

No. 637

February 2021

**Eine Machbarkeitsstudie zu schnellen
FEM-Poisson-Lösern auf Beschleunigerhardware
als Beispiel für Hardware-orientierte Numerik**

**S. Turek, D. Ruda, D. Ribbrock,
H. Poelstra, P. Zajac**

ISSN: 2190-1767

Eine Machbarkeitsstudie zu schnellen FEM-Poisson-Lösern auf Beschleunigerhardware als Beispiel für Hardware-orientierte Numerik

S. Turek, D. Ruda, D. Ribbrock, H. Poelstra, P. Zajac (TU Dortmund)

I) Motivation

Moderne Computersysteme beinhalten heutzutage mehr und mehr „Beschleunigerhardware“, insbesondere Grafikkarten mit KI-Komponenten (z.B. NVIDIA Volta V100 mit „Tensorcores“). Diese versprechen mehrere TFLOP/s Rechenleistung (obige V100 mit ca. 8 TFLOP/s in doppelter Genauigkeit und sogar mit bis zu 125 TFLOP/s in „half precision“ [1,2,3]). Damit stellt sich natürlich insbesondere für Anwendungen in der numerischen Kontinuumsmechanik, die oftmals Modelle auf der Basis von partiellen Differentialgleichungen verwenden, die dann beispielweise mit Finite Elemente Methoden (FEM) diskretisiert werden, die Frage:

Können Basisbausteine für FEM-Simulationen bei signifikanter Ausnutzung dieser sehr hohen Rechenleistungen auf moderner Beschleunigerhardware realisiert werden?

Ziel dieser „proof-of-concept“ Studie (aus einem laufenden Promotionsvorhaben) ist es, unter Ausnutzung von problemspezifischen Ansätzen der Hardware-orientierten Numerik zu zeigen, wie zugehörige FEM-Simulationskomponenten um mehrere Größenordnungen im Vergleich zu „Standardmethodik“ (konkret: geometrisches Mehrgitterverfahren) auf „Standardhardware“ (konkret: Multicore CPUs) beschleunigt werden können.

II) Ein Beispiel für moderne Beschleunigerhardware

Als Prototyp für moderne „Beschleunigerhardware“ betrachten wir die genannte NVIDIA Volta V100 SXM2 [1], die durch die folgenden Leistungsdaten charakterisiert werden kann: 7.8 TFLOP/s in „double precision“ (DP), 15.7 TFLOP/s in „single precision“ (SP), 31.3 TFLOP/s in „half precision“ (HP) und sogar 125 TFLOP/s bei HP mit Tensorcores; die Speicherbandbreite beträgt 900 GByte/s. Im Vergleich dazu betrachten wir die folgende „x64“ Konfiguration (AMD EPYC 7542, 32 Cores pro CPU, 128 MB L3-Cache [4]), die moderne („Standard“) Rechnerplattformen, insbesondere in Rechenzentren, für FEM-Simulationen prototypisch repräsentieren soll. Hier lauten die entsprechenden Leistungsdaten: 1.5 TF/s in DP, 3 TFLOP/s in SP (HP wird nicht unterstützt) und eine Speicherbandbreite von 205 GByte/s.

Diese angegebenen TFLOP/s („Peak“) Raten beruhen auf BLAS3 Techniken, die für Anwendungen mit vollbesetzten Matrizen relevant sind (und tatsächlich auch im Rahmen von direkten Verfahren zur Lösung linearer Gleichungssysteme erreicht werden, allerdings dann mit bekannter kubischer Komplexität). Dagegen basieren moderne Lösungsverfahren für Steifigkeitsmatrizen, die aus FEM-Diskretisierungen resultieren und i.d.R. sehr dünnbesetzt („sparse“) sind, auf iterativen Lösungsmethoden vom Typ Krylov-Mehrgitter-Verfahren. Diese wiederum nutzen vor allem entsprechende Matrix-Vektor-Anwendungen als schnelle Basisbausteine, deren Performance allerdings durch die Speicherbandbreite charakterisiert wird. Für den betrachteten Testfall (hier: Poisson-Probleme, bilineare (Q1) FEM, 2D, Einheitsquadrat, CSR-Speichertechnik [5]) bedeutet das für die beiden betrachteten Hardwarekonfigurationen (NVIDIA Volta V100, AMD EPYC 7542), dass für solche „sparse“ Matrix-Vektor (MV) bzw. Matrix-Matrix (MM) Anwendungen, d.h. für MV-Anwendungen mit vielen Vektoren bzw. für die simultane Lösung von Poisson-Problemen mit vielen rechten Seiten, die erzielte Rechenleistung weit entfernt von den theoretischen „Peak“ Raten ist: Die V100 Grafikkarte ist für den untersuchten Testfall zwar schneller als die AMD CPU, aber eben „nur“ um einen Faktor 4-6 (für Gitterweite $h = 1/1024$, d.h. ca. 1 Million Unbekannte, in obigem Testfall),

