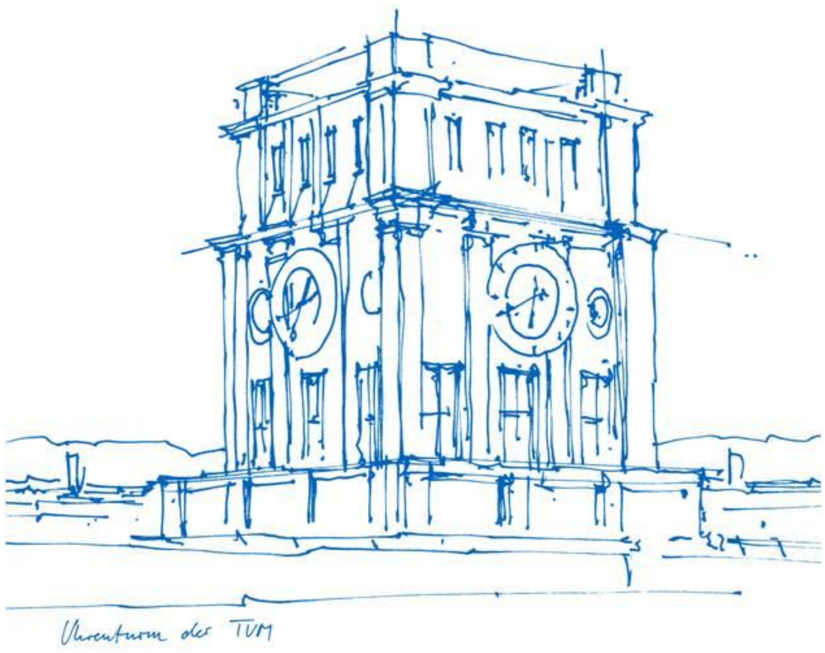
A Survey of Mixed Precision Multigrid Methods

Technical University of Munich

TUM School of Computation, Information and Technology

Jinwen Pan

07.07.2023



Content

Mixed Precision Algorithms

Motivation:

-> exponential growth of computational amount and increasing demand for speed
-> traditional high precision algorithms often become computationally prohibitive
-> balance between accuracy and computational efficientcy is needed

Definition:

Mixed precision algorithms: employ two or more precisions, selected from a limited set of available precisions (half, single, double by hardware and quadruple by software).

Multiprecision or variable precision algorithms: utilize one or more arbitrary precisions, which can vary based on the specific problem and are implemented through software.

Mixed Precision Algorithms

Accuracy:

- higher precision (represented by more bits) generally leads to higher accuracy
- computations involving extremely large or minuscule values, as well as those with subtle differences, will be more accurately captured.

Runtime:

- typically consists of data communication and computation
- data communication: a hardware-independent linear correlation with length of precision
- computational overhead: largely contingent upon the specific hardware architectures

Implementation:

Substitute high precision with low precision in the parts which are performance-sensitive but not error-sensitive.

Mixed Precision Algorithms

Physical simulations:

- PDE-based models: molecular dynamics simulation, computational fluid dynamics, computational electromagnetics
- most common and computationally intensive

Climate modelling and weather forecasting:

- inclination towards utilization of low precision since 2014
- low precision is sufficient to be consistent with observations on which a model is constructed

Machine learning:

- mixed precision training optimizer, such as the NVIDIA Apex library
- hardware accelerators that support mixed precision computations, such as NVIDIA's Tensor Cores

Multigrid Methods

- solve a linear system of equations (LSE) arising from PDEs
- compute across multiple grids of different resolutions
- **prolongation:** interpolate coarse grid to finer grid
- **restriction:** reduce fine grid to coarser grid
- **relaxation:** arbitrary LSE iterative solver

Multigrid Methods

Geometric multigrid methods:

- operate directly on a hierarchy of grids that are nested within each other
- further categorized: V-cycle, W-cycle, or Full Multigrid (FMG).

Algebraic multigrid methods:

- operate directly on the LSE derived from the discretization of the PDE
- construct a hierarchy of approximations based on the algebraic structure of the problem
- start with an initial approximation to the solution using coarse grid corrections

Multigrid Methods

Advantages:

- **Convergence**: exploit the multilevel structure to capture error components at different scales, leading to faster convergence rates
- **Scalability**: exhibit excellent scalability with problem size
- **Flexibility**: can be applied to structured, unstructured, and adaptive grids

Mixed precision algorithms:

- employ pure low precision multigrid methods as preconditioners for high precision solvers
- utilize different precisions at each level of the multigrid hierarchy

IEEE 754 Floating-Point Number Systems

Definition:

$$e \in \{e_{min}, \ldots, e_{max}\}$$

$$d_0, \ldots, d_{p-1} \in \{0, \ldots, b-1\}$$

$$x = \pm b^e \times \left(d_0 + \frac{d_1}{b} + \frac{d_2}{b^2} + \ldots + \frac{d_{p-1}}{b^{p-1}} \right)$$

Alternatively:

$$m \in \{0, b, \ldots, b^p - 1\}$$

$$x = \pm m \times b^{e-p+1}$$

Normalization:

$$d_0 \in \{1, \ldots, b-1\}$$

$$m \in \{b^{p-1}, b^{p-1} + b, \ldots, b^p - 1\}$$

Range:

$$b^{e_{min}} \leq |x| \leq b^{e_{max}} \times (b - b^{1-p})$$

IEEE 754 Floating-Point Number Systems

Format	Sign	Exponent	Significand	e_{min}	e_{max}	Machine Epsilon	x_{min}	x_{max}
FP16	1 bit	5 bits	10 bits	-14	+15	9.76×10^{-4}	6.10×10^{-5}	6.55×10^{4}
FP32	1 bit	8 bits	23 bits	-126	+127	1.19×10^{-7}	1.18×10^{-38}	3.40×10^{38}
FP64	1 bit	11 bits	52 bits	-1022	+1023	2.22×10^{-16}	2.22×10^{-308}	1.80×10^{308}
FP128	1 bit	15 bits	112 bits	-16382	+16383	1.93×10^{-34}	3.36×10^{-4932}	1.19×10^{4932}

Machine epsilon:

$$\epsilon = b^{1-p}$$

Unit roundoff:

$$u = \frac{1}{2} b^{1-p}$$

Subnormal numbers:

$$e = e^{e_{min}}$$
$$d_0 = 0$$
$$m \in \{0, b, ..., b^{p-1} - 1\}$$

Rounding Error Analysis Model

Floating-point estimates:

$$fl(x) = x(1+\delta), \ |\delta| \le u$$

Binary operations:

$$fl(x \star y) = (x \star y)(1+\delta), \ |\delta| \le u$$

Linear transformation:

$$fl(Ax) = Ax + \delta, \ |\delta| \le \frac{nu}{1-nu}|A| \cdot |x|$$

Residual calculation:

$$fl(Ax-b) = Ax - b + \delta, \ |\delta| \le \frac{(n+1)u}{1-(n+1)u}(|b| + |A| \cdot |x|)$$

$$\mathrm{fl}(Ax-b) = Ax - b + \delta, \quad |\delta| \le \varepsilon|Ax-b| + (1+\varepsilon)\frac{(n+1)\bar{\varepsilon}}{1-(n+1)\bar{\varepsilon}}(|b| + |A| \cdot |x|)$$

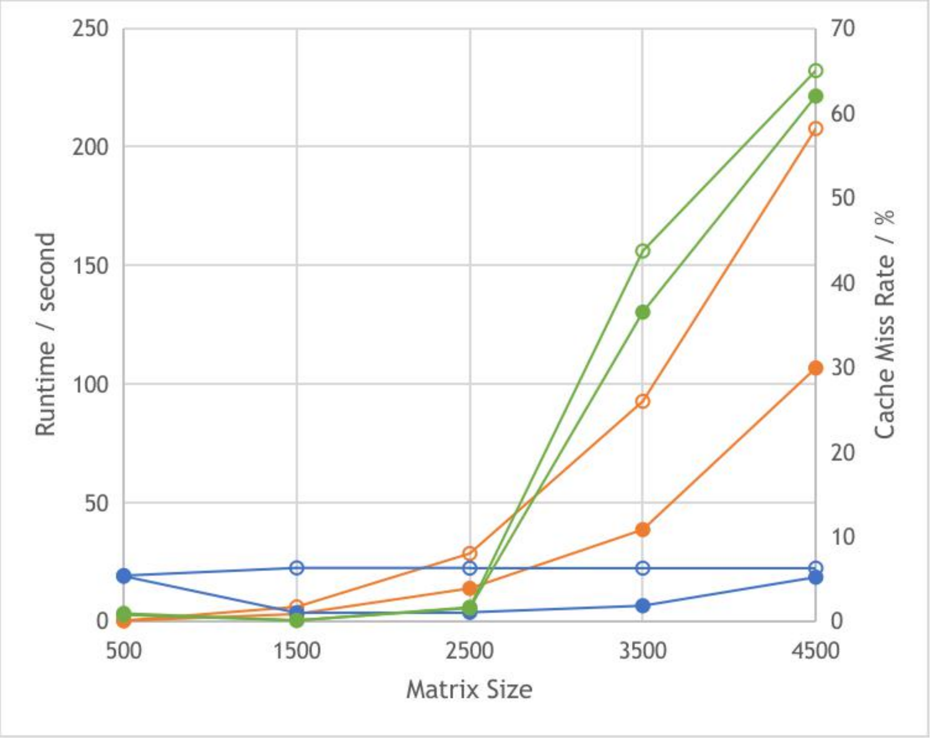
SuperMUC-NG

- Peak performance: 26.8 Peta FLOPS
- 9 islands, 6480 nodes
- 2 sockets per node
- 24 cores per socket (48 threads including hyperthreads per socket)
- Skylake EP: Intel Xeon 8174
- Frequencies:
  - Standard 2.7 GHz
  - Nominal 3.1 GHz
  - Peak 3.9 GHz
  - AVX 2.3 GHz
- Memory:
  - L1 instruction: 32KB
  - L1 data: 32KB
  - L2: 1MB
  - L3: 1.3MB / core, non-inclusive victim cache
  - DRAM: 96GB per node, aggregated bandwidth 128 GB / s

Matrix-Matrix Multiplication - Cache Optimization

```
for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        for (int k = 0; k < n; k++) {
            C[i * n + j] += A[i * n + k] * B
                [k * n + j];
        }
    }
}
```

```
for (int i = 0; i < n; i++) {
    for (int k = 0; k < n; k++) {
        for (int j = 0; j < n; j++) {
            C[i * n + j] += A[i * n + k] * B
                [k * n + j];
        }
    }
}
```



orange: runtime
blue: L2 miss rate
green: L3 miss rate

empty symbols: original
filled symbols: optimized
single precision and -O1

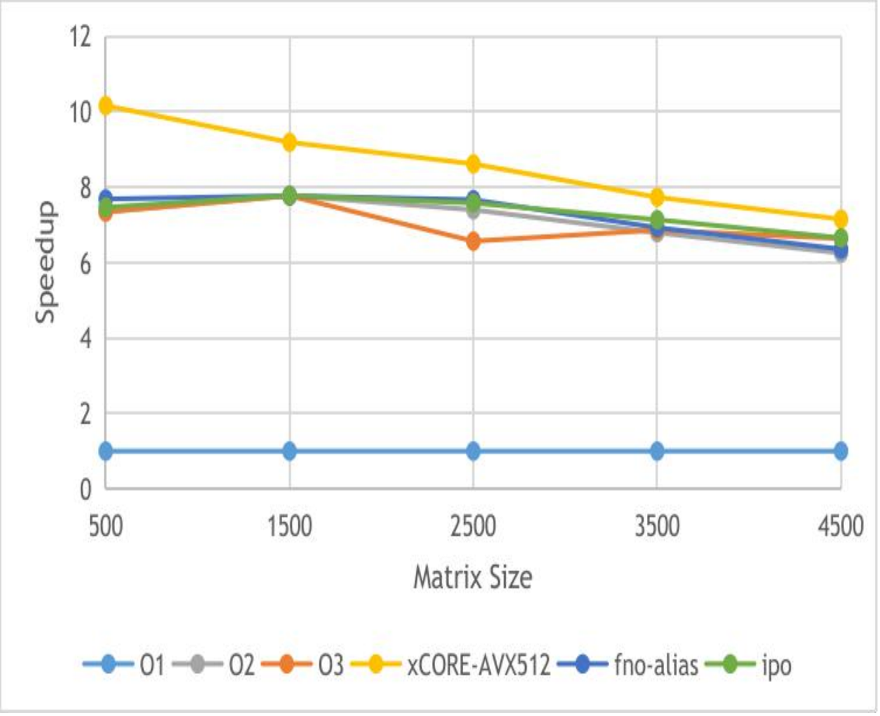
Matrix-Matrix Multiplication - Compiler Optimization

- -O1, -O2, and -O3: progressively enhanced optimization levels. The latter two enables SIMD vectorization. -O2 is used when no one is specified.

- -ipo: interprocedural optimization: perform optimization across different functions

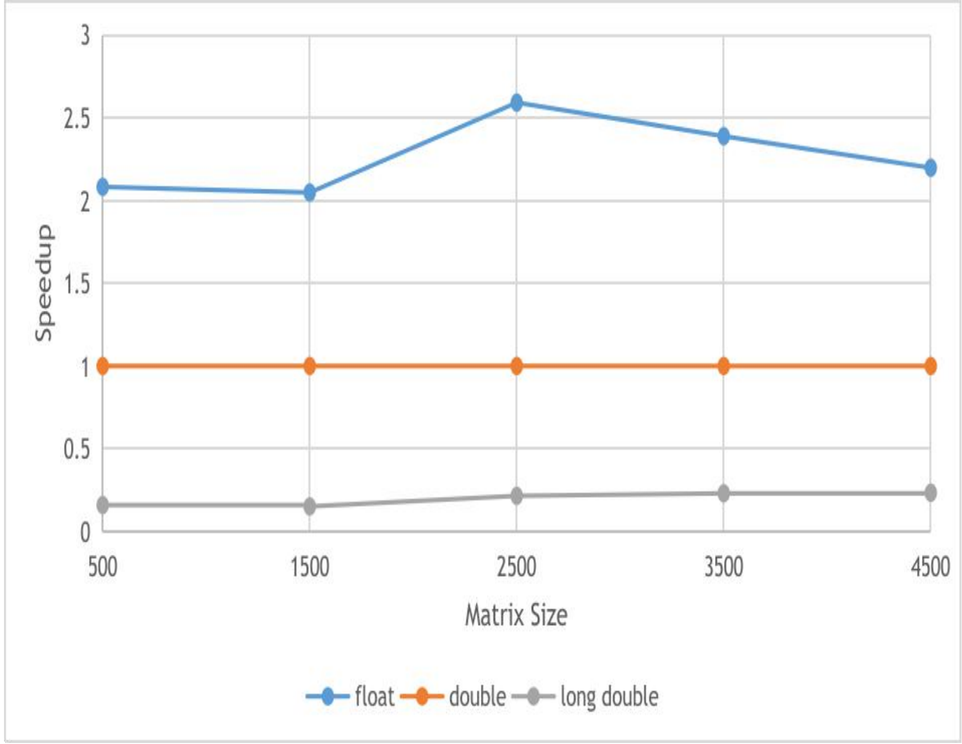
- -fno-alias: assume there is not memory aliasing and optimize further

- -xCORE-AVX512: utilizes broader vector registers and supplementary SIMD operations provided by the AVX-512 instruction set supported on SuperMUC-NG



single precision and optimized kernel

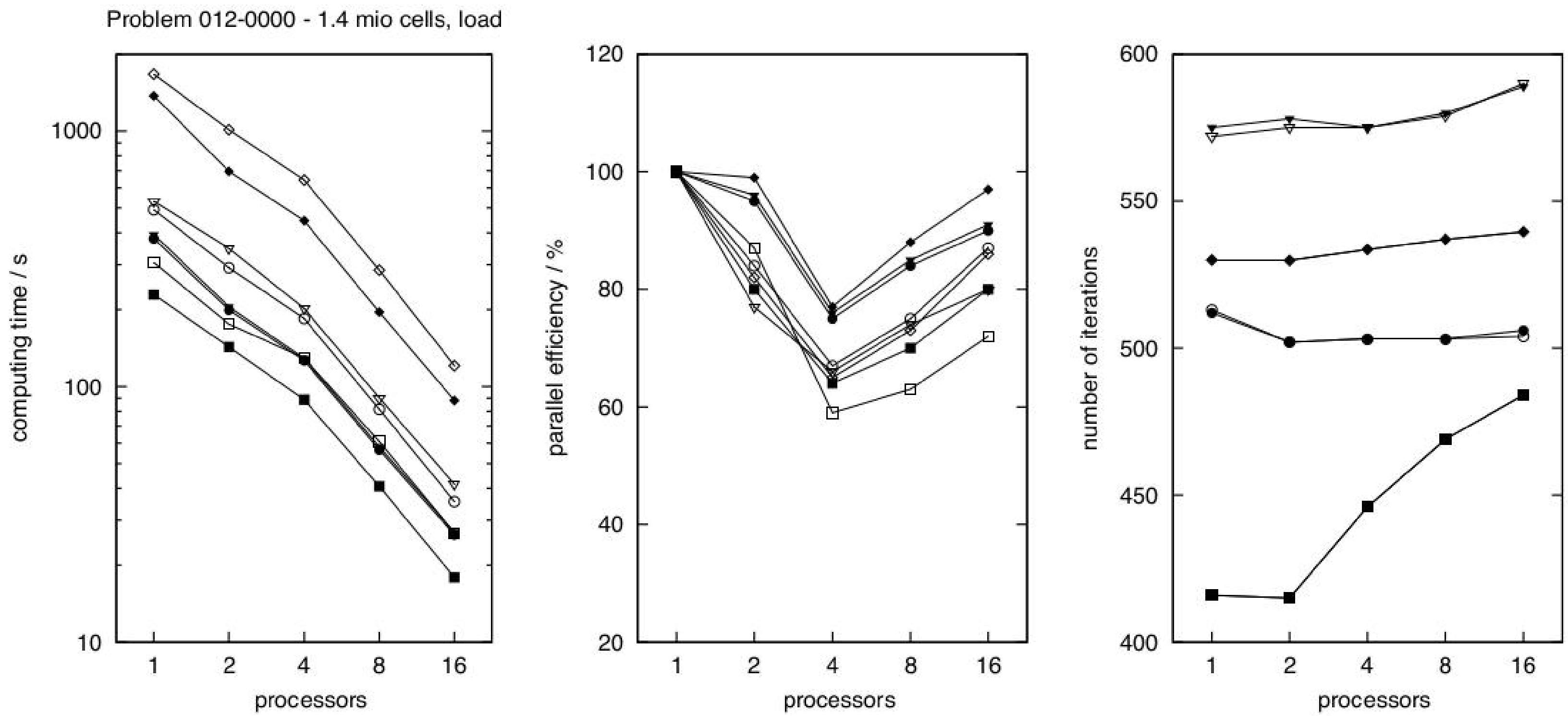
Matrix-Matrix Multiplication - Precision Optimization



optimized kernel and -O2

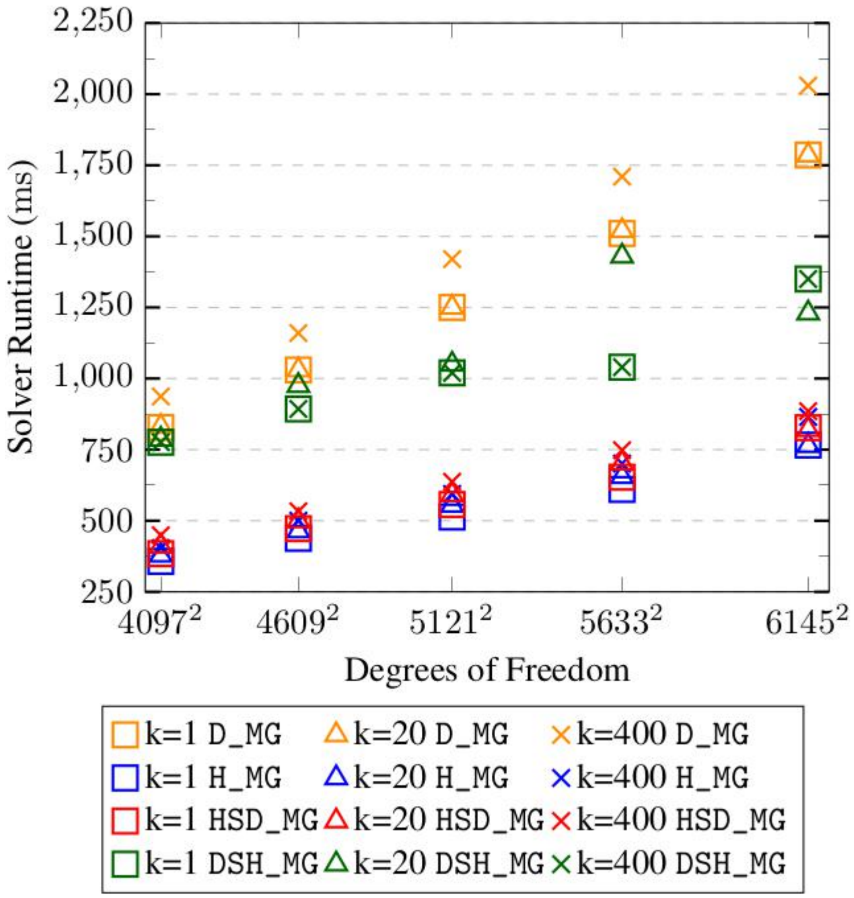
Performance Evaluation - CFD Simulation on CPU

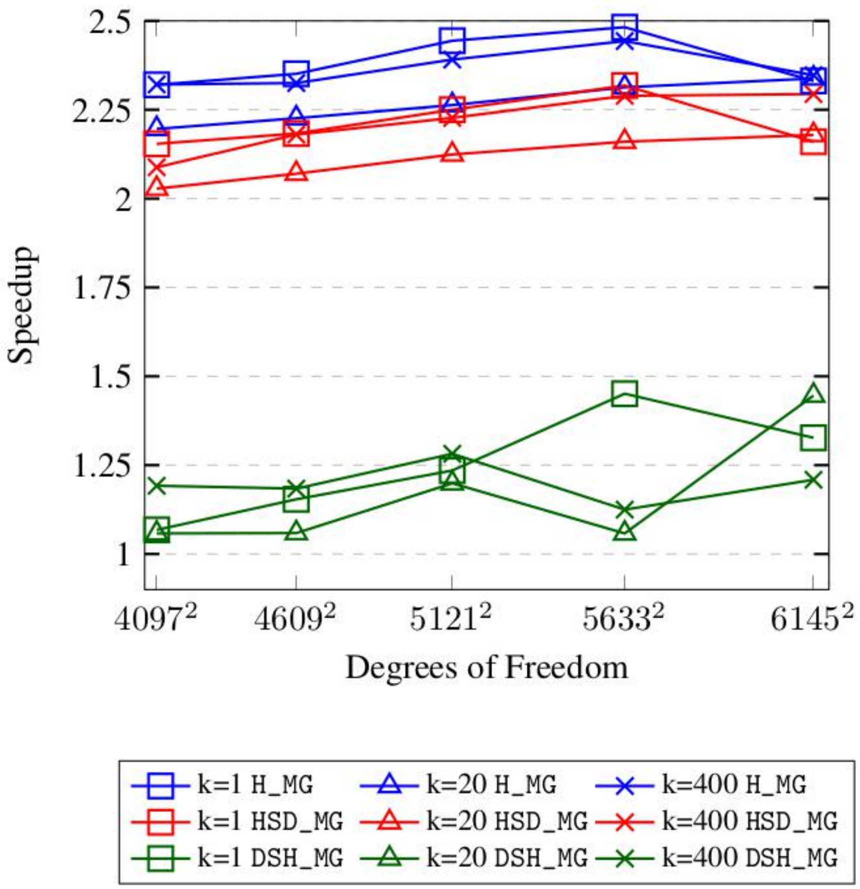
Double precision conjugate gradient algorithm preconditioned by **single** precision multigrid methods



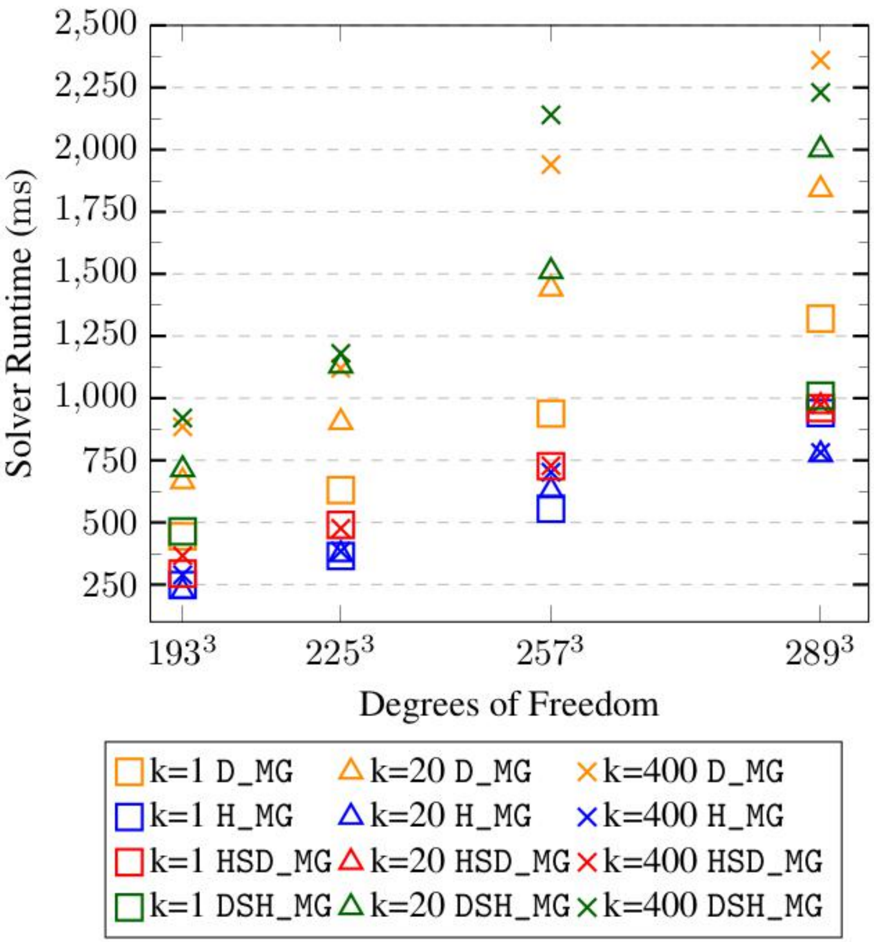
empty symbols: 8-bit; filled symbols: 4-bit; different markers mean different multigrid methods

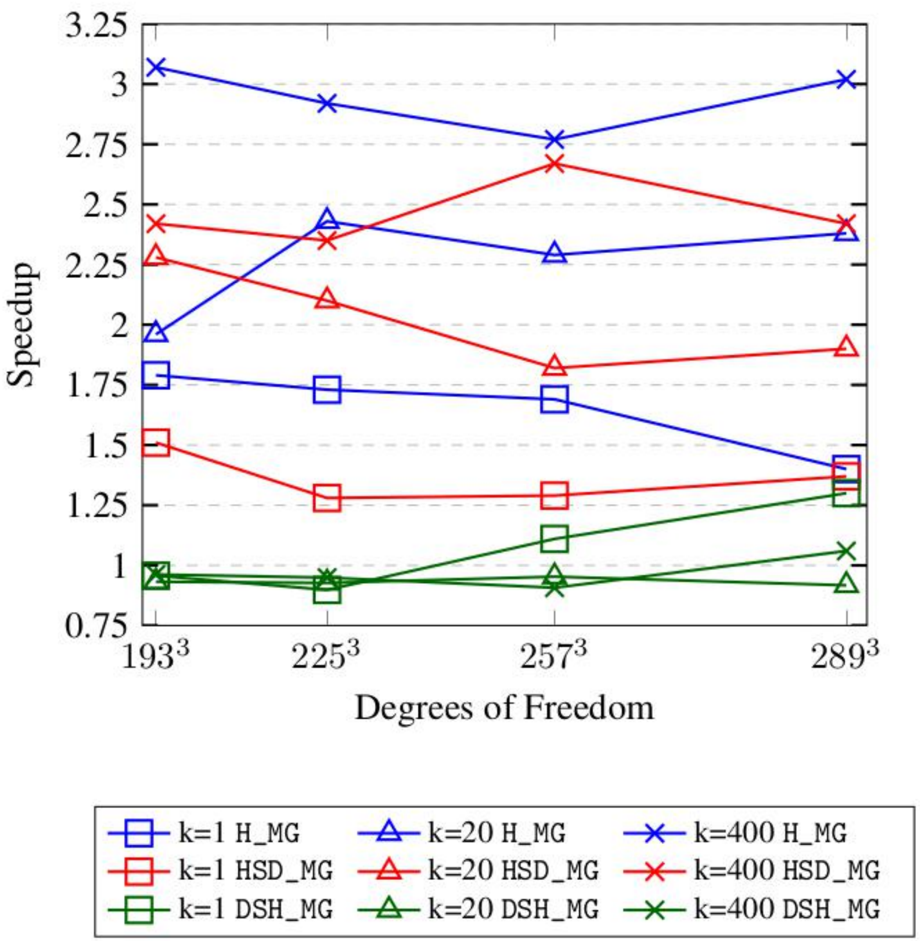
Performance Evaluation - 2D Poisson's Equation



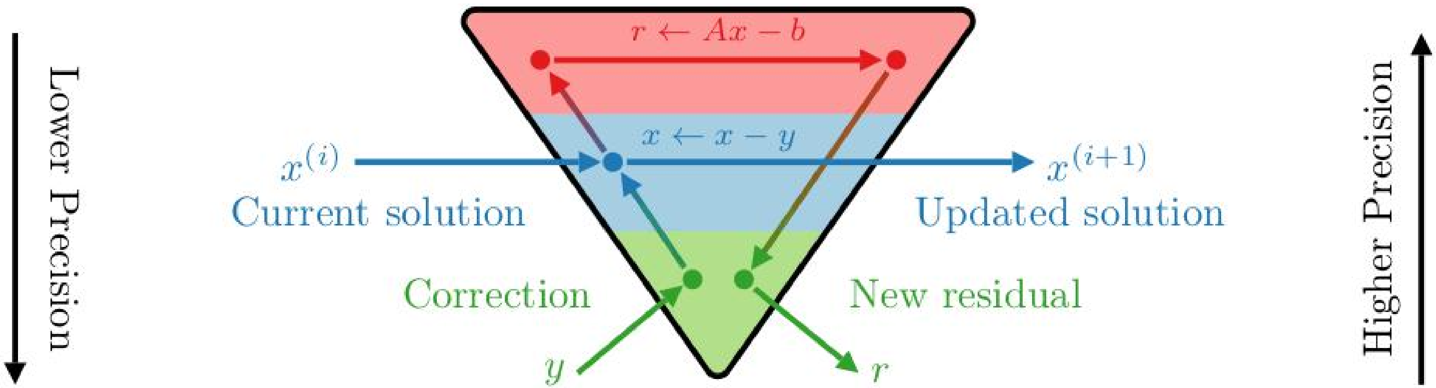


Performance Evaluation - 3D Poisson's Equation





Rounding Error Analysis - Iterative Refinement



Algorithm 3.1 `Iterative Refinement` ($\mathcal{IR}$)

Input: A, b, x initial guess, tol > 0 convergence tolerance.

1: $r \leftarrow Ax - b$ ▷ Compute $\mathcal{IR}$ Residual and Round
2: **if** $\|r\| <$ tol **then**
3:     **return** x ▷ Return Solution of $Ax = b$
4: **end if**
5: $y \leftarrow \mathtt{InnerSolve}(A, r)$ ▷ Compute Approximate Solution of $Ay = r$
6: $x \leftarrow x - y$ ▷ Update Approximate Solution of $Ax = b$
7: **goto** 1

Rounding Error Analysis - Iterative Refinement

Discrete energy norm:

$$\|x\|_A = \|A^{\frac{1}{2}}x\|, x \in \mathbb{R}^n$$

Condition numbers:

$$\kappa(A) = \|A\| \cdot \|A^{-1}\|$$

$$\underline{\kappa}(A) = \psi\|A^{-1}\|$$

$$\psi(A) = \||A|\|$$

Sparsity factor:

$$\bar{m}_A^+ = \frac{m_A + 1}{1 - (m_A + 1)\bar{\varepsilon}}$$

Bound-related parameters:

$$\dot{\tau} = \kappa^{\frac{1}{2}}\dot{\varepsilon}$$

$$\tau = \kappa^{\frac{1}{2}}\varepsilon$$

$$\bar{\tau} = \kappa\bar{\varepsilon}$$

$$\gamma = \frac{\kappa^{\frac{1}{2}} + \underline{\kappa}}{\kappa}$$

$$\bar{\varepsilon} \leq \varepsilon \leq \dot{\varepsilon}$$

Rounding Error Analysis - Iterative Refinement

Error bound of inner solver:

$$\|y - A^{-1}r\|_A \leq \rho\|A^{-1}r\|_A$$

Relative error bound:

$$\frac{\|x^{(i+1)} - A^{-1}b\|_A}{\|A^{-1}b\|_A} \leq \rho_{ir}\frac{\|x^{(i)} - A^{-1}b\|_A}{\|A^{-1}b\|_A} + \chi$$

Convergence factor:

$$\rho_{ir} = \rho + \delta_{\rho_{ir}}, \quad \delta_{\rho_{ir}} = \frac{(1+2\rho)\tau + \gamma(1+\rho)(1+\varepsilon)\bar{m}_A^+\bar{\tau}}{1-\tau}$$

Limiting accuracy:

$$\chi = \frac{\tau + \gamma(1+\rho)(1+\varepsilon)\bar{m}_A^+\bar{\tau}}{1-\tau}$$

Rounding Error Analysis - Iterative Refinement

Proof:

$$r = \underbrace{Ax^{(i)} - b}_{\text{exact residual}} + \underbrace{\delta_1}_{\varepsilon\text{-}\bar{\varepsilon}\text{ error}}, \quad |\delta_1| \leq \varepsilon|Ax^{(i)} - b| + (1+\varepsilon)\bar{m}_A^+\bar{\varepsilon}\left(|b| + |A|\cdot|x^{(i)}|\right)$$

$$\begin{aligned}
\left\|A^{-\frac{1}{2}}\left(|b| + |A|\cdot|x^{(i)}|\right)\right\| &\leq \|A^{-\frac{1}{2}}\|\left(\|AA^{-1}b\| + \psi\|x^{(i)}\|\right) \\
&\leq \|A^{-\frac{1}{2}}\|\cdot\|A^{\frac{1}{2}}\|\cdot\|A^{-1}b\|_A + \psi\|A^{-1}\|\cdot\|x^{(i)}\|_A \\
&= \kappa^{\frac{1}{2}}\|A^{-1}b\|_A + \underline{\kappa}\|x^{(i)}\|_A \\
&\leq (\kappa^{\frac{1}{2}} + \underline{\kappa})\|A^{-1}b\|_A + \underline{\kappa}\|x^{(i)} - A^{-1}b\|_A.
\end{aligned}$$

$$\begin{aligned}
\|A^{-1}\delta_1\|_A &= \|A^{-\frac{1}{2}}\delta_1\| \\
&\leq \varepsilon\|A^{-\frac{1}{2}}\|\cdot\|Ax^{(i)} - b\| + (1+\varepsilon)\bar{m}_A^+\bar{\varepsilon}\left\|A^{-\frac{1}{2}}\left(|b| + |A|\cdot|x^{(i)}|\right)\right\| \\
&\leq \varepsilon\kappa^{\frac{1}{2}}\|x^{(i)} - A^{-1}b\|_A \\
&\quad + (1+\varepsilon)\bar{m}_A^+\bar{\varepsilon}\left((\kappa^{\frac{1}{2}} + \underline{\kappa})\|A^{-1}b\|_A + \underline{\kappa}\|x^{(i)} - A^{-1}b\|_A\right).
\end{aligned}$$

Rounding Error Analysis - Iterative Refinement

$$
\begin{aligned}
&\|y - A^{-1}r\|_A \\
&\quad \le \rho\|A^{-1}r\|_A \\
&\quad = \rho\|A^{-1}\left(Ax^{(i)} - b + \delta_1\right)\|_A \\
&\quad \le \rho\left[\|x^{(i)} - A^{-1}b\|_A + \|A^{-1}\delta_1\|_A\right] \\
&\quad \le \rho\Big[(1+\varepsilon\kappa^{\frac{1}{2}})\|x^{(i)} - A^{-1}b\|_A \\
&\qquad + (1+\varepsilon)\bar{m}_A^{+}\bar{\varepsilon}\left((\kappa^{\frac{1}{2}} + \underline{\kappa})\|A^{-1}b\|_A + \underline{\kappa}\|x^{(i)} - A^{-1}b\|_A\right)\Big]
\end{aligned}
$$

$$
x^{(i+1)} = \underbrace{x^{(i)} - y}_{\text{exact update}} + \underbrace{\delta_2}_{\varepsilon \text{ error}}, \quad |\delta_2| \le \varepsilon|x^{(i+1)}|
$$

$$
\begin{aligned}
\|x^{(i+1)} - A^{-1}b\|_A &= \|x^{(i)} - A^{-1}b - A^{-1}r - (y - A^{-1}r) + \delta_2\|_A \\
&\le \|A^{-1}\delta_1 + (y - A^{-1}r)\|_A + \|\delta_2\|_A \\
&\le \|A^{-1}\delta_1\|_A + \|y - A^{-1}r\|_A + \varepsilon\|A^{\frac{1}{2}}\| \cdot \|x^{(i+1)}\| \\
&\le \left(\rho + (1+\rho)\varepsilon\kappa^{\frac{1}{2}})\right)\|x^{(i)} - A^{-1}b\|_A \\
&\quad + (1+\rho)(1+\varepsilon)\bar{m}_A^{+}\bar{\varepsilon}\left((\kappa^{\frac{1}{2}} + \underline{\kappa})\|A^{-1}b\|_A + \underline{\kappa}\|x^{(i)} - A^{-1}b\|_A\right) \\
&\quad + \varepsilon\kappa^{\frac{1}{2}}(\|x^{(i+1)} - A^{-1}b\|_A + \|A^{-1}b\|_A) \\
&= \varepsilon\kappa^{\frac{1}{2}}\|x^{(i+1)} - A^{-1}b\|_A + (\rho + \delta_3)\|x^{(i)} - A^{-1}b\|_A \\
&\quad + \left(\varepsilon\kappa^{\frac{1}{2}} + (1+\rho)(1+\varepsilon)\bar{m}_A^{+}\bar{\varepsilon}(\kappa^{\frac{1}{2}} + \underline{\kappa})\right)\|A^{-1}b\|_A,
\end{aligned}
$$

$$
\delta_3 = (1+\rho)\left(\varepsilon\kappa^{\frac{1}{2}} + (1+\varepsilon)\bar{m}_A^{+}\bar{\varepsilon}\underline{\kappa}\right)
$$

$$
\frac{\rho + \delta_3}{1 - \varepsilon\kappa^{\frac{1}{2}}} = \rho + \frac{\delta_3 + \rho\varepsilon\kappa^{\frac{1}{2}}}{1 - \varepsilon\kappa^{\frac{1}{2}}} = \rho + \delta_{\rho_{ir}}
$$

$$
\sum_{i=0}^{N-1}(\rho + \delta_{\rho_{ir}})^i \le \frac{1}{1 - (\rho + \delta_{\rho_{ir}})}
$$

$\square$

Rounding Error Analysis - Two Grid

Algorithm 5.1 `Two-Grid (`$\mathcal{TG}$`) Correction Scheme`

Input: A, r, P, M.

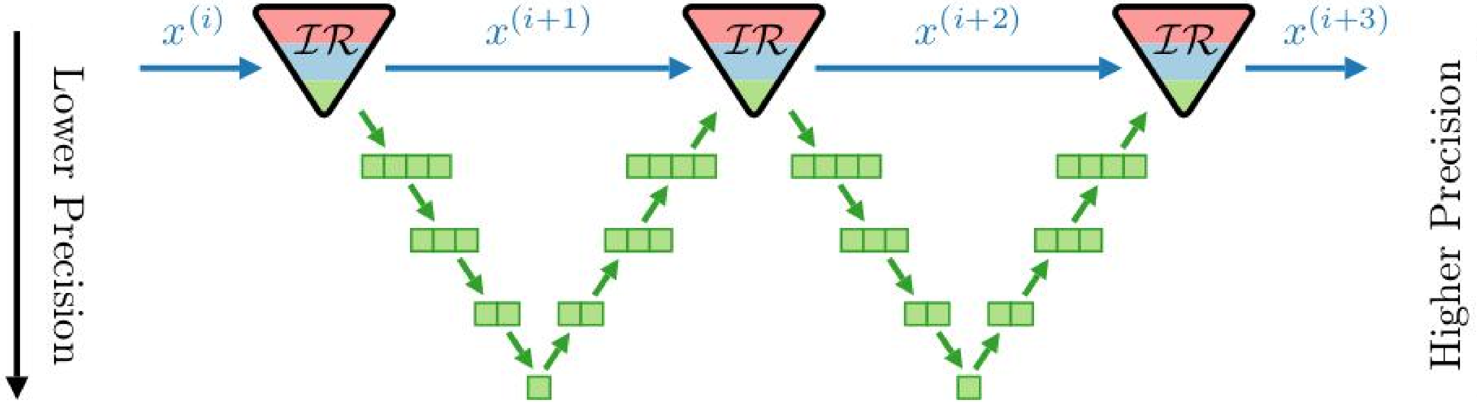
$r \leftarrow r$ ▷ Round RHS and Initialize $\mathcal{TG}$
$y \leftarrow Mr$ ▷ Relax on Current Approximation ($y = 0$)
$r_{\mathrm{tg}} \leftarrow Ay - r$ ▷ Evaluate $\mathcal{TG}$ Residual
$b_c \leftarrow P^t r_{\mathrm{tg}}$ ▷ Restrict $\mathcal{TG}$ Residual to Coarse-Level
$d_c \leftarrow B_c(P^t AP)^{-1} b_c$ ▷ Solve Coarse-Level Equation
$d \leftarrow Pd_c$ ▷ Interpolate Correction to Fine Level
$y \leftarrow y - d$ ▷ Update Approximate Solution of $Ay = r$
return y ▷ Return Approximate Solution of $Ay = r$

Convergence established:

$$\|y - A^{-1}r\|_A \leq \rho_{tg}\|A^{-1}r\|_A, \quad \rho_{tg} = \rho_{tg}^* + \delta_{\rho_{tg}}$$

$$\delta_{\rho_{tg}} = \delta_{\rho_{tg}}(\dot{\tau}) = a_1\dot{\tau} + a_2\dot{\tau}^2 + a_3\dot{\tau}^3$$

Rounding Error Analysis - V(1,0)-Cycle

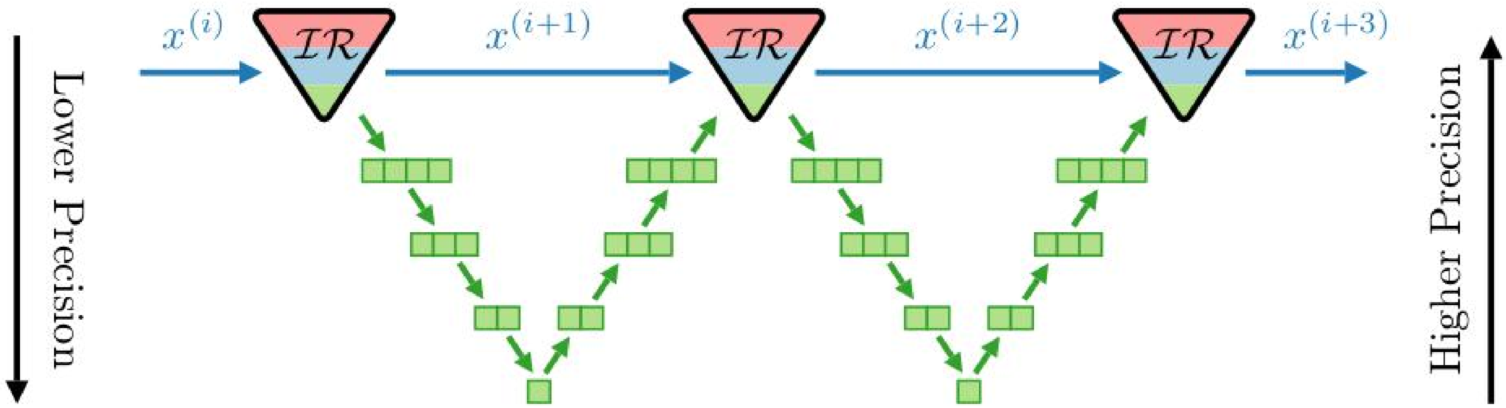


Algorithm 6.1 `V(1,0)-Cycle (`$\mathcal{V}$`) Correction Scheme`

Input: A, r, P, $\ell \geq 1$ $\mathcal{V}$ levels.

$r \leftarrow r$ ▷ Round RHS and Initialize $\mathcal{V}$
$y \leftarrow Mr$ ▷ Relax on Current Approximation ($y = 0$)
if $\ell > 1$ **then** ▷ Check for Coarser Level
  $r_{\mathrm{V}} \leftarrow Ay - r$ ▷ Evaluate $\mathcal{V}$ Residual
  $r_{\ell-1} \leftarrow P^t r_{\mathrm{V}}$ ▷ Restrict $\mathcal{V}$ Residual to Coarse-Level
  $d_{\ell-1} \leftarrow \mathcal{V}(A_{\ell-1}, r_{\ell-1}, P_{\ell-1}, \ell - 1)$ ▷ Compute Correction from Coarser Levels
  $d \leftarrow P d_{\ell-1}$ ▷ Interpolate Correction to Fine Level
  $y \leftarrow y - d$ ▷ Update Approximate Solution of $Ay = r$
end if
return y ▷ Return Approximate Solution of $Ay = r$

Rounding Error Analysis - V(1,0)-Cycle



Precision coarsening factor:

$$\dot{\zeta}_j = \frac{\dot{\varepsilon}_{j-1}}{\dot{\varepsilon}_j}, 2 \leq j \leq \ell$$

Pseudo mesh-refinement factor:

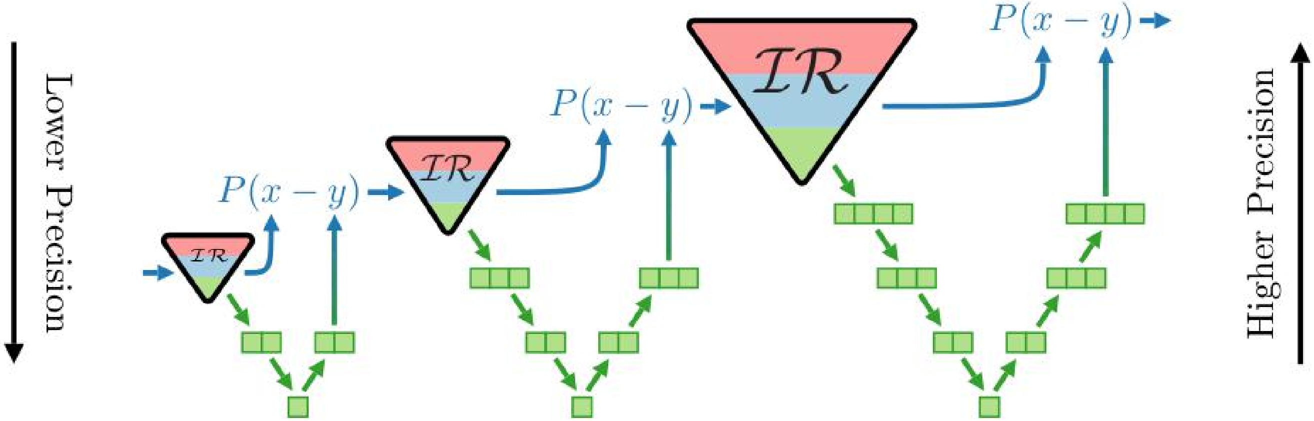
$$\theta_j = \frac{h_{j-1}}{h_j}, \quad 2 \leq j \leq \ell$$

Convergence established:

$$\|y - A^{-1}r\|_A \leq \rho_v \|A^{-1}r\|_A, \; \rho_v = \rho_v^* + \delta_{\rho_v}, \; \delta_{\rho_v} = \delta_{\rho_v}(\dot{\tau}_\ell) = \frac{\vartheta^m}{\vartheta^m - 1}\delta_{\rho_{tg}}(\dot{\tau}_\ell) \;,$$

where $\vartheta = \min_{1 \leq j \leq \ell} \{\theta_j \dot{\zeta}_j^{-\frac{1}{m}}\}$

Rounding Error Analysis - Full Multigrid

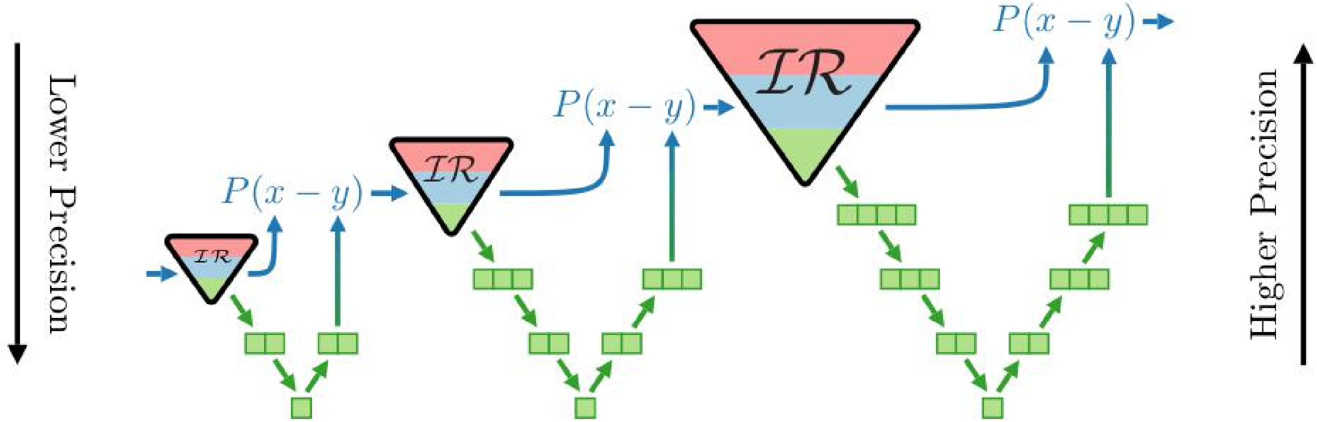


Algorithm 9.1 $\texttt{FMG}(1,0)\texttt{-Cycle}$ $(\mathcal{FMG})$

Input: A, b, P, $N \geq 1$ $\mathcal{IR}$ cycles (using one V$(1,0)$ each), $\ell \geq 1$ $\mathcal{FMG}$ levels.

$x \leftarrow 0$ ▷ Initialize $\mathcal{FMG}$
if $\ell > 1$ **then** ▷ Check for Coarser Level
  $x_{\ell-1} \leftarrow \mathcal{FMG}(A_{\ell-1}, b_{\ell-1}, P_{\ell-1}, \ell-1, N)$ ▷ Compute Coarse-Level Approximation
  $x \leftarrow P x_{\ell-1}$ ▷ Interpolate Approximation to Fine Level
end if
$i \leftarrow 0$ ▷ Initialize $\mathcal{IR}$
while $i < N$ **do**
  $r \leftarrow Ax - b$ ▷ Update $\mathcal{IR}$ Residual and Round
  $y \leftarrow \mathcal{V}(A, r, P, \ell)$ ▷ Compute Correction by $\mathcal{V}$
  $i \leftarrow i + 1$ ▷ Increment $\mathcal{IR}$ Cycle Counter
  $x \leftarrow x - y$ ▷ Update Approximate Solution of $Ax = b$
end while
return x ▷ Return Approximate Solution of $Ax = b$

Rounding Error Analysis - Full Multigrid



Assume:

$\rho_v + \delta_{\rho_{ir}} < 1$

χ is small enough

N is large enough

Then:

$$(\rho_v + \delta_{\rho_{ir}})^N \left((\sqrt{2} + \mu\tau)\theta^q C h^q + \mu\tau \right) + \frac{\chi}{1 - (\rho_v + \delta_{\rho_{ir}})} \leq C h^q$$ holds on all levels j.

Here $\mu = \mu_j = \kappa^{\frac{1}{2}}(P_j^t P_j) m_P^+$, h is pseudo mesh size, C and q are positive constants.

References

Main references for the presentation:

- A. Abdelfattah, H. Anzt, E. G. Boman, E. Carson, T. Cojean, J. Dongarra, A. Fox, M. Gates, N. J. Higham, X. S. Li, J. Loe, P. Luszczek, S. Pranesh, S. Rajamanickam, T. Ribizel, B. F. Smith, K. Swirydowicz, S. Thomas, S. Tomov, Y. M. Tsai, and U. M. Yang, "A survey of numerical linear algebra methods utilizing mixed-precision arithmetic," The International Journal of High Performance Computing Applications, vol. 35, pp. 344–369, July 2021.
- M. Kronbichler and K. Ljungkvist, "Multigrid for Matrix-Free High-Order Finite Element Computations on Graphics Processors," ACM Transactions on Parallel Computing, vol. 6, pp. 2:1–2:32, May 2019.
- S. F. McCormick, J. Benzaken, and R. Tamstorf, "Algebraic Error Analysis for Mixe Precision Multigrid Solvers," SIAM Journal on Scientific Computing, vol. 43, pp. S392–S419, Jan. 2021.

The full list is presented in my paper.

THANKS