

Mehrgitter- Verfahren in gemischter Genauigkeit auf GPUs

Daniel
Fink

Vortrag

-

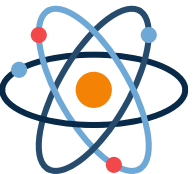


Bachelorarbeit

31. Oktober 2019

- Motivation
- Zielsetzung
- Mehrgitterverfahren
- Gemischt genaue Mehrgitterverfahren
 - Theoretischer Speed-Up
 - Fehlerentwicklung
 - Experimenteller Speed-Up
- Fazit und Ausblick

Motivation

- Was bedeutet gemischte Genauigkeit?
 - Verwendung unterschiedlicher Datentypen für geringere(n)
 - Welche Datentypen gibt es?
- Ausführungszeit**
Speicherbedarf
Kommunikationsaufwand

| Double | Single | Half |
|---|---|--|
|  64 Bit lang $\epsilon \approx 10^{-16}$ |  32 Bit lang $\epsilon \approx 10^{-8}$ |  16 Bit lang $\epsilon \approx 10^{-4}$ GPU |

- NVIDIA Tesla V100:

Single = $\frac{1}{2}$ Double
Half = $\frac{1}{4}$ Double } Sowohl Rechen- als auch Ladezeit → Gesamt: Faktor 2 bzw. 4

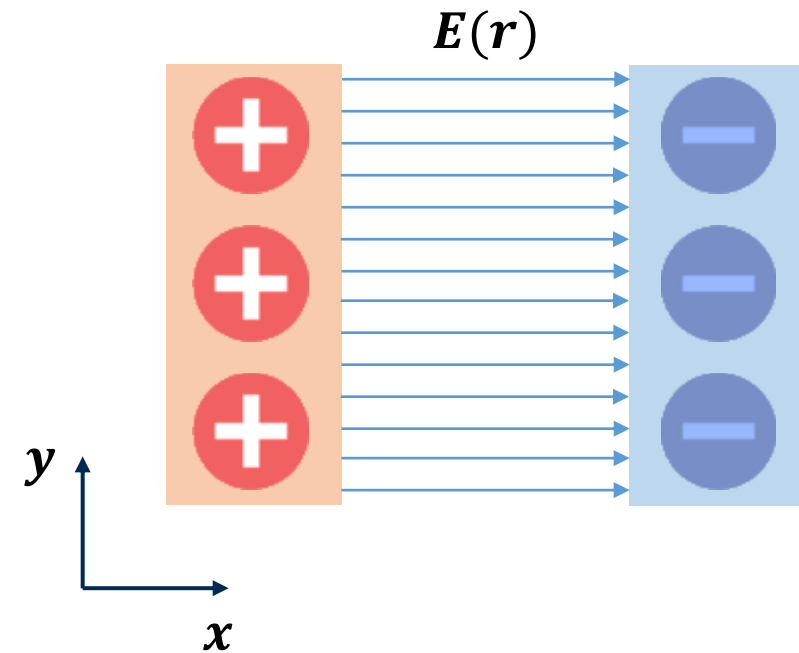
- Anwendungsbereich: Simulationen (Lösen von PDEs, ODEs, ...)

- Hier: Poisson-Gleichung

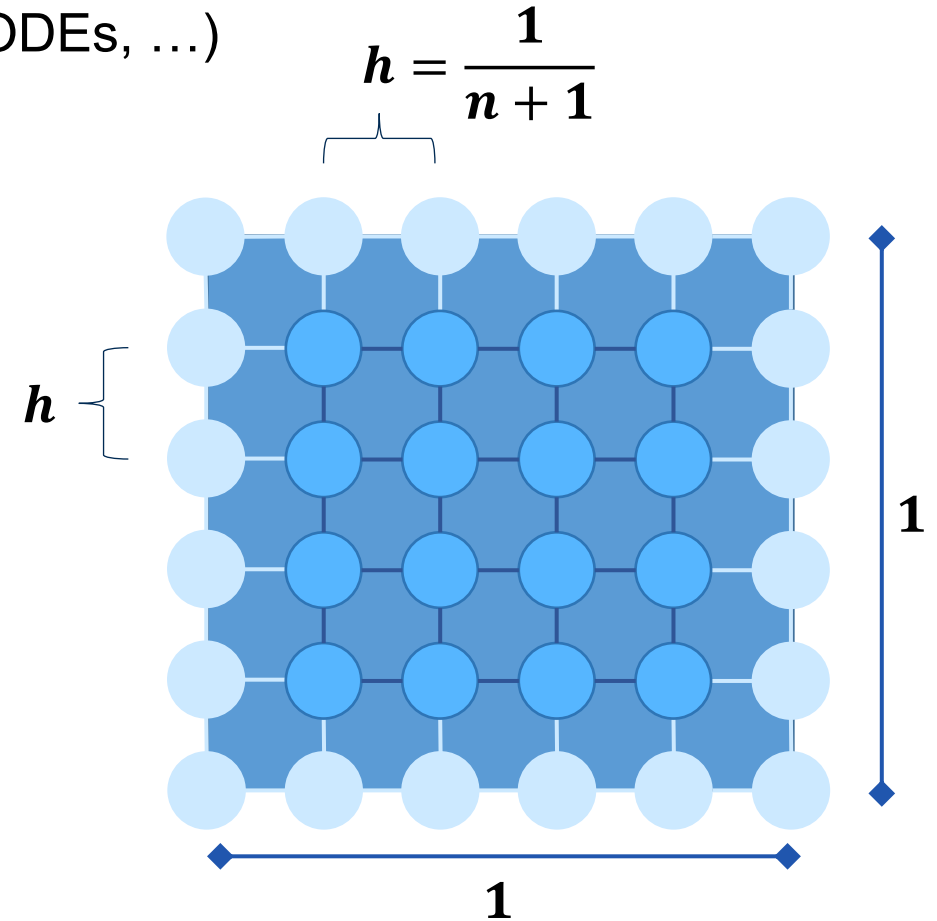
$$-\Delta u = f \quad \text{in} \quad \Omega = (0,1) \times (0,1)$$

$$u = g \quad \text{auf} \quad \partial\Omega$$

wobei $u \in C^2$ und $f, g \in C$



- Anwendungsbereich: Simulationen (Lösen von PDEs, ODEs, ...)
- Hier: Poisson-Gleichung
$$-\Delta u = f \quad \text{in } \Omega = (0,1) \times (0,1)$$
$$u = g \quad \text{auf } \partial\Omega$$
wobei $u \in C^2$ und $f, g \in C$
- Vorgehen:
 - Diskretisieren (Hier: Finite Differenzen)



- Anwendungsbereich: Simulationen (Lösen von PDEs, ODEs, ...)

- Hier: Poisson-Gleichung

$$-\Delta u = f \quad \text{in} \quad \Omega = (0,1) \times (0,1)$$

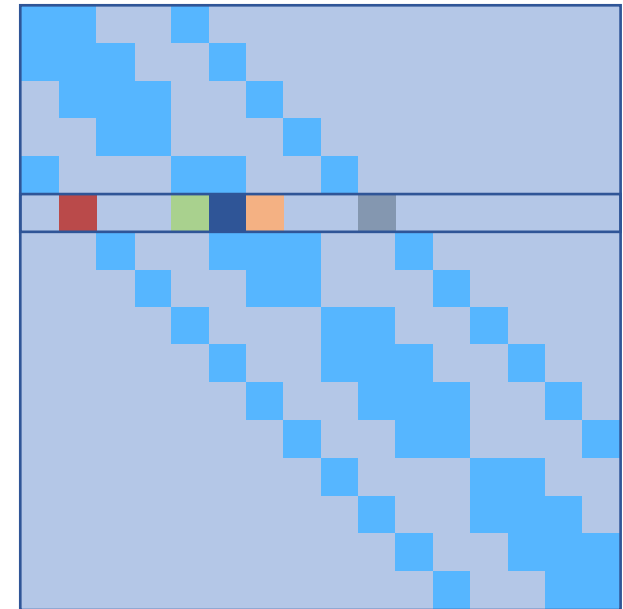
$$u = g \quad \text{auf} \quad \partial\Omega$$

wobei $u \in C^2$ und $f, g \in C$

- Vorgehen:

- Diskretisieren (Hier: Finite Differenzen)

→ (dünnbesetztes) Gleichungssystem $Au = f$



Systemmatrix $A \in \mathbb{R}^{n^2 \times n^2}$

- Anwendungsbereich: Simulationen (Lösen von PDEs, ODEs, ...)

- Hier: Poisson-Gleichung

$$-\Delta u = f \quad \text{in} \quad \Omega = (0,1) \times (0,1)$$

$$u = g \quad \text{auf} \quad \partial\Omega$$

wobei $u \in C^2$ und $f, g \in C$

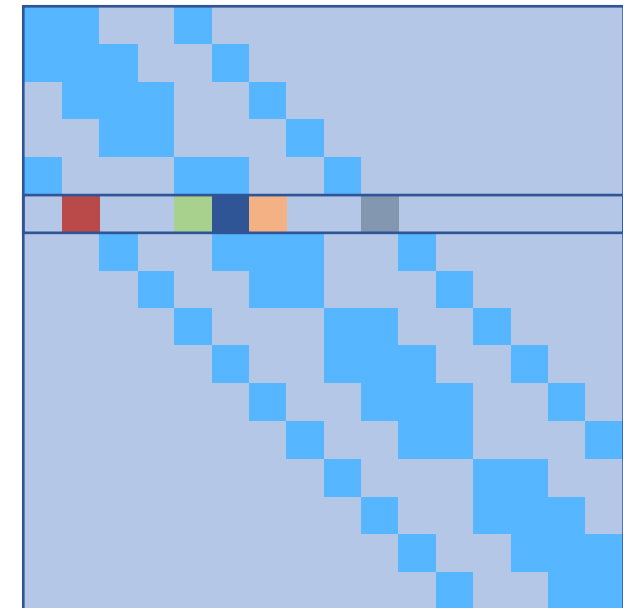
- Vorgehen:

- Diskretisieren (Hier: Finite Differenzen)