und die Diskrepanz zu den „erhofften“ Peak Raten im mehrfachen TFLOP/s Bereich ist offensichtlich (siehe Abb. 1).

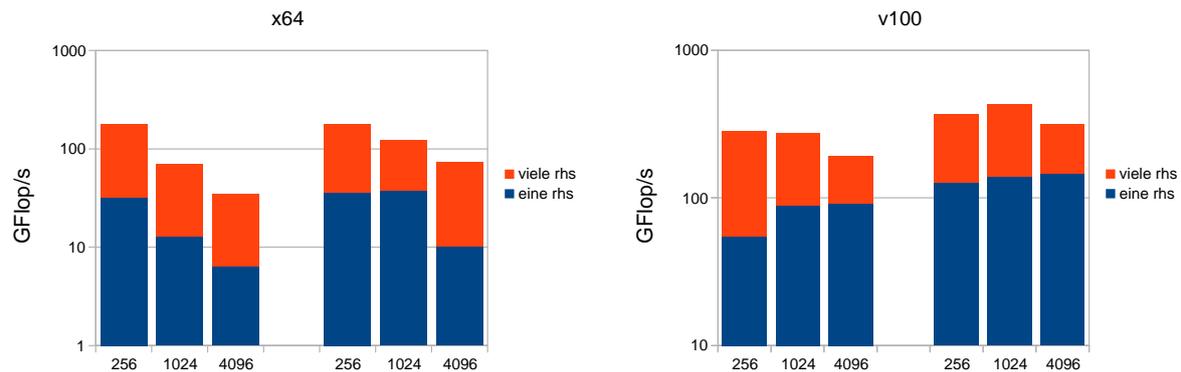


Abb.1: GFlop/s Raten für „sparse“ MV („eine rhs“) und „sparse“ MM („viele rhs“) in DP (jeweils linke Säulen) und in SP (jeweils rechte Säulen).

Die tatsächlich erzielbare Rechenleistung (siehe Abb. 2) erhöht sich signifikant, wenn entsprechende Ergebnisse für „dense“ Matrix-Vektor bzw. insbesondere für Matrix-Matrix-Multiplikationen betrachtet werden. Auffällig ist dabei, dass gerade in „lower precision“, d.h. in „single“ und vor allem in „half precision“, eine sehr hohe Performance erreicht werden kann, da zusätzlich die genannten Tensorcores [6] der V100 ausgenutzt werden können.

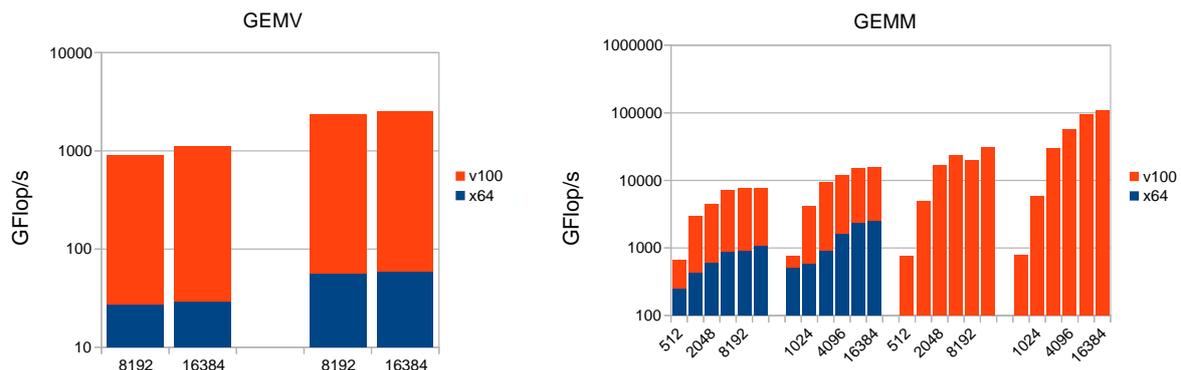


Abb.2: GFlop/s Raten für „dense“ MV („GEMV“) und „dense“ MM („GEMM“) in DP (jeweils linke Säulen), in SP (jeweils zweite Säulen von links), in HP (dritte Säulen von links für V100) und für HP mit Tensorcores (rechte Säulen für V100).

Diese Ergebnisse illustrieren, wie groß einerseits die Unterschiede zwischen „realer“ Rechenleistung für FEM-Anwendungen, die in der Regel auf „sparse“ MV-Techniken in „double precision“ beruhen, und erreichbarer Peak Performance sind. Und sie zeigen andererseits, wie hoch das Potential für numerische Simulationen in der Kontinuumsmechanik sein kann, wenn tatsächlich „lower precision“ und Tensorcores ausgenutzt werden können. Aber natürlich stellt sich dann sofort die Frage:

Kann man FEM-Simulationen ohne signifikante Genauigkeitsverluste in „lower precision“ durchführen und können zugehörige schnelle Löser realisiert werden, die zu großen Teilen auf „dense“ MV- bzw. MM-Anwendungen beruhen und damit die sehr hohen Rechenleistungen moderner Beschleunigerhardware ausnutzen können?

Im Folgenden werden wir in Kurzform zuerst das Konzept des „prehandling“ [7,8] erläutern. Dieses erlaubt, FEM-Matrizen, die zu (linearen) Differentialoperatoren – im Folgenden für Poisson-Probleme als wesentliche Bausteine in vielen numerischen Simulationen – gehören, in

„lower precision“ zu behandeln und dennoch eine vergleichbare Genauigkeit wie in doppelter Genauigkeit zu erhalten. Als weiteren Schritt werden wir eine spezielle Variante dieser „Vorbehandlung“, nämlich mittels „hierarchischer Finite Elemente Methoden“ (HFEM [9]), vorstellen. Für diese lassen sich spezielle Schurkomplement-artige Lösungsverfahren entwickeln und realisieren, die auf Multiplikationen mit vollbesetzten (Teil-)Matrizen beruhen und damit auch das Potential der Tensorcores ausnutzen können. Im anschließenden Teil stellen wir erste numerische Studien zu diesen Überlegungen vor, die als „proof-of-concept“ demonstrieren, dass tatsächlich mittels spezieller Hardware-orientierter Techniken moderne Beschleunigerhardware adäquat ausgenutzt werden kann.

III) Prehandling als Konzept für FEM-Matrizen in „lower precision“

Die FEM-Steifigkeitsmatrizen (im Folgenden mit A_h bezeichnet) zu elliptischen Differentialgleichungen (von 2.Ordnung), und damit insbesondere für Poisson-Probleme als betrachtete Prototypen, lassen sich dadurch charakterisieren, dass sich ihre „Konditionszahl“ wie $\text{cond}(A_h) = \mathcal{O}(h^{-2})$ verhält, wobei h die Gitterweite beschreibt. Da Störungen in den Daten (z.B. in den Matrix- bzw. Lastvektoreinträgen durch Rundungsfehler) auch zu Störungen in der zugehörigen Lösung führen, die von $\text{cond}(A_h)$ abhängen, erhält man für die (relativen) „Störungsfehler“

$$\frac{\|u_h - \tilde{u}_h\|}{\|u_h\|} \approx \text{cond}(A_h) \cdot \text{TOL} = \mathcal{O}(h^{-2}) \cdot \text{TOL}$$

wobei TOL (mindestens) die „Maschinengenauigkeit“ beschreibt, d.h. $\text{TOL} \approx 10^{-16}$ für DP, 10^{-7} für SP und (nur) $\text{TOL} \approx 10^{-3}$ für HP. Da sich gleichzeitig – für lineare bzw. multilineare FEM-Ansätze – die (relativen) Diskretisierungsfehler wie $\mathcal{O}(h^2)$ verhalten

$$\|u - u_h\|/\|u\| = \mathcal{O}(h^2)$$

ergibt sich die Problematik, dass sich der Gesamtfehler $u - \tilde{u}_h$ aus Anteilen des Diskretisierungs- und des Störungsfehlers zusammensetzt, die sich jeweils wie $\mathcal{O}(h^2)$ bzw. $\mathcal{O}(h^{-2})$ verhalten: Konkret bedeutet das, dass h nicht zu klein gewählt werden darf, da ansonsten der Störungsfehler dominiert, wie Abb. 3 zeigt:

