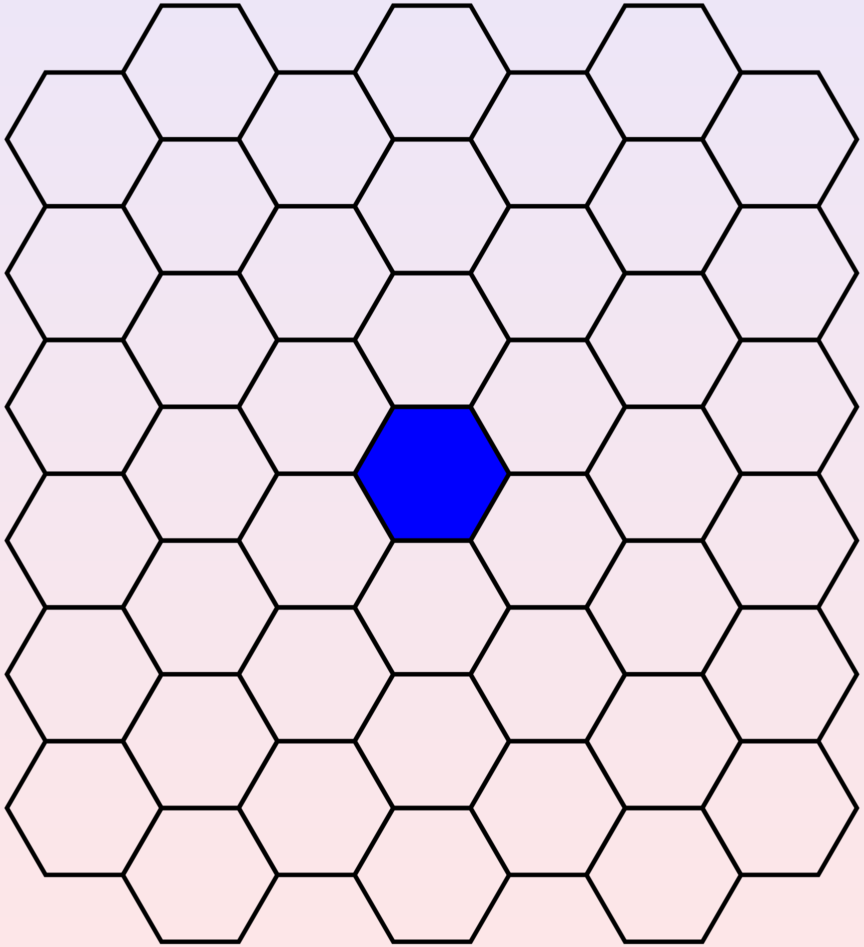
Off the shelf clusters with more than 1000 cores are available.
Scalability is no longer optional.

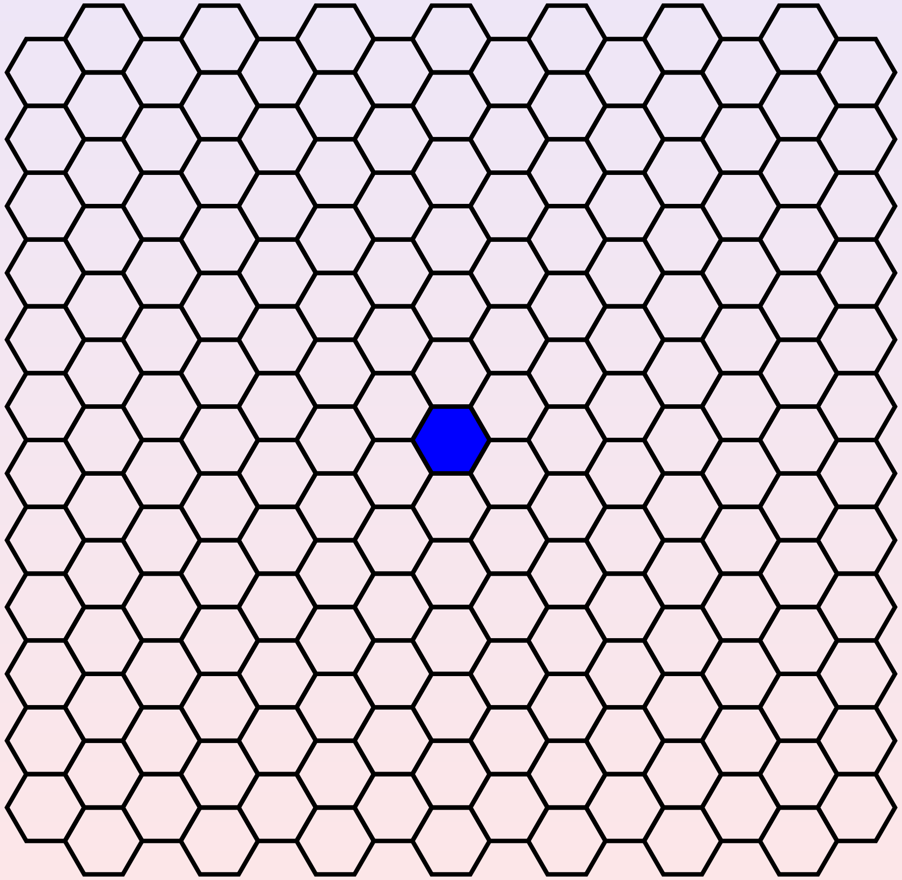
Scalability

Strong scalability: For a given problem size, the computation time goes down as the number of computation units increase.