→ (dünnbesetztes) Gleichungssystem $Au = f$

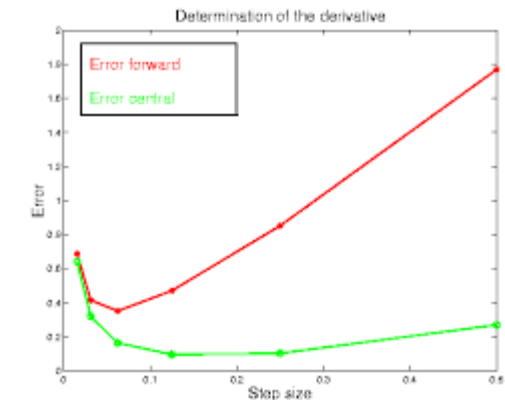
- Lösen mittels Mehrgitterverfahren

- Hierarchischer Aufbau → gemischte Genauigkeit



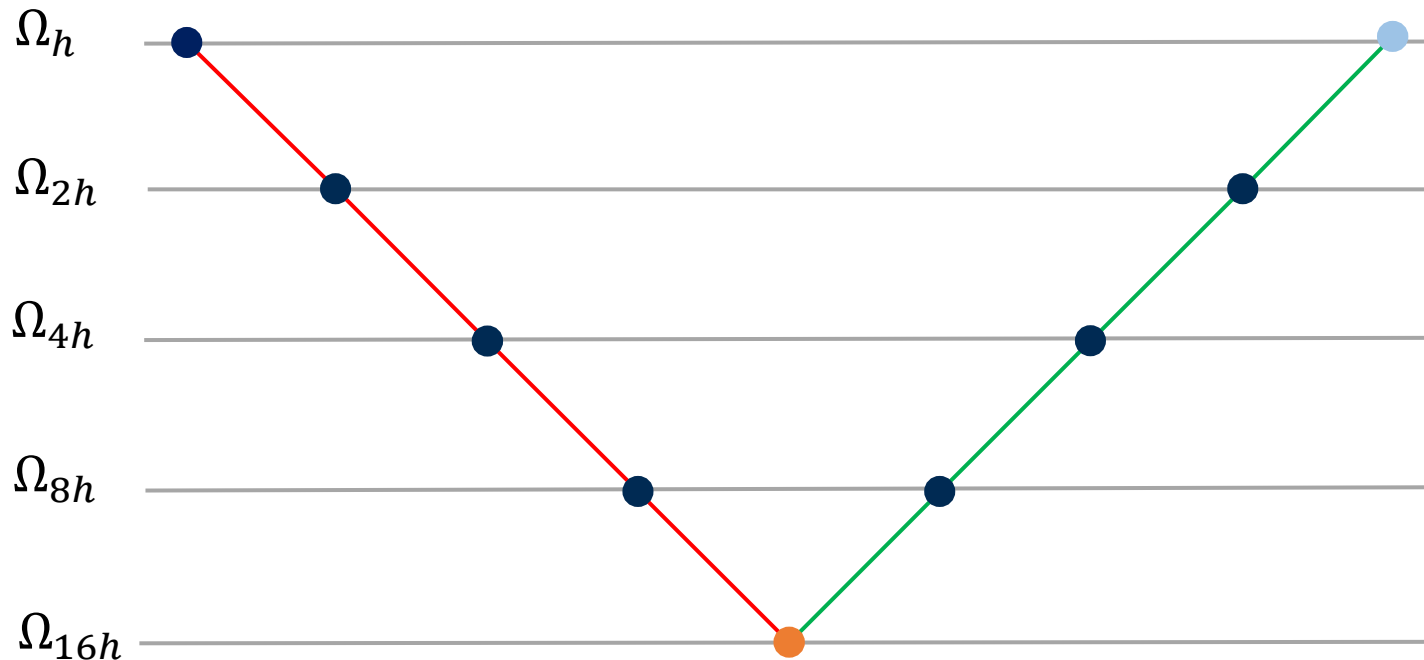
Systemmatrix $A \in \mathbb{R}^{n^2 \times n^2}$

- Mehrgitterverfahren mit **Double**, **Single** und **Half**
 - Problem wird in doppelter Genauigkeit gestellt
 - Lösung soll am Ende in doppelter Genauigkeit vorliegen
 - Vergleich mit MG-Verfahren in Double
- Fehlerentwicklung und Ausführungszeit



Mehrgitterverfahren

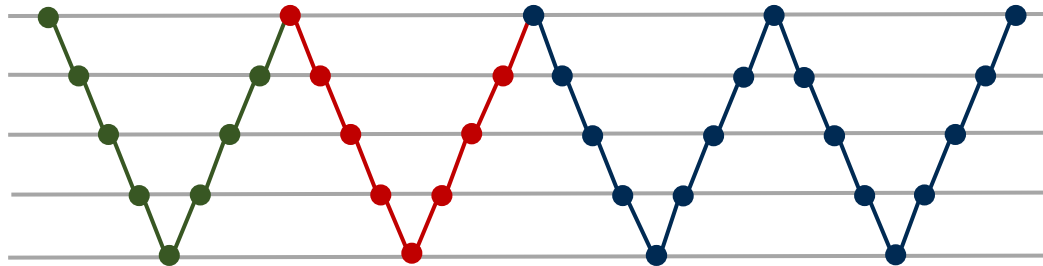
V-Zyklus



| | |
|---------------------|---------------------|
| Relaxiere $Au = f$ | Jacobi-Verfahren |
| Restringiere r | halb-gewichtet |
| Relaxiere $Ae = r$ | Jacobi-Verfahren |
| ⋮ | |
| Löse $Ae = r$ | Jacobi-Verfahren |
| Prolongiere e | Bilineare Interpol. |
| Relaxiere $Ae = r$ | Jacobi-Verfahren |
| ⋮ | |
| Approximation u^* | |

Gemischte genaue MG-Verfahren

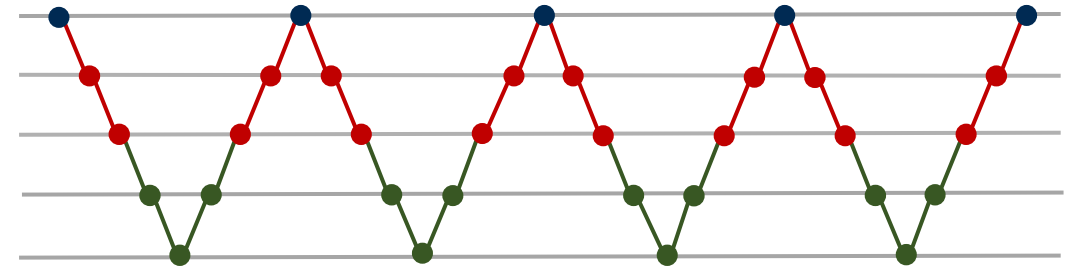
Horizontal-Verfahren



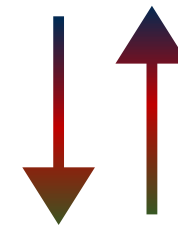
Half Single Double ...



Vertikal-Verfahren



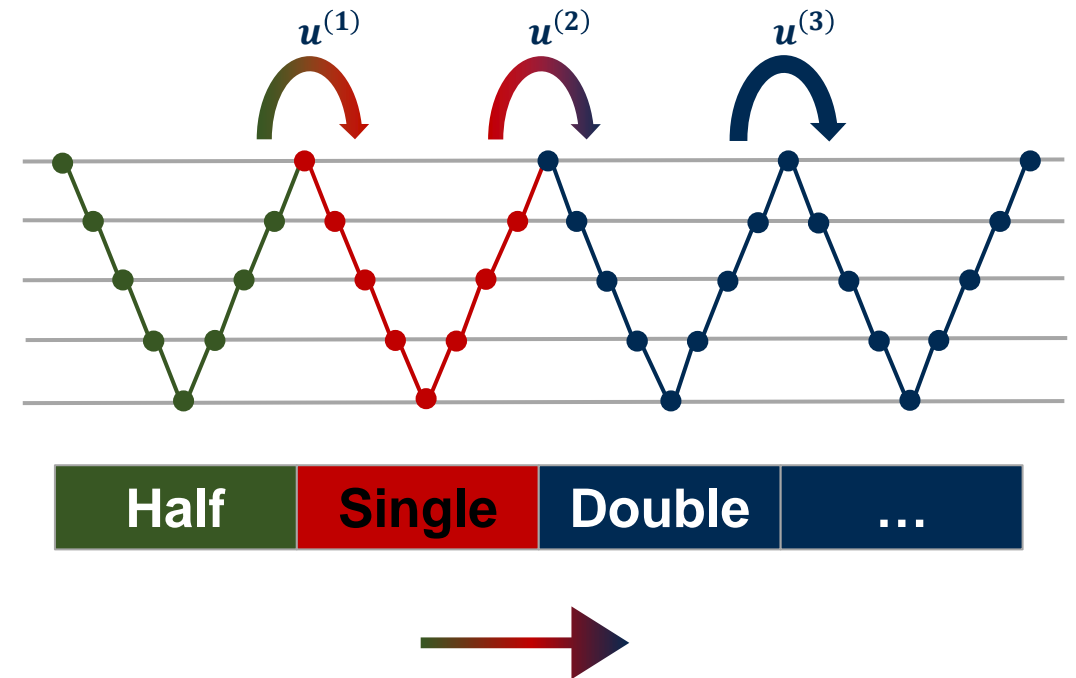
Double
Single
Half



Double
Single
Half

Horizontal-Verfahren

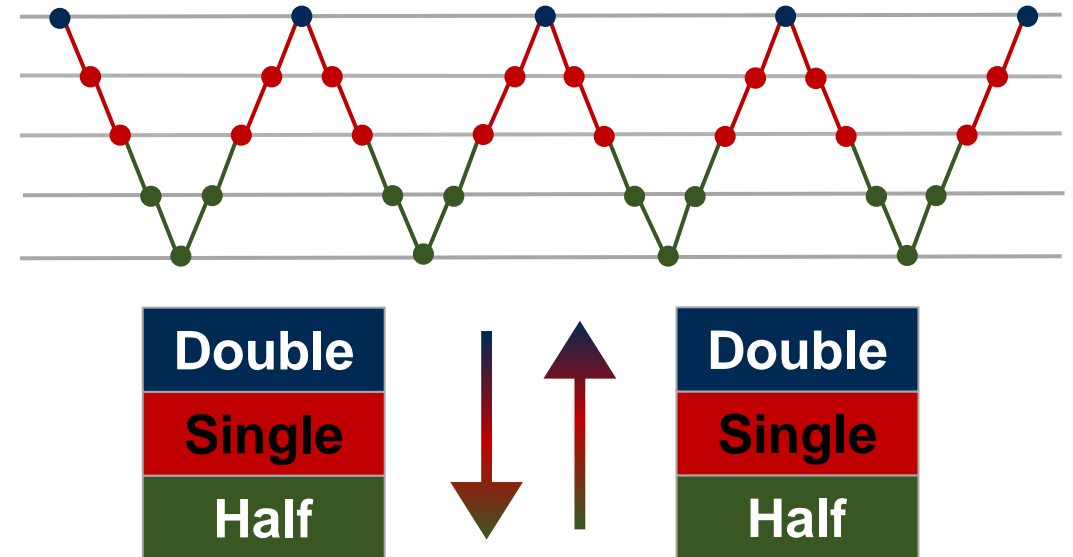
- V-Zyklus → Verbesserte Startlösung
- Ohnehin mehrere Iterationen notwendig
- **Half/Single**-MG als Vorkonditionierer
- Wann ändert man den Datentyp?
- Konvergenzrate oder feste Vorgabe



Zwei Verfahren { **Single/Double** → H2SMG-Verfahren
Half/Single/Double → H3SMG-Verfahren

Vertikal-Verfahren

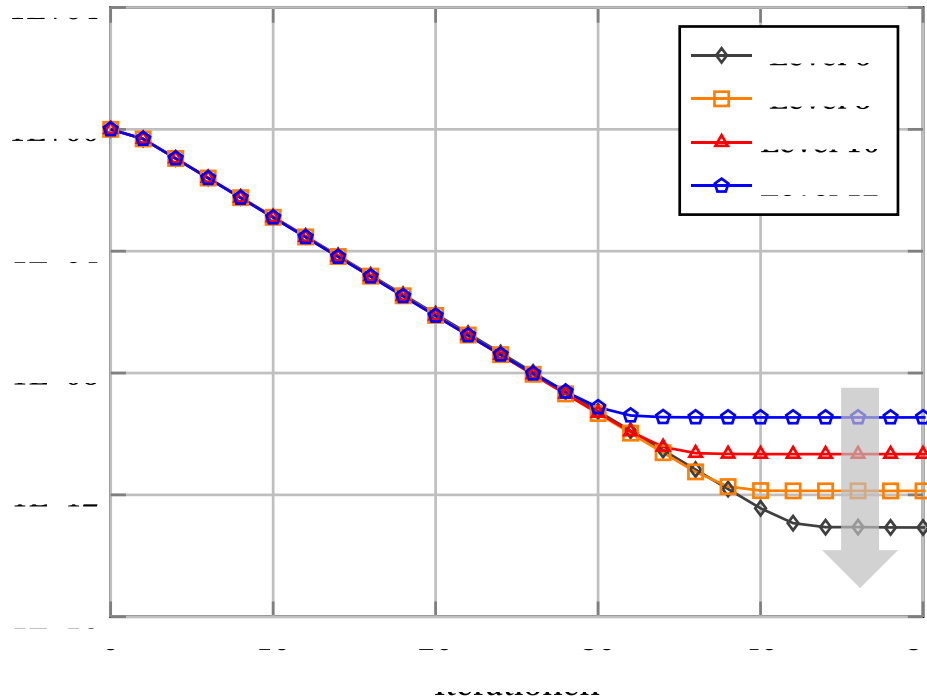
- Größerer Diskretisierungsfehler auf groben Gittern
→ Double viel zu genau
- Wann ändert man den Datentyp?
→ Feste Vorgabe der einzelnen Level



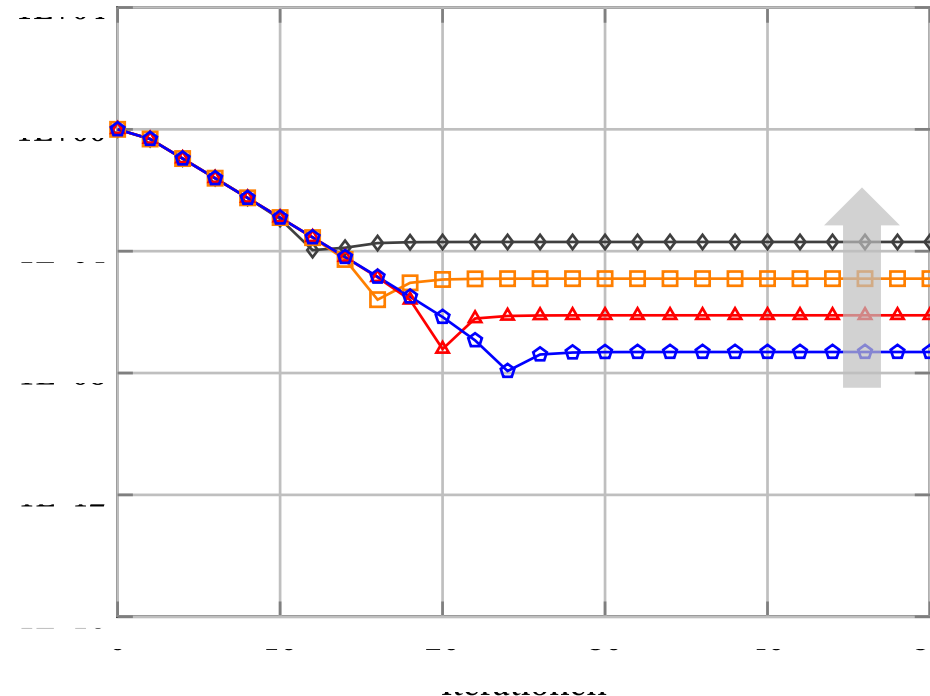
Zwei Verfahren {
 Single/Double → V2SMG-Verfahren
 Half/Single/Double → V3SMG-Verfahren

MG-Verfahren in Double

Relatives Residuum $\frac{\|f - Au^{(m)}\|}{\|f\|}$

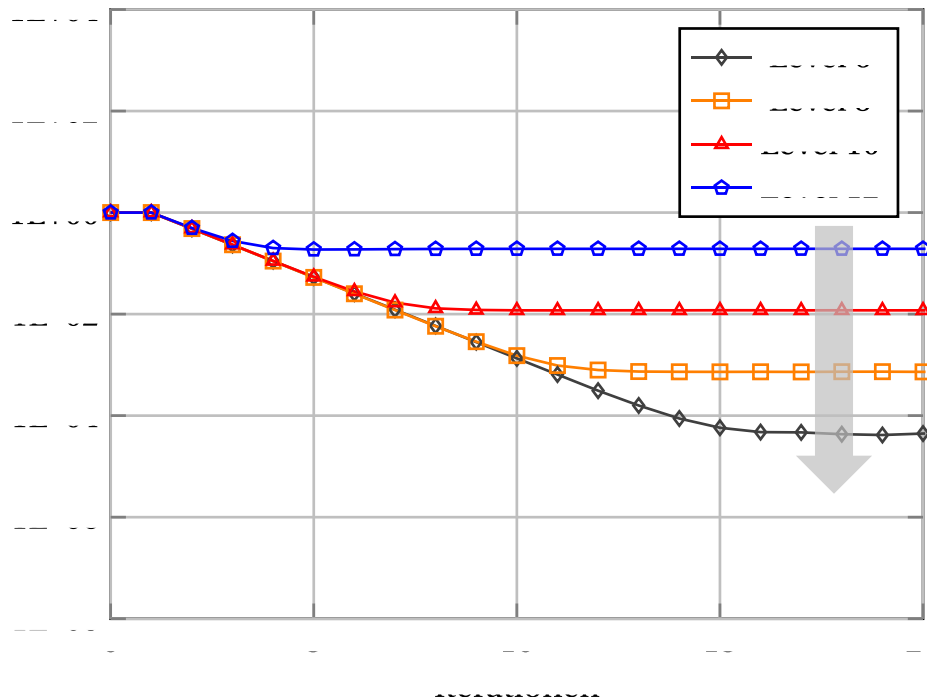


Relativer Fehler $\frac{\|u_{\text{exakt}} - u^{(m)}\|}{\|u_{\text{exakt}}\|}$

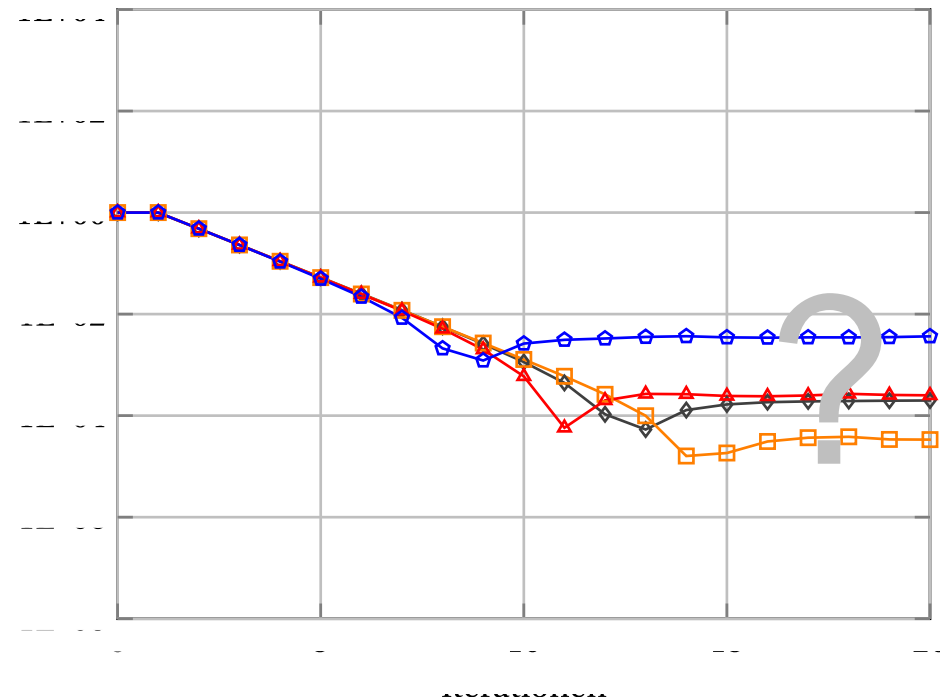


MG-Verfahren in Single

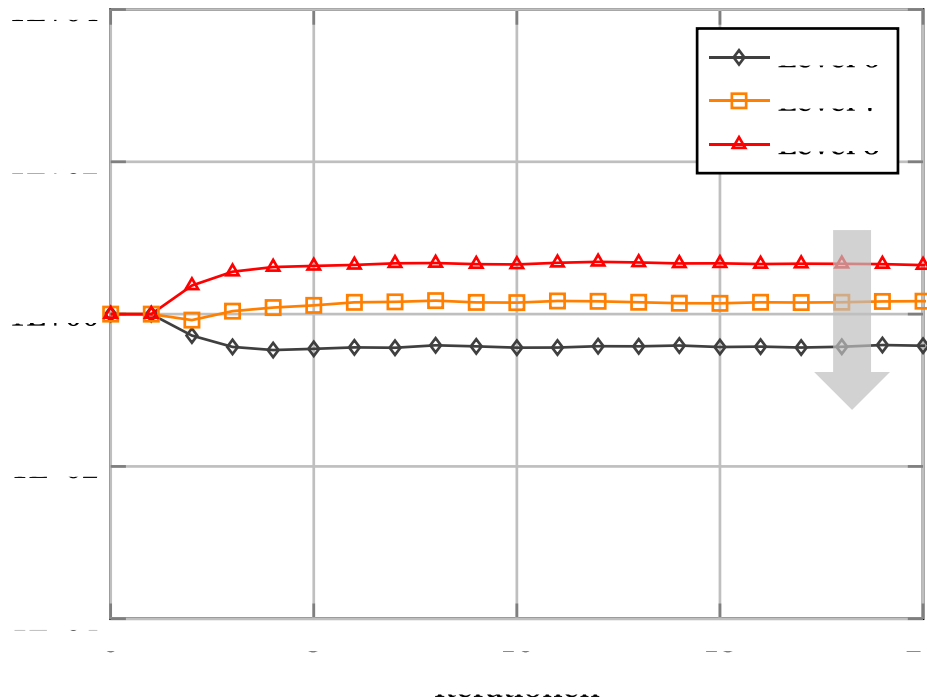
Relatives Residuum $\frac{\|f - Au^{(m)}\|}{\|f\|}$



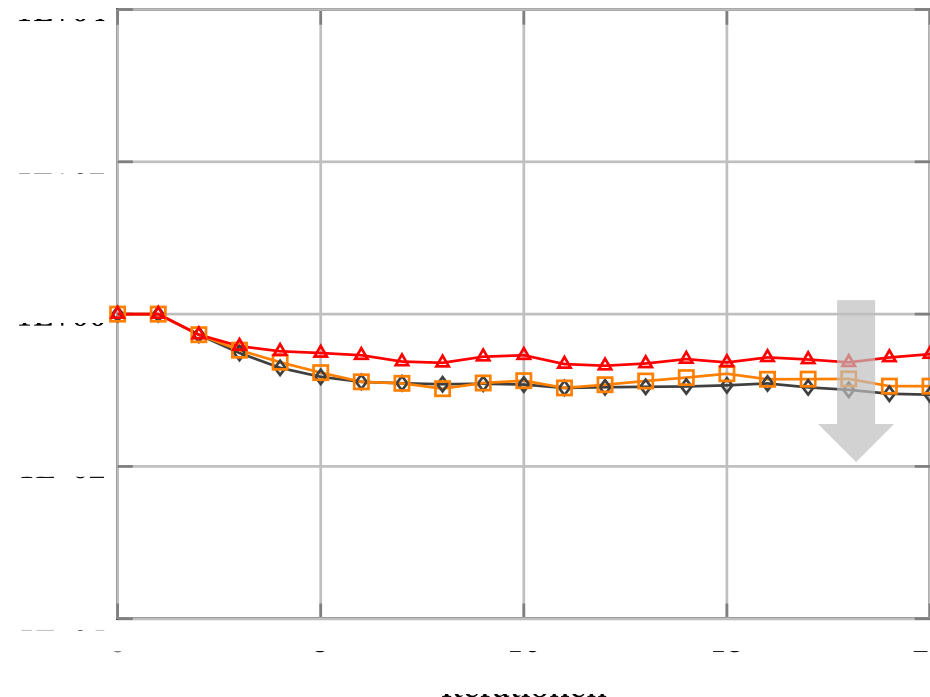
Relativer Fehler $\frac{\|u_{\text{exakt}} - u^{(m)}\|}{\|u_{\text{exakt}}\|}$

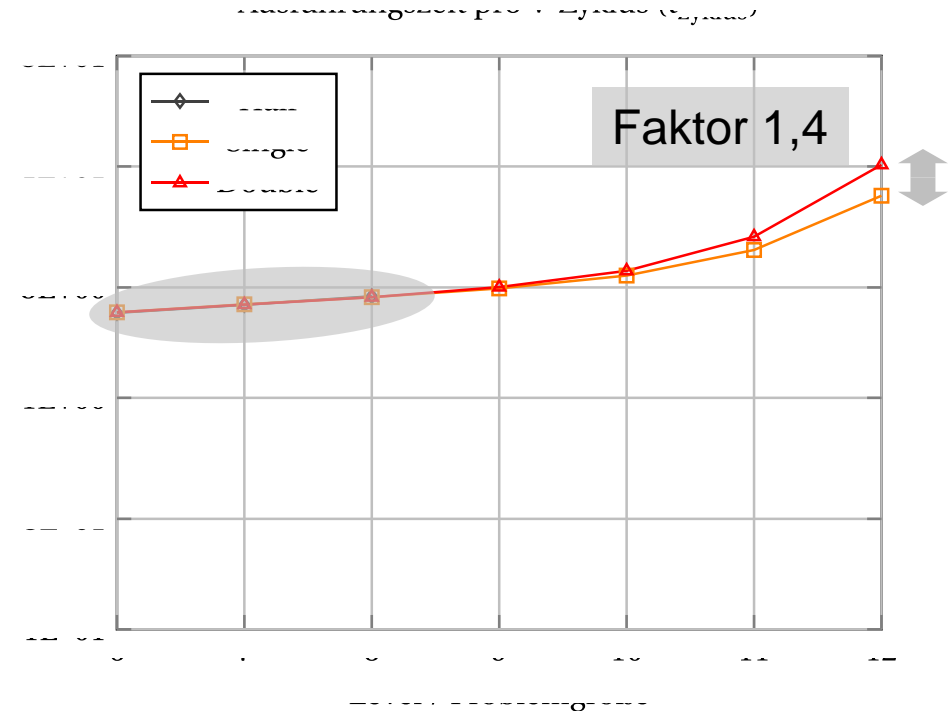
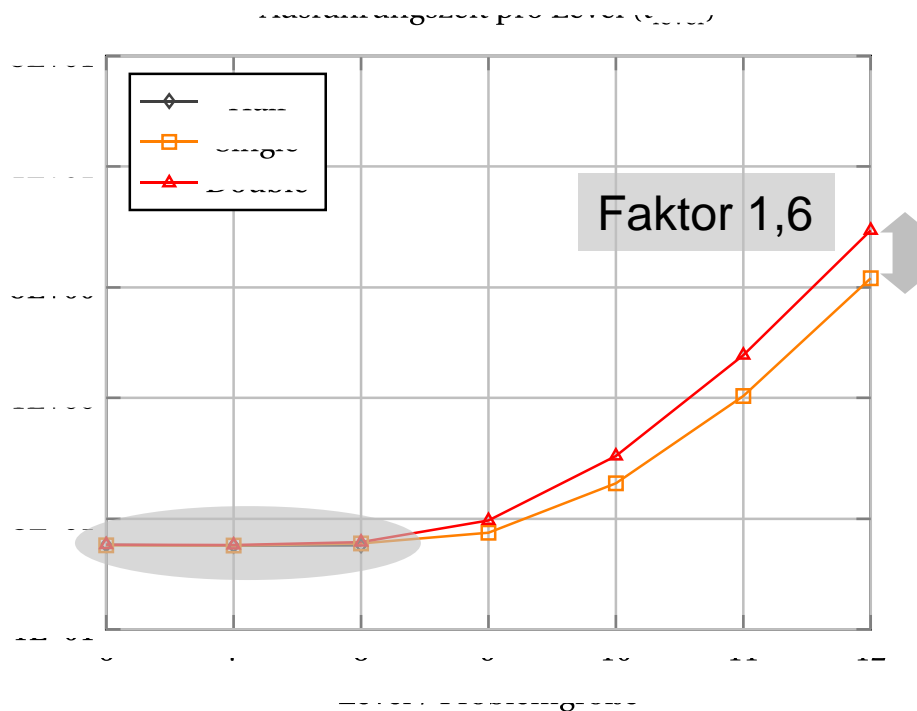


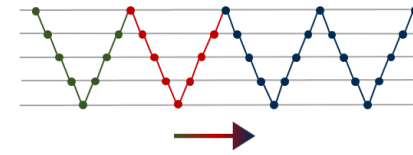
Relatives Residuum $\frac{\|f - Au^{(m)}\|}{\|f\|}$



Relativer Fehler $\frac{\|u_{\text{exakt}} - u^{(m)}\|}{\|u_{\text{exakt}}\|}$







H2SMG (Single/Double)

- Speed-Up hängt maßgeblich ab von

$$\alpha = \frac{\text{Anzahl Single - Iterationen}}{\text{Anzahl gesamter Iterationen}}$$

$$\text{Speed - Up} = \frac{d_{T_{vz}}}{\alpha \textcolor{red}{s}T_{vz} + (1-\alpha) d_{T_{vz}}}$$

| | | Problemgröße | |
|----------|------|--------------|------|
| | | 8 | 12 |
| α | 1,00 | 1,88 | 1,85 |
| | 0,75 | 1,54 | 1,53 |
| | 0,50 | 1,30 | 1,30 |
| | 0,25 | 1,13 | 1,13 |

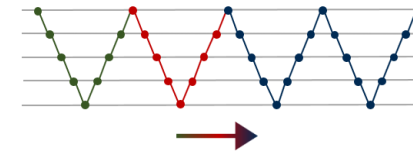
H3SMG (Half/Single/Double)

- Speed-Up hängt maßgeblich ab von α und

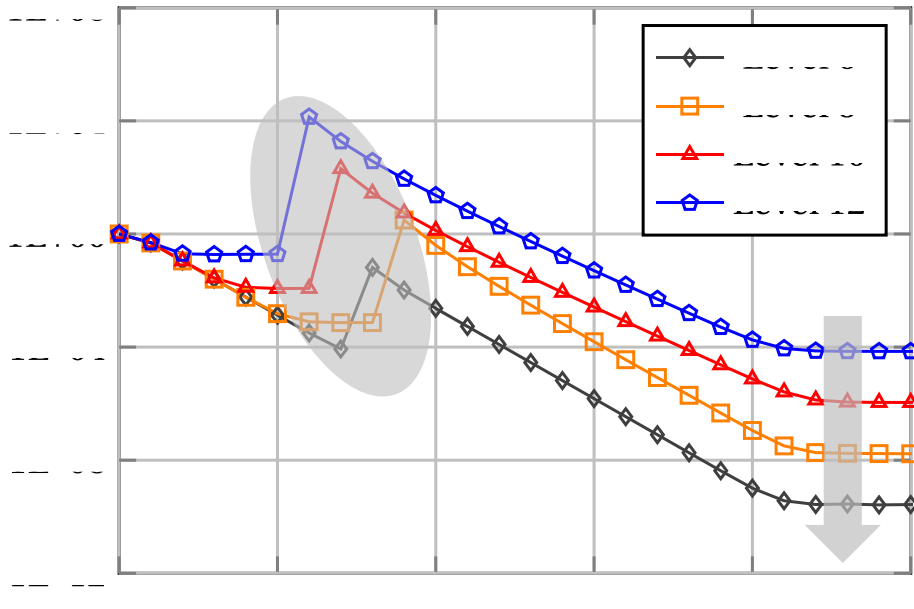
$$\beta = \frac{\text{Anzahl Half - Iterationen}}{\text{Anzahl gesamter Iterationen}}$$

$$\text{Speed - Up} = \frac{d_{T_{vz}}}{\beta \textcolor{teal}{h}T_{vz} + \alpha \textcolor{red}{s}T_{vz} + (1-\alpha-\beta) d_{T_{vz}}}$$

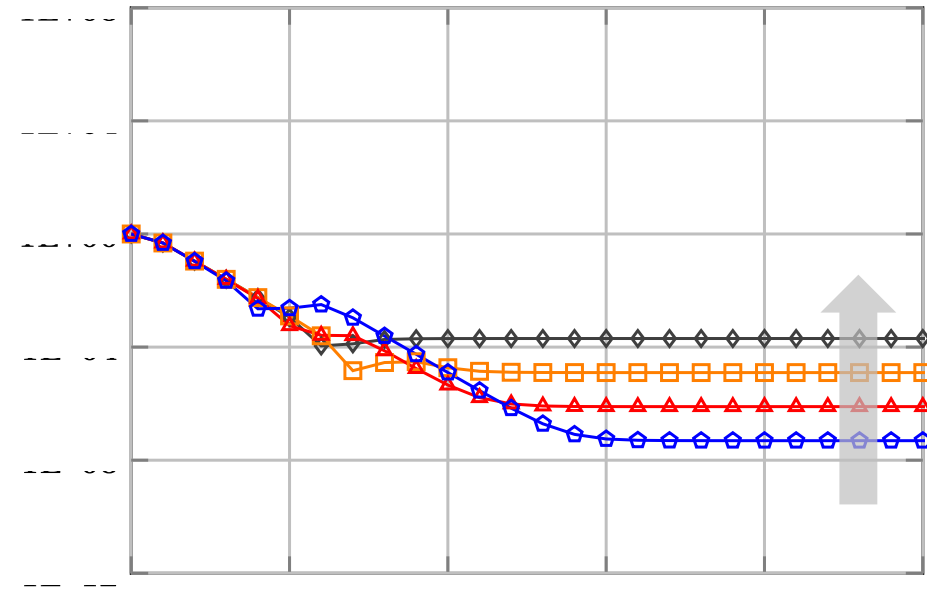
| | | Problemgröße | |
|--------------------|-------------|--------------|------|
| | | 8 | 12 |
| $(\alpha ; \beta)$ | (1,0 ; 0,0) | 3,34 | 3,24 |
| | (0,8 ; 0,2) | 2,89 | 2,82 |
| | (0,6 ; 0,2) | 2,06 | 2,03 |
| | (0,4 ; 0,4) | 1,88 | 1,85 |
| | (0,2 ; 0,6) | 1,73 | 1,71 |

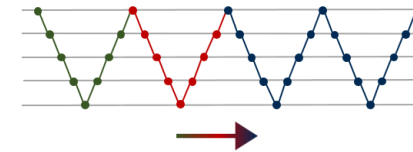
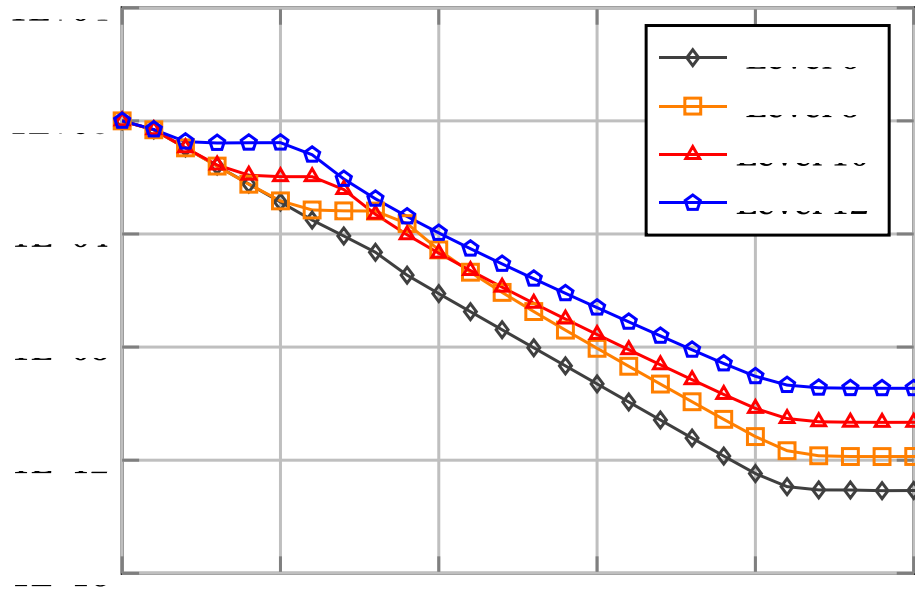
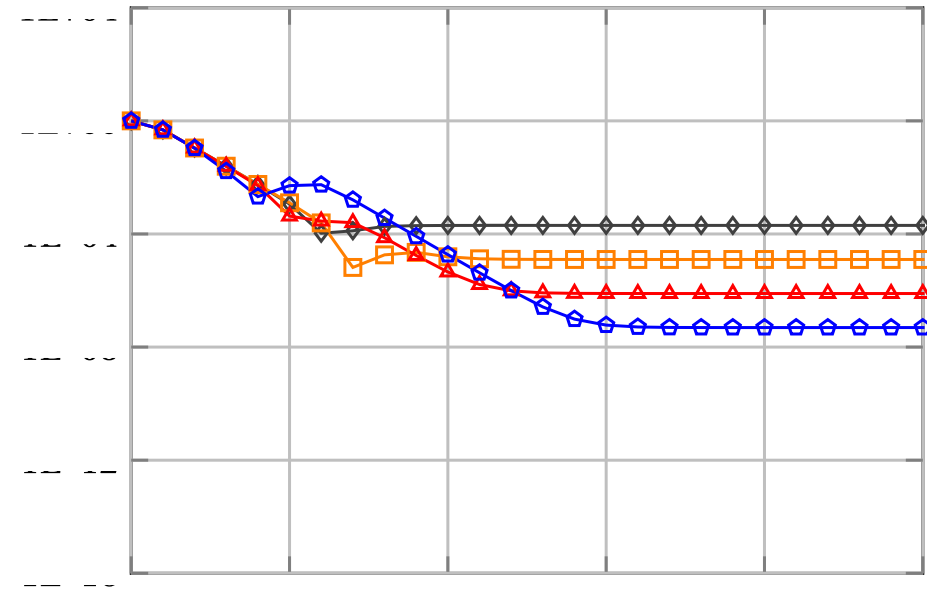


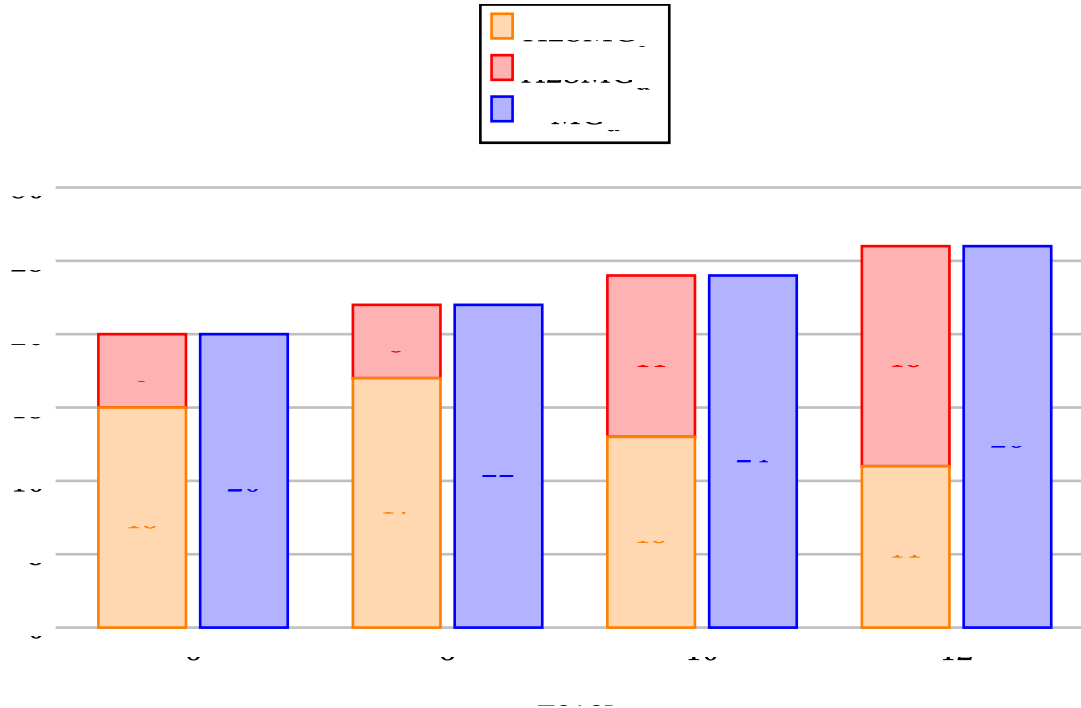
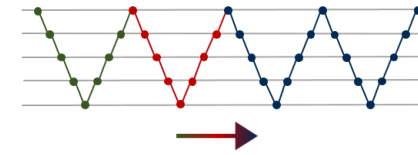
Relatives Residuum $\frac{\|f - Au^{(m)}\|}{\|f\|}$



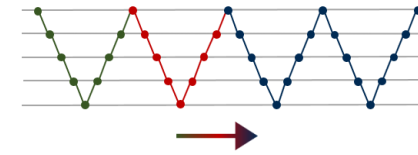
Relativer Fehler $\frac{\|u_{\text{exakt}} - u^{(m)}\|}{\|u_{\text{exakt}}\|}$



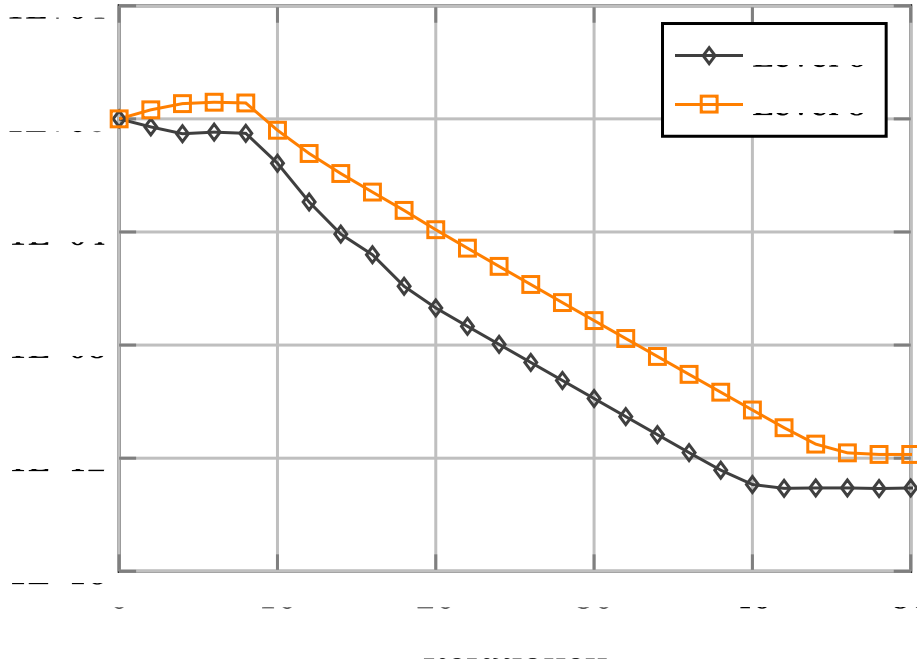
Relatives Residuum $\frac{\|f - Au^{(m)}\|}{\|f\|}$ Relativer Fehler $\frac{\|u_{\text{exakt}} - u^{(m)}\|}{\|u_{\text{exakt}}\|}$ 



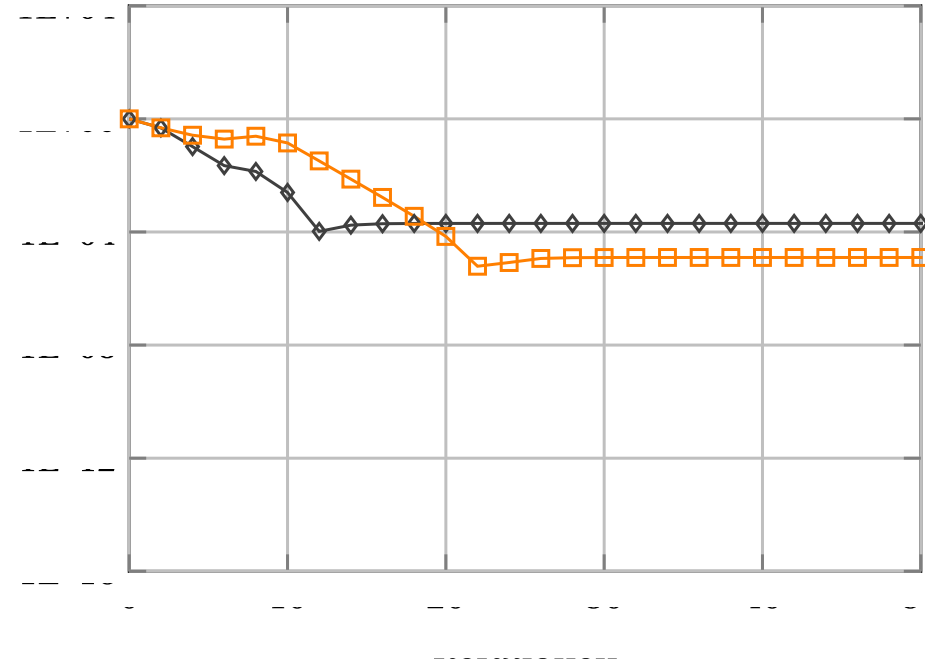
| Level | Exp. Speed-Up | Theo. Speed-Up | $ \Delta_{rf} $ |
|-------|---------------|----------------|-----------------|
| 6 | 1,05 | 1,58 | 8,80E-08 |
| 8 | 1,07 | 1,56 | 8,72E-07 |
| 10 | 1,09 | 1,33 | 2,29E-07 |
| 12 | 1,25 | 1,24 | 2,22E-07 |

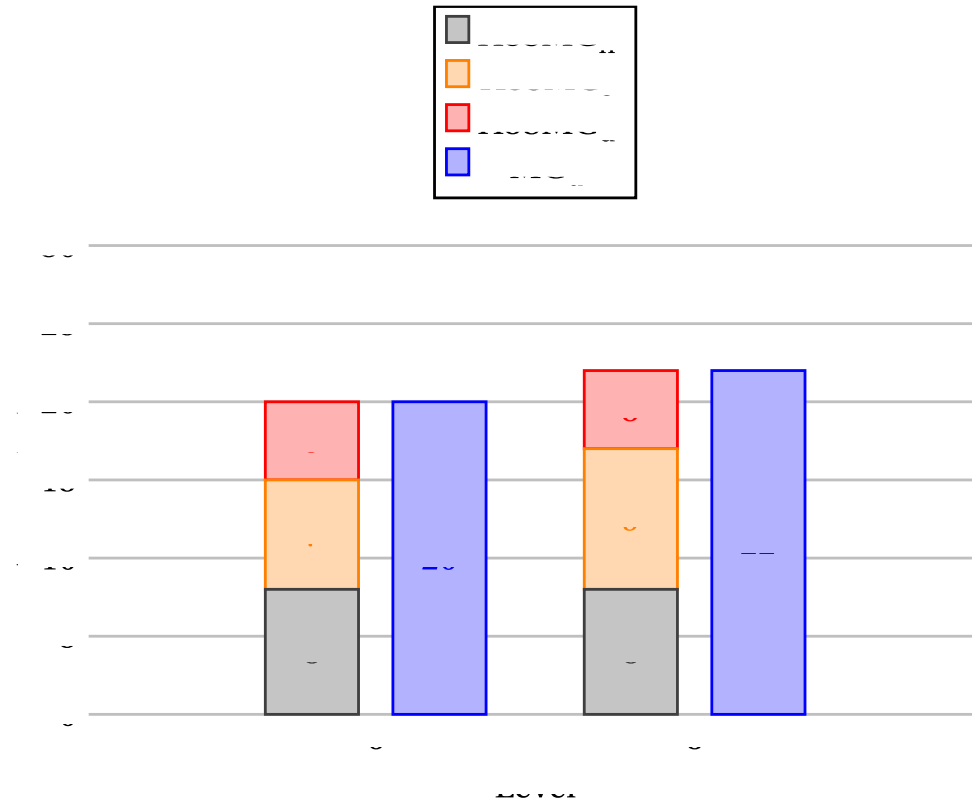
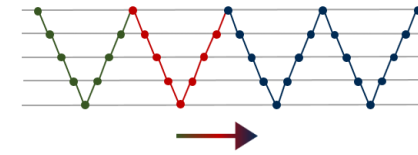


Relatives Residuum $\frac{\|f - Au^{(m)}\|}{\|f\|}$

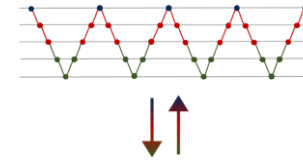


Relativer Fehler $\frac{\|u_{\text{exakt}} - u^{(m)}\|}{\|u_{\text{exakt}}\|}$





| Level | Exp. Speed-Up | Theo. Speed-Up | $ \Delta_{rf} $ |
|-------|---------------|----------------|-----------------|
| 6 | 1,05 | 1,87 | 5,73E-08 |
| 8 | 1,06 | 1,81 | 5,23E-06 |



V2SMG (Single/Double)

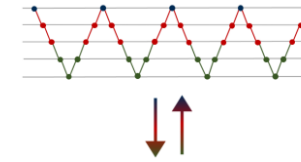
- Speed-Up hängt maßgeblich ab von
 $k_s = \text{Level des Wechsels von Single auf Double}$
- Speed - Up = $\frac{d_{T_{vz}}}{ds_{T_{vz}}(k_s)}$

| | | Problemgröße | |
|-------|----|--------------|------|
| | | 8 | 12 |
| k_s | 12 | | 1,90 |
| | 11 | | 1,13 |
| | 10 | | 1,03 |
| | 9 | | 1,00 |
| | 8 | 1,91 | |
| | 7 | 1,14 | |
| | 6 | 1,03 | |

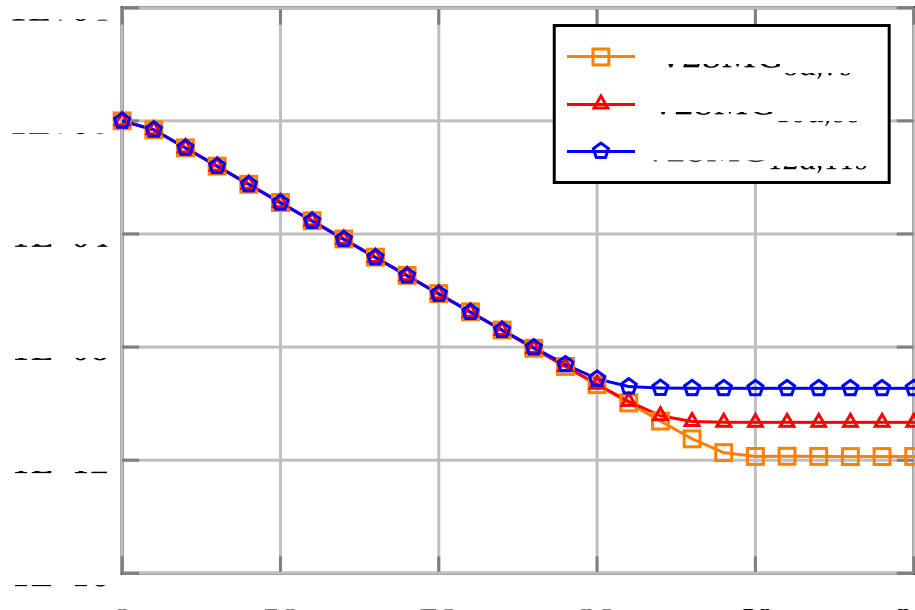
V3SMG (Half/Single/Double)

- Speed-Up hängt maßgeblich ab von k_s und
 $k_h = \text{Level des Wechsels von Half auf Single}$
- Speed - Up = $\frac{d_{T_{vz}}}{dsh_{T_{vz}}(k_h, k_s)}$

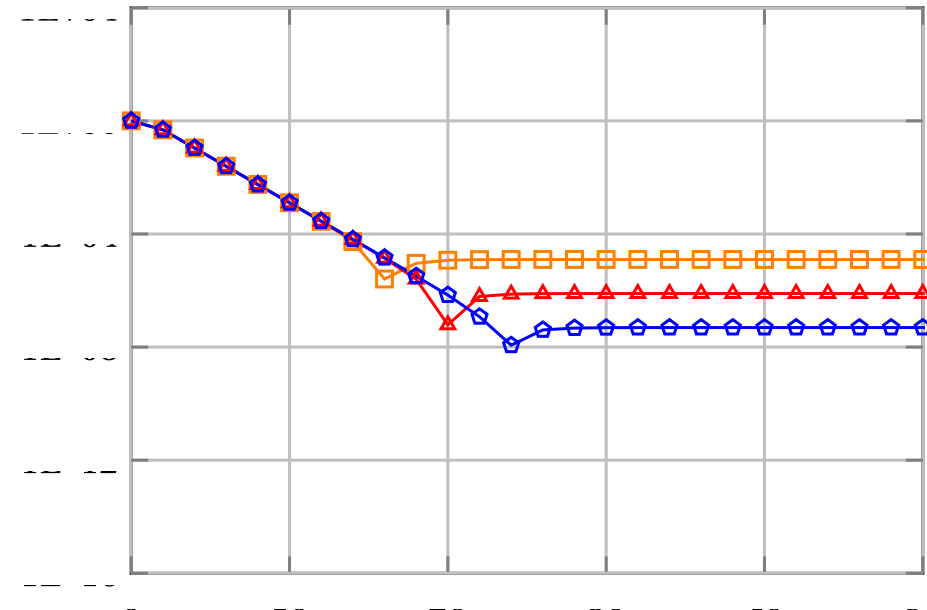
| | | Problemgröße | |
|---------------|-----------|--------------|------|
| | | 8 | 12 |
| $(k_s ; k_h)$ | (12 ; 11) | | 2,10 |
| | (12 ; 10) | | 1,95 |
| | (11 ; 10) | | 1,15 |
| | (11 ; 9) | | 1,14 |
| | (8 ; 7) | 2,27 | |
| | (8 ; 6) | 2,13 | |
| | (7 ; 6) | 1,30 | |

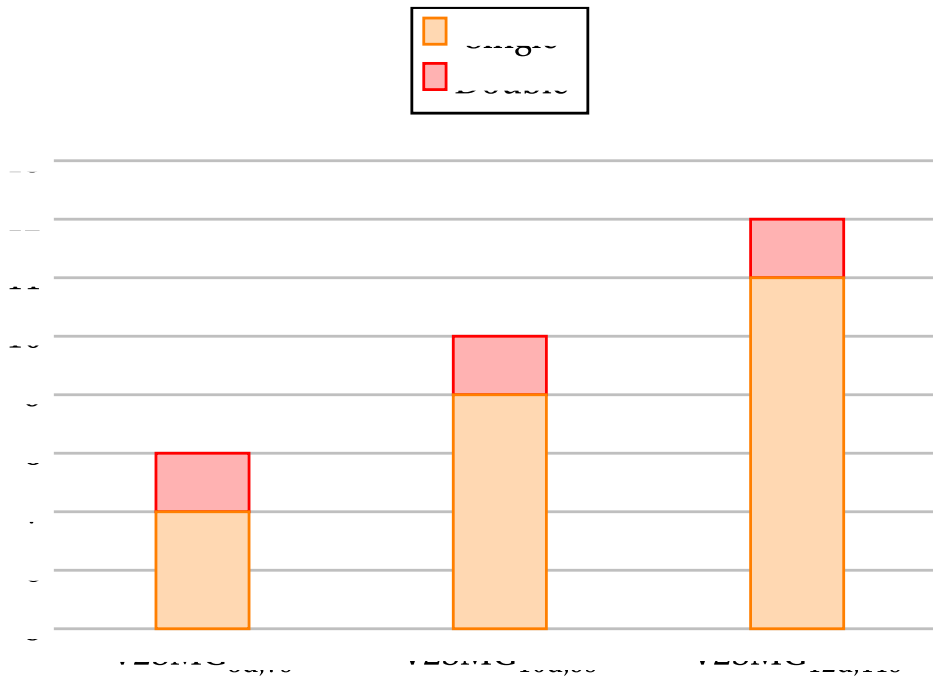
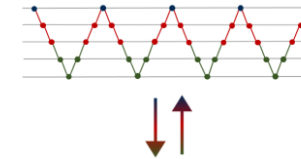


Relatives Residuum $\frac{\|f - Au^{(m)}\|}{\|f\|}$

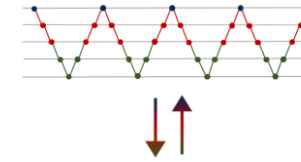


Relativer Fehler $\frac{\|u_{\text{exakt}} - u^{(m)}\|}{\|u_{\text{exakt}}\|}$

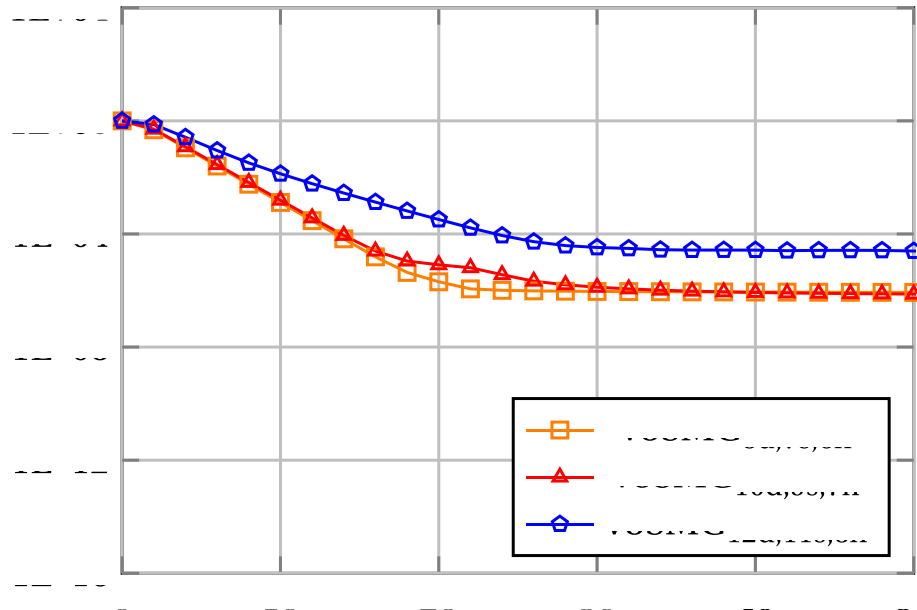




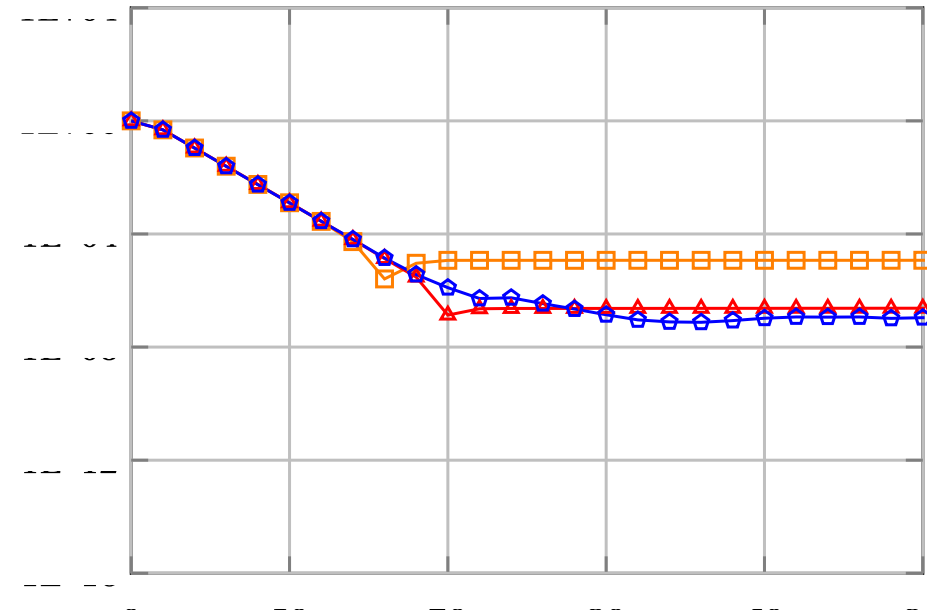
| Konfiguration | Exp. Speed-Up | Theo. Speed-Up | $ \Delta_{rf} $ |
|---------------|---------------|----------------|-----------------|
| 8d,7s | 1,06 | 1,92 | 1,23E-12 |
| 10d,9s | 1,07 | 1,91 | 1,22E-12 |
| 12d,11s | 1,21 | 1,90 | 1,30E-12 |

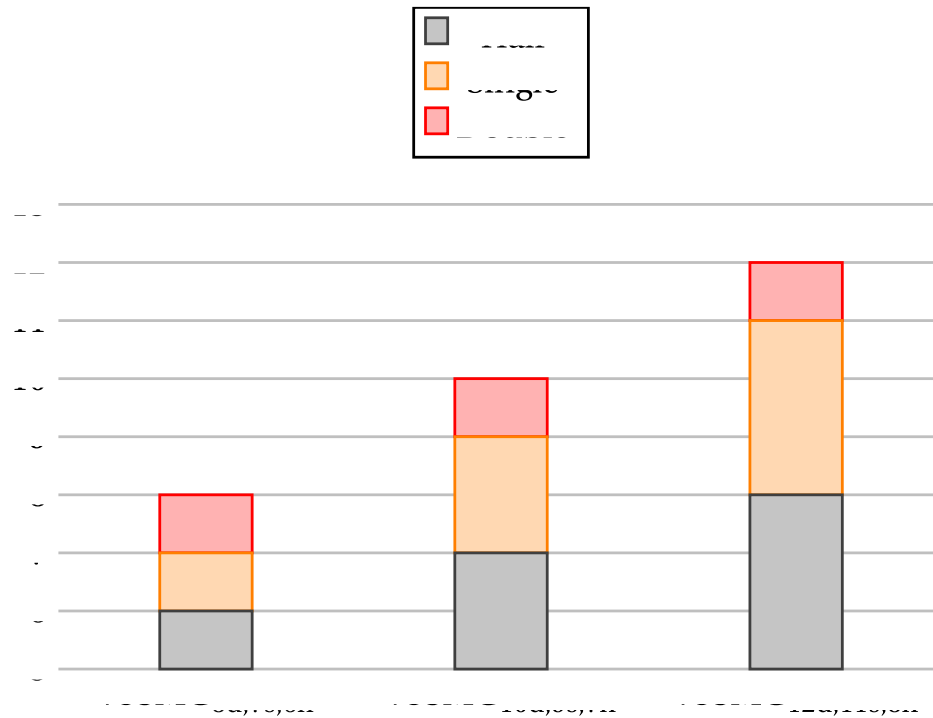
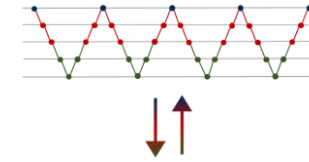


Relatives Residuum $\frac{\|f - Au^{(m)}\|}{\|f\|}$



Relativer Fehler $\frac{\|u_{\text{exakt}} - u^{(m)}\|}{\|u_{\text{exakt}}\|}$





| Konfiguration | Exp. Speed-Up | Theo. Speed-Up | $ \Delta_{rf} $ |
|---------------|---------------|----------------|-----------------|
| 8d,7s,6h | 1,06 | 2,27 | 6,76E-7 |
| 10d,9s,7h | 1,07 | 1,96 | 5,14E-7 |
| 12d,11s,8h | 1,20 | 1,91 | 3,04E-7 |

Fazit

- ⊕ Kaum Auswirkungen auf den relativen Fehler
- ⊕ Diskretisierungsfehler verhält sich wie erwartet
- ⊕ Lässt sich auf alle (F)-MG-Verfahren übertragen
- ⊕ Sehr viel Potential für weitere Untersuchungen
- ⊖ Abbruchkriterien für den Wechsel sind schwer zu definieren
- ⊖ Problemgröße ist zum Teil eingeschränkt (H3SMG)
- ⊖ Feintuning für die einzelnen Datentypen erforderlich
- ⊖ Programmierung technisch deutlich aufwändiger

Ausblick

- Verwendung als gemischt genauer Vorkonditionierer
- Größerer Speed-Up für Mittelklasse-Grafikkarten
- Kombination mit emulierten Datentypen (z.B. Quadruple)
- Gemischt genauer 9-Punkte Stencil
- FPGAs mit beliebiger Genauigkeit
- LR-Zerlegung mit Tensor-Core auf unterstem Level

- Jinn-Liang Liu. Poisson's Equation in Electrostatics. <http://www.nhcue.edu.tw/~jinnliu/proj/Device/3DPoisson.pdf>. Abgerufen: Juni 2019.
- IEEE Computer Society. IEEE Standard for Floating-Point Arithmetic. http://www.dsc.ufcg.edu.br/~cnum/modulos/Modulo2/IEEE754_2008.pdf. Abgerufen: Juni 2019.
- ufcg.edu.br/~cnum/modulos/Modulo2/IEEE754_2008.pdf. Abgerufen: Juni 2019. Thomas Jahn. "Implementierung numerischer Algorithmen auf CUDA-Systemen". Abgerufen: Juni 2019. Diplomarbeit. Universität Bayreuth.
- NVIDIA Corporation. NVIDIA CUDA - Compute Unified Device Architecture - Programming Guide - Vers http://developer.download.nvidia.com/compute/cuda/1.0/NVIDIA_CUDA_Programming_Guide_1.0.pdf. Abgerufen: Mai 2019.
- The Khronos Group Inc. Khronos OpenCL Registry. <https://www.khronos.org/registry/OpenCL/>. Abgerufen: Juni 2019.
- NVIDIA Corporation. Whitepaper - NVIDIA's Next Generation CUDA™ Compute Architecture: Fermi™. https://www.nvidia.com/content/PDF/fermi_white_papers/NVIDIA_Fermi_Compute_Architecture_Whitepaper.pdf. Abgerufen: Februar 2019.

- NVIDIA Corporation. NVIDIA Tesla V100 GPU Architecture. <https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf>. Abgerufen: Februar 2019.
- James H. Wilkinson. Rundungsfehler. pub-Springer:adr-B: pub-Springer, 1969.
- Cleve B. Moler. “Iterative Refinement in Floating Point”. In: J. ACM 14.2 (Apr. 1967), S. 316–321. ISSN: 0004-5411. DOI: 10.1145/321386.321394. URL: <http://doi.acm.org/10.1145/321386.321394>.
- Alfredo Buttari et al. “Mixed Precision Iterative Refinement Techniques for the Solution of Dense Linear Systems”. In: The International Journal of High Performance Computing Applications 21.4 (2007), S. 457–466. DOI: 10.1177/1094342007084026. URL: <https://doi.org/10.1177/1094342007084026>.
- Alfredo Buttari et al. “Using Mixed Precision for Sparse Matrix Computations to Enhance the Performance while Achieving 64-bit Accuracy”. In: ACM Transactions on Mathematical Software 34 (Juli 2008). DOI: 10.1145/1377596.1377597.

- Dominik Göddeke, Robert Strzodka und Stefan Turek. “Performance and accuracy of hardware-oriented native-, emulated- and mixed-precision solvers in FEM simulations”. In: International Journal of Parallel, Emergent and Distributed Systems 22.4 (2007), S. 221–256. DOI: 10.1080/17445760601122076. URL: <https://doi.org/10.1080/17445760601122076>.
- Dominik Göddeke. Wissenschaftliches Rechnen. Vorlesungsskript. März 2017.
- Wolfgang Hackbusch. Multi-grid methods and applications. Springer, 1985.
- Dominik Göddeke. “Fast and Accurate Finite-Element Multigrid Solvers for PDE Simulations on GPU Clusters”. Diss. Technische Universität Dortmund, Feb. 2010.
- NVIDIA Corporation. CUBLAS Library - User Guide. https://docs.nvidia.com/cuda/pdf/CUBLAS_Library.pdf. Abgerufen: August 2019.
- NVIDIA Corporation. NVIDIA TESLA V100 GPU ACCELERATOR. <https://images.nvidia.com/content/technologies/volta/pdf/tesla-volta-v100-datasheet-letterfml-web.pdf>. Abgerufen: Juni 2019
- NVIDIA Corporation. NVIDIA TESLA V100 GPU ACCELERATOR. <https://www.anandtech.com/show/12576/nvidia-bumps-all-tesla-v100-models-to-32gb>. Abgerufen: Juni 2019.

- T. Washio C.W. Oosterlee. On the Use of Multigrid as a Preconditioner. <https://pdfs.semanticscholar.org/dcc1/9ad91450753e7f47157e323fade5c2b4e320.pdf>. Abgerufen: Juli 2019.
- Osamu Tatebe. The Multigrid Preconditioned Conjugate Gradient Method. <http://www.hpcs.cs.tsukuba.ac.jp/~tatebe/research/paper/CM93-tatebe.pdf>. Abgerufen: Juli 2019.
- **Bilder**
 - Scientific Datatype, https://cdn.icon-icons.com/icons2/539/PNG/512/atom_icon-icons.com_53030.png
 - Graphical Datatype, <https://icon-library.net/images/games-icon/games-icon-18.jpg>
 - Neural Network Datatype, <https://icon-library.net/images/icon-artificial-intelligence/icon-artificial-intelligence-6.jpg>
 - Time Measurement, <https://cdn2.iconfinder.com/data/icons/social-productivity-line-black-1/3/14-512.png>
 - Numerical Error, <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRF2dNX6SpaYge-O3CF-XLMrAl0l1hNNtXzwDAI4txzKA0EpuwH>