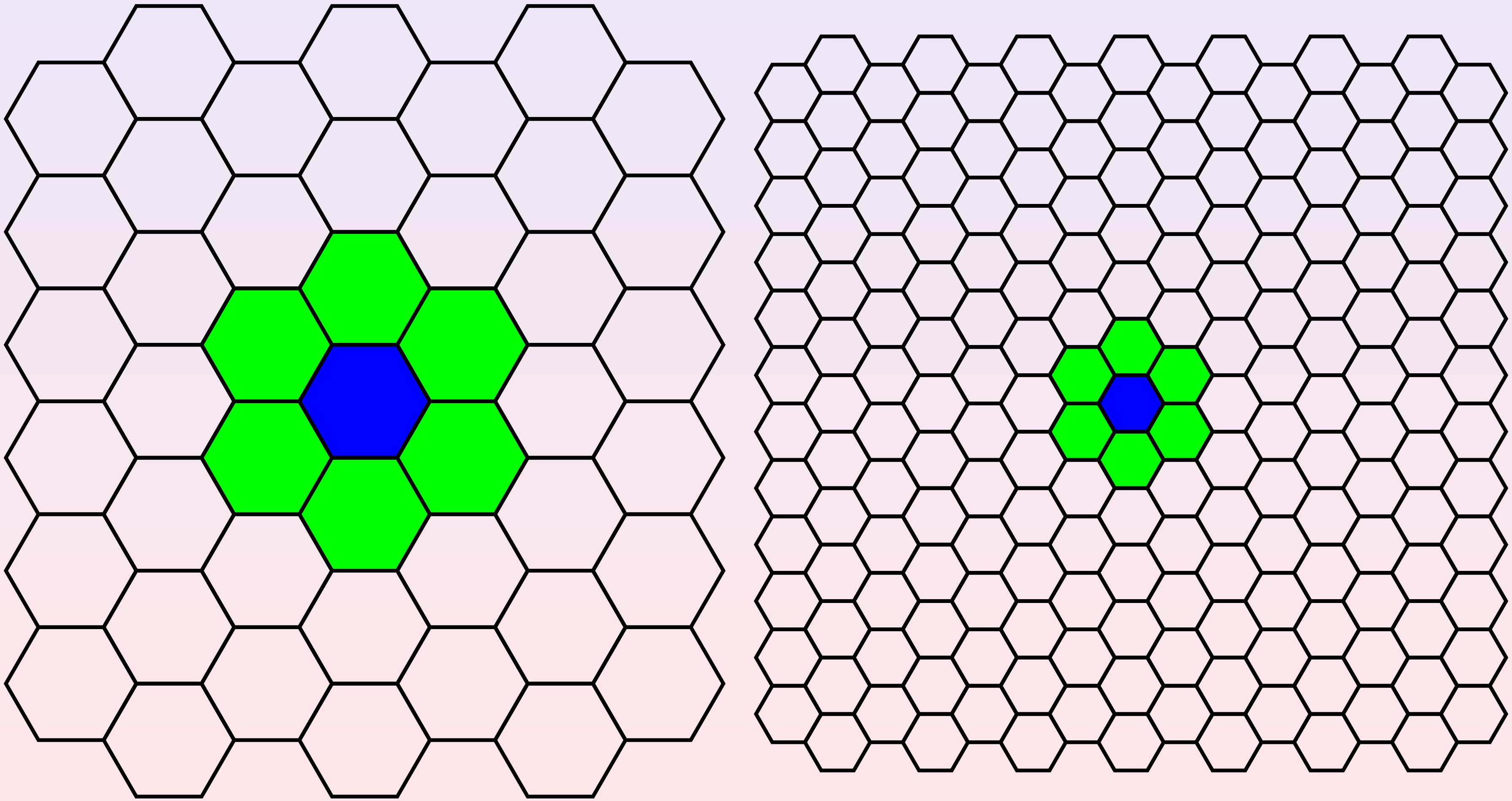
Weak scalability: More computation units allows to solve bigger problems in the same amount of time.

Non Scalability of one-level DDM

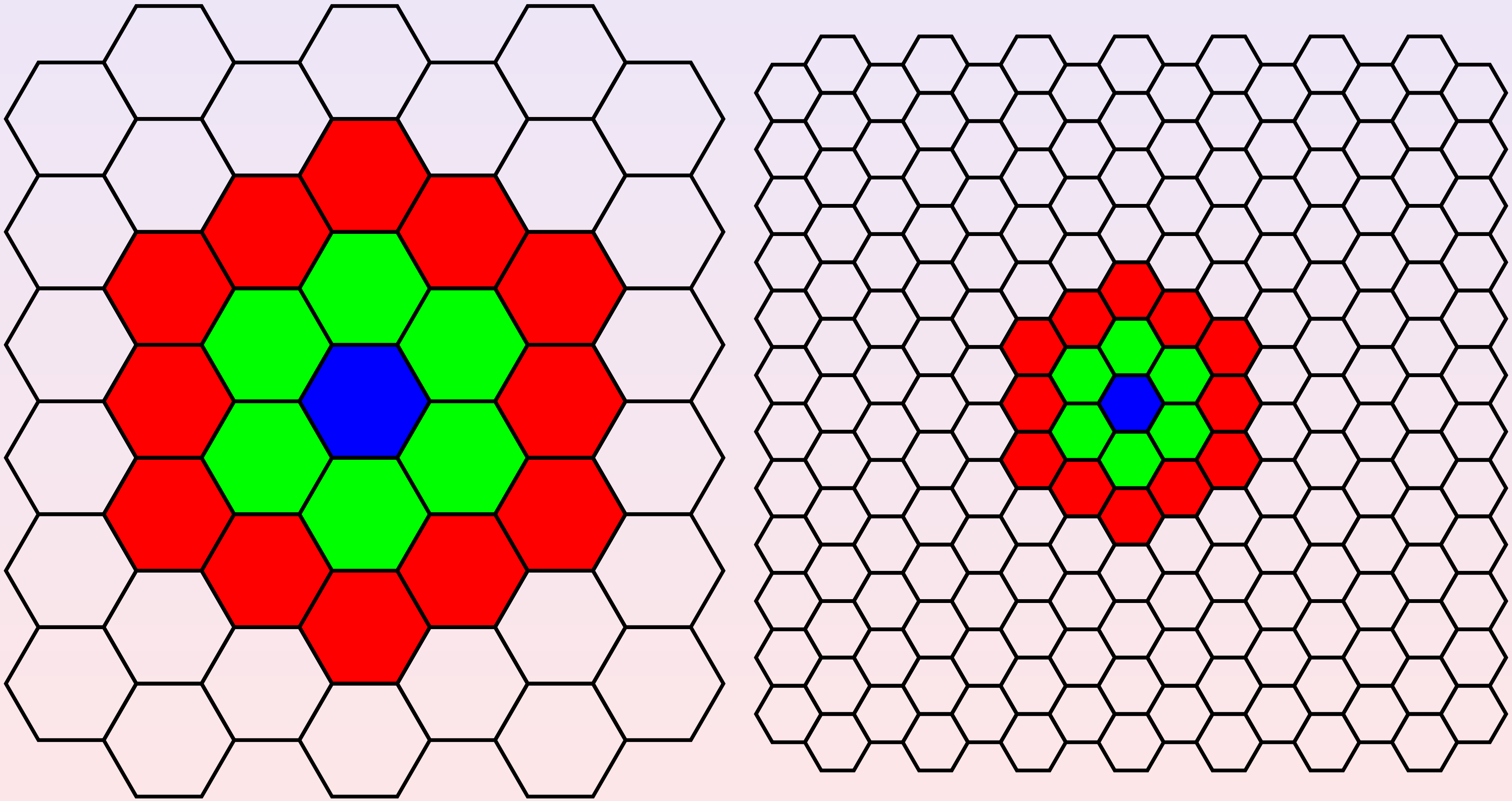




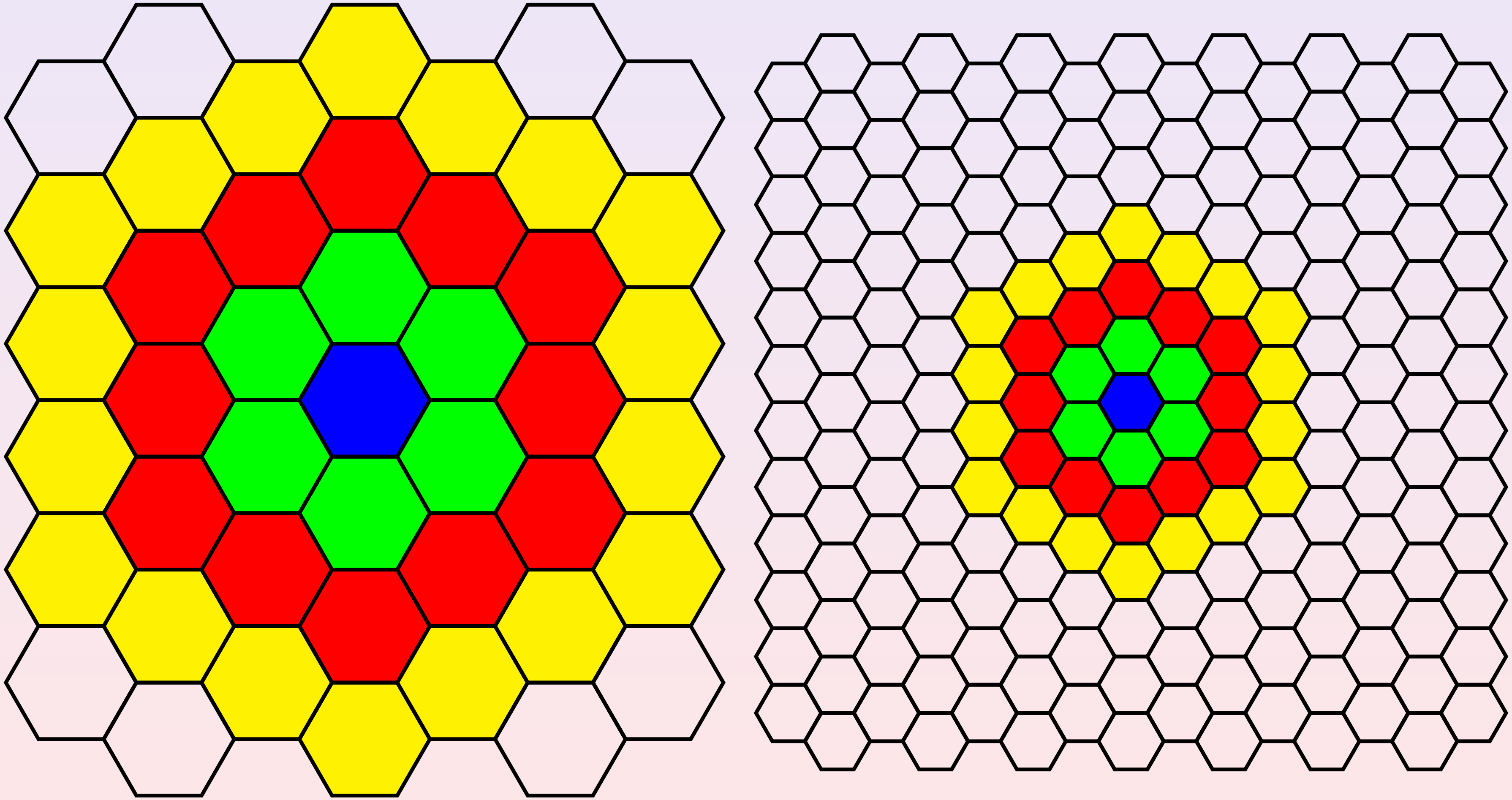
Non Scalability of one-level DDM



Non Scalability of one-level DDM



Non Scalability of one-level DDM



The standard way of getting scalability: coarse spaces

Choose a coarse space X

1. Either set u_i^0 to 0 or to the coarse solution.
2. Until convergence
   1. Compute the uncorrected iterates $u_i^{n+1/2}$ using Optimized Schwarz.

$$\mathcal{L}u_i^{n+1/2} = f \quad \text{in } \Omega_i$$

$$\mathcal{B}_{ij}u_i^{n+1/2} = \mathcal{B}_{ij}u_j^n \quad \text{on } \partial\Omega_i \cap \partial\Omega_j$$

$$u_i^{n+1/2} = g \quad \text{on } \partial\Omega_i \cap \partial\Omega$$

   2. Compute in some way a coarse correction U^{n+1} in X defined over Ω, then compute the corrected iterates

$$u_i^{n+1} := u_i^{n+1/2} + U^{n+1}{}_{|\Omega_i}$$

Only applying the coarse correction every few iterates

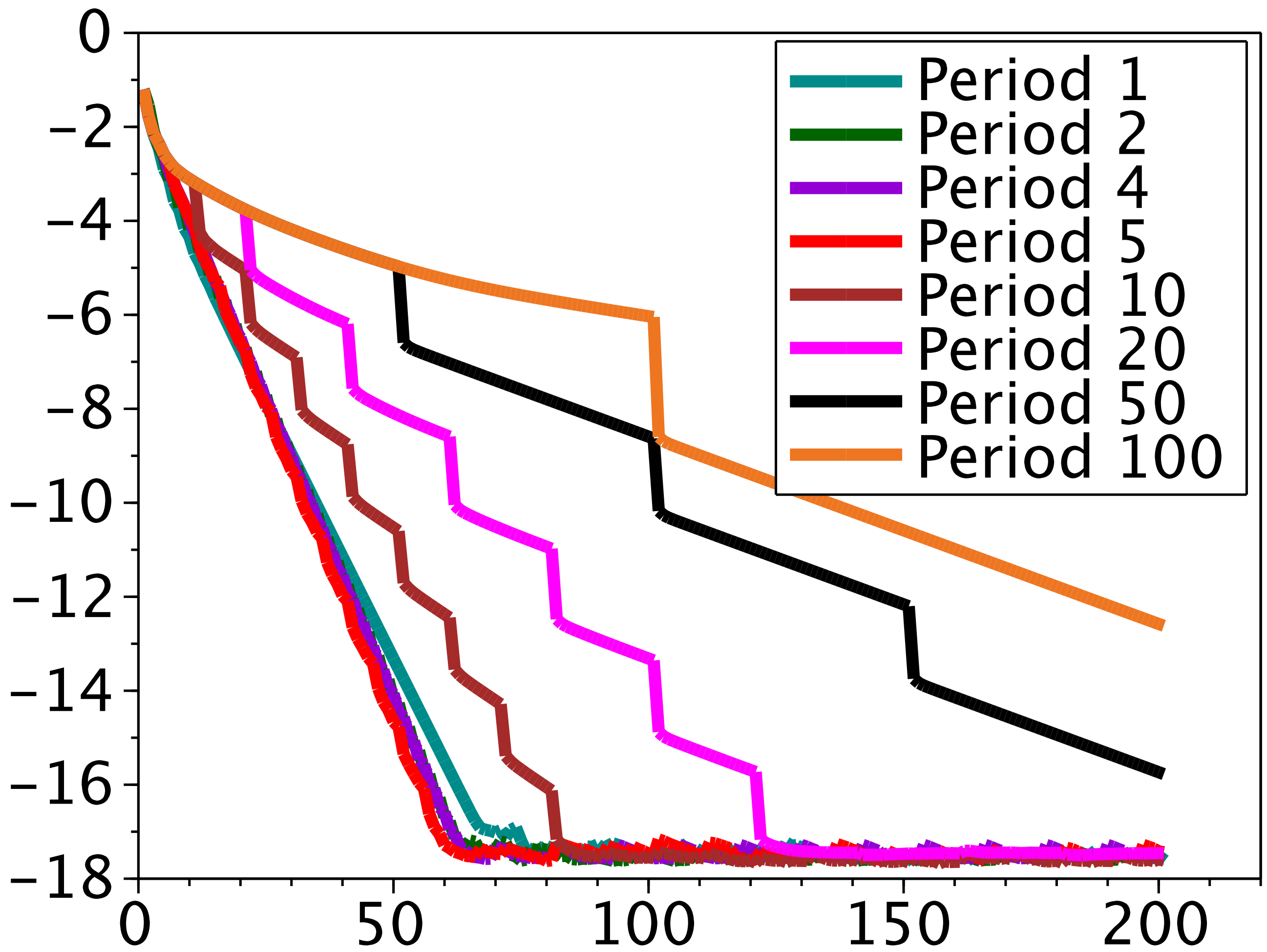


Figure : DCS-DMNV Algorithm presented during the DD21

Stopping coarse correction after 20 iterates

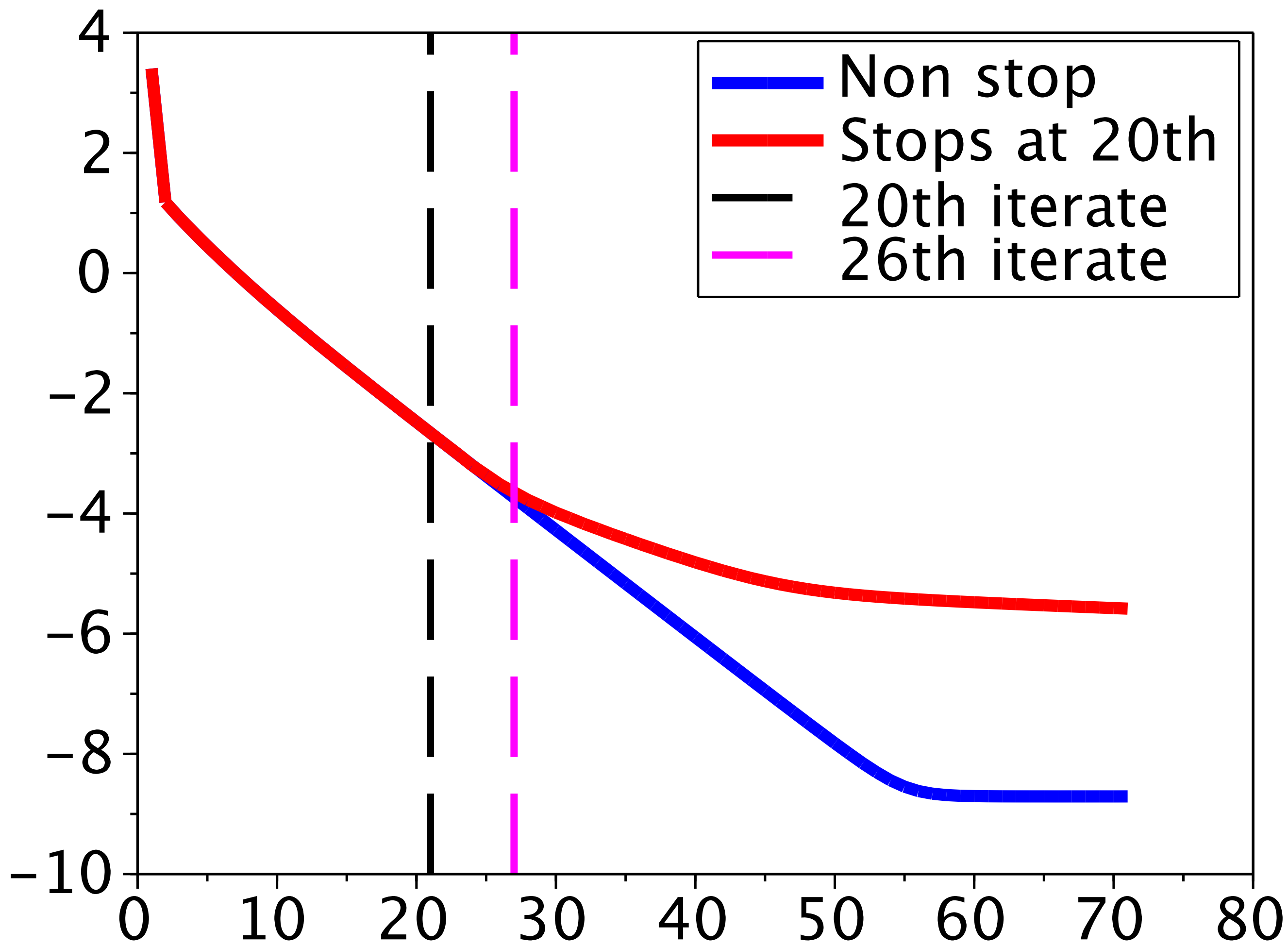


Figure : DCS-DMNV Algorithm presented during the DD21

Changing the coarse space every iterate

Observations

1. No need to apply coarse correction every iterate.
2. Using the same coarse space every iterate: always correcting the same errors.

Coarse functions should

1. Satisfy the interior equation inside each subdomain.
2. Should be discontinuous.

Cheap coarse spaces

If

$$\mathcal{L}(u^n) = f \quad \text{for all } n.$$

Then

$$\mathcal{L}(u^{n+1} - u^n) = 0 \quad \text{for all } n.$$

1. Use the successive $u^{n+1} - u^n$ as coarse functions.
2. Or use
$$\begin{cases} u_i^{n+1} - u_i^n & \text{in } \Omega_i, \\ 0 & \text{in } \Omega_j \text{ when } j \neq i, \end{cases}$$
as coarse functions.

Acceleration of Domain decomposition methods

Iterative methods are not used standalone in practice.

Use of DDM as preconditioners

- Discrete/continuous linear differential operator $\mathbf{A}_h$.
- $\mathbf{A}_h$ is sparse. Its inverse is not.
- "Inverting" in parallel on each subdomain a restriction of $\mathbf{A}_i$ to Ω_i.
- Define the preconditioner as a combination of the inverse matrices computed in parallel and get the preconditioner $\mathbf{P}$.

Iterative method : Richardson on the preconditioned operator $\mathbf{PA}_h$.

Krylov methods as extrapolation methods

Extrapolation methods

Find "best" λ_k^n

$$u_{\mathcal{K}}^n = \sum_{k=0}^{n} \lambda_k^n u^k, \qquad \sum_{k=0}^{n} \lambda_k^n = 1.$$

Key properties

1. In **exact** arithmetic, Krylov methods equivalent to extrapolation methods.
2. In **floating point** arithmetic, Krylov methods are much more stable than extrapolation methods.
3. $u_{\mathcal{K}}^n$ satisfy the interior equation inside every subdomain

Piecewise extrapolation

If λ_k^n depended on the subdomain then .

Extrapolation methods

Find "best" $\lambda_{i,k}^n$

$$u_{\mathcal{K}}^n = \sum_{i=0}^{n} \lambda_{i,k}^n u_i^k, \qquad \sum_{k=0}^{n} \lambda_{i,k}^n = 1.$$

Per-subdomain λ parameters in extrapolation replaces coarse spaces for scalability. In practice too many parameters.

Piecewise extrapolation with Robin jump minimizer

1. Set an initial guess.
2. Until convergence:
   1. Set $u_i^{n+1/2}$ as the unique solution to

$$\eta u_i^{n+1/2} - \triangle u_i^{n+1/2} = f \text{ in } \Omega_i,$$
$$\frac{\partial u_i^{n+1/2}}{\partial \mathbf{n}_i} + p u_i^{n+1/2} = \frac{\partial u_j^n}{\partial \mathbf{n}_i} + p u_j^n \text{ on } \partial\Omega_i \cap \partial\Omega_j,$$
$$u_i^{n+1/2} = 0 \text{ on } \partial\Omega_i \cap \partial\Omega.$$

   2. Set $u_i^{n+1} := u_i^{n+1/2} + \sum_{k=0}^{n} \lambda_{i,k}^{n+1}(u_i^{k+1/2} - u_i^k)$ such that

$$\sum_{ij} \int_{\Gamma_{ij}} \left| \left(\frac{\partial u_i^{n+1}}{\partial \mathbf{n}_i} + q u_i^{n+1} \right) - \left(\frac{\partial u_j^{n+1}}{\partial \mathbf{n}_i} + q u_j^{n+1} \right) \right|^2$$

is minimized where q is another Robion parameter.

Richardson with Piecewise line search algorithm

Use only a single direction per subdomain: $u_i^{n+1/2} - u_i^n$.

1. Set an initial guess.
2. Until convergence:
   1. Set $u_i^{n+1/2}$ using Optimized Schwarz.
   2. Compute N scalars λ_i^{n+1} and set $u_i^{n+1} := (1-\lambda_i^{n+1})u_i^{n+1/2} + \lambda_i^{n+1} u_i^n$ such that

$$\sum_{ij} \int_{\Gamma_{ij}} \left| \left(\frac{\partial u_i^{n+1}}{\partial \mathbf{n}_i} + p u_i^{n+1} \right) - \left(\frac{\partial u_j^{n+1}}{\partial \mathbf{n}_i} + p u_j^{n+1} \right) \right|^2$$

is minimized.

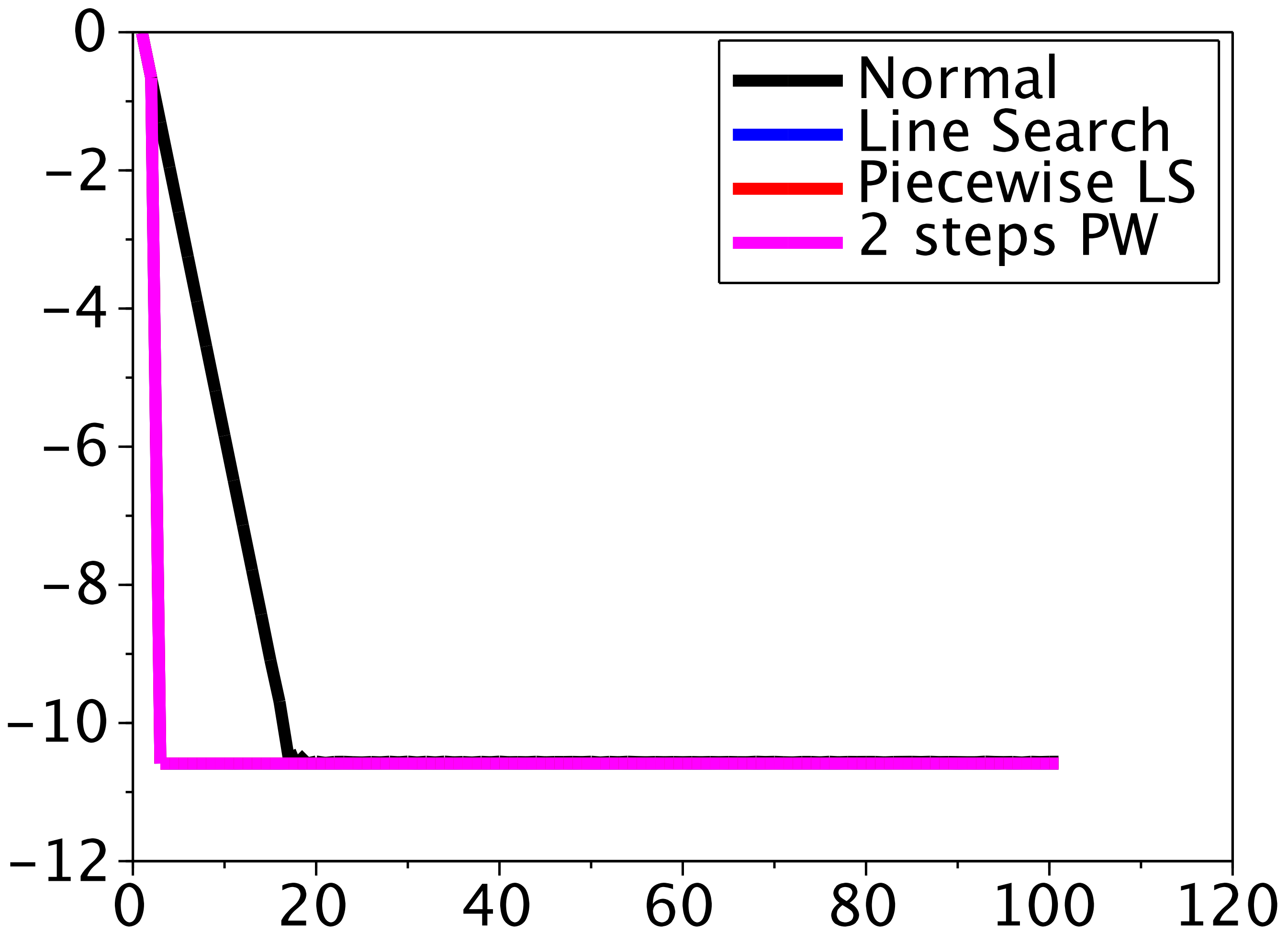
Full Robin Minimizing Piecewise Extrapolation algorithm

Use only two directions per subdomain: $u_i^{n+1/2} - u_i^n$ and $u_i^{n-1/2} - u_i^{n-1}$.

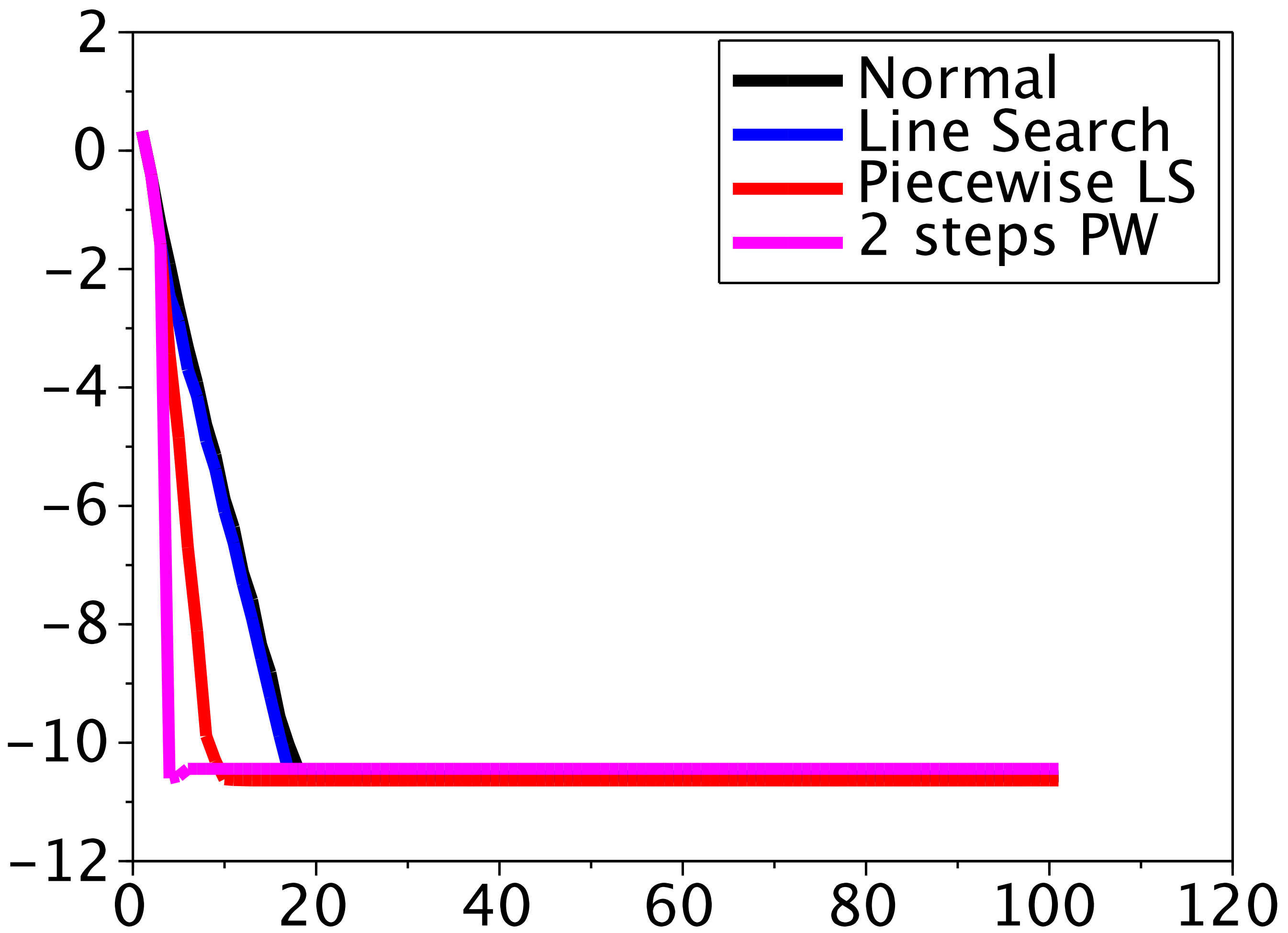
1. Set an initial guess.
2. Until convergence:
   1. Set $u_i^{n+1/2}$ using optimized Schwarz
   2. Set $u_i^{n+1} := u_i^{n+1/2} + \sum_i \lambda_i^{n+1}(u_i^{n+1/2} - u_i^n)$ so as to minimize

$$\sum_{ij} \int_{\Gamma_{ij}} \left| \left(\frac{\partial u_i^{n+1}}{\partial \mathbf{n}_i} + p u_i^{n+1} \right) - \left(\frac{\partial u_j^{n+1}}{\partial \mathbf{n}_i} + p u_j^{n+1} \right) \right|^2 .$$

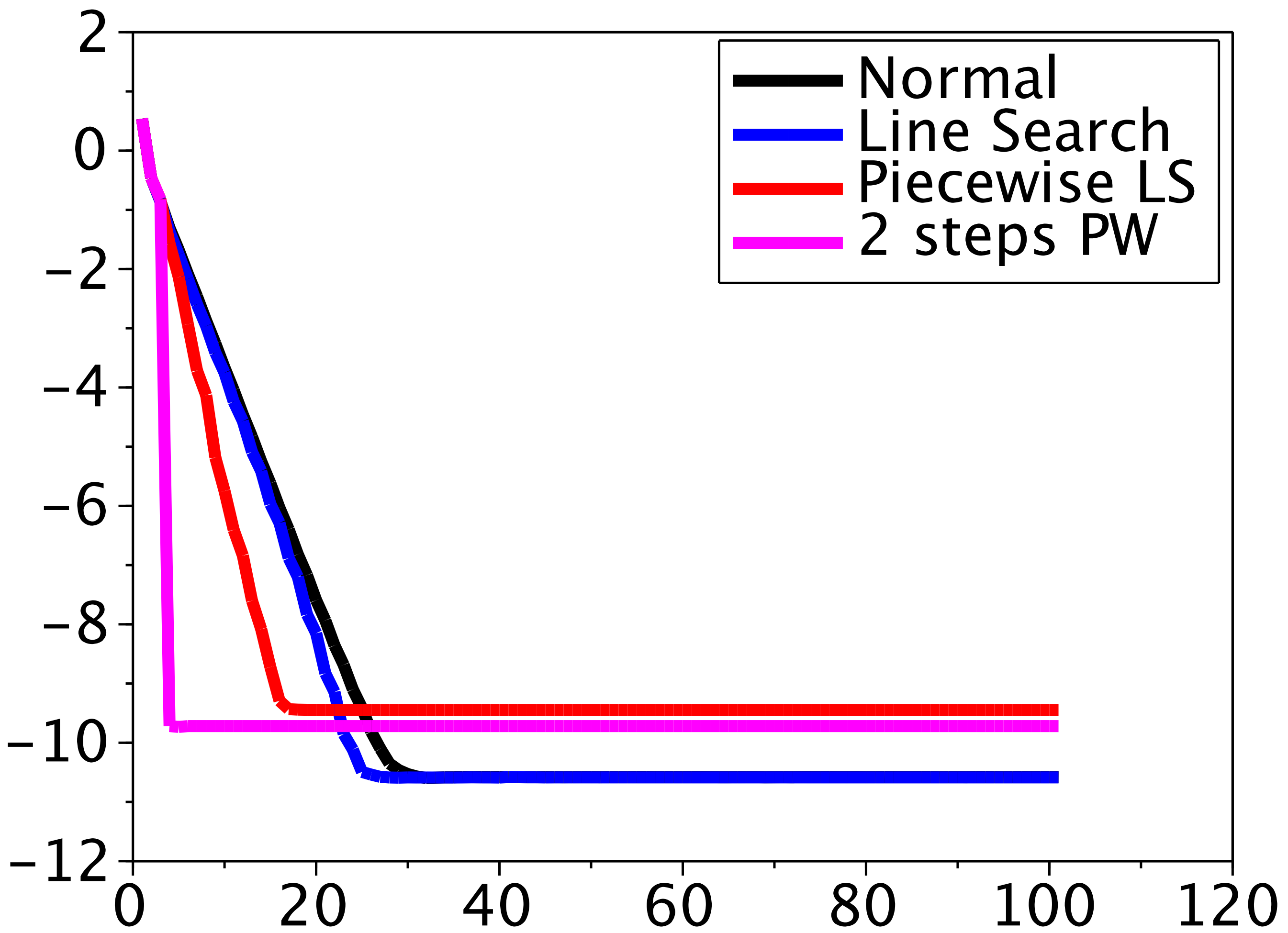
Influence of numbers of subdomains: 2 subdomains in 1D



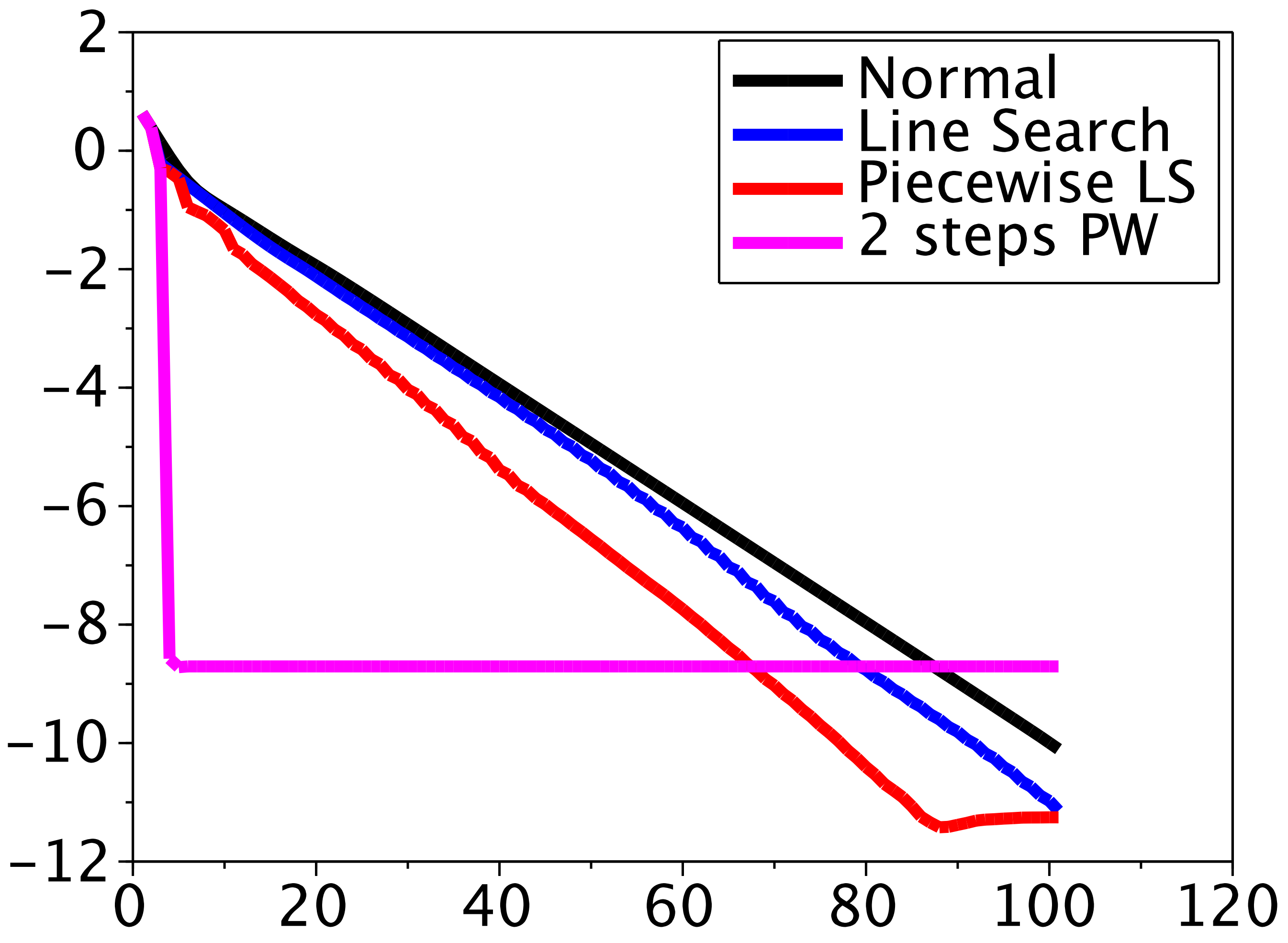
Influence of numbers of subdomains: 5 subdomains in 1D



Influence of numbers of subdomains: 10 subdomains in 1D



Influence of numbers of subdomains: 40 subdomains in $1D$



Influence of numbers of subdomains: in $2D$

Conclusion

1. Rationales for exploring Piecewise Krylov.
2. First numerical simulations in $1d$ and $2d$.
3. Piecewise extrapolation works better than extrapolation on tested algorithms.
4. Piecewise extrapolation don't remove the need for a coarse space.

Future works

1. Implementing Krylov instead of extrapolation (better numerical stability).
2. Elegant Piecewise GMRES using only piecewise Arnoldi coefficients?
3. Comparing fixed coarse spaces methods and piecewise Krylov.
4. Use both Discontinuous coarse spaces and piecewise Krylov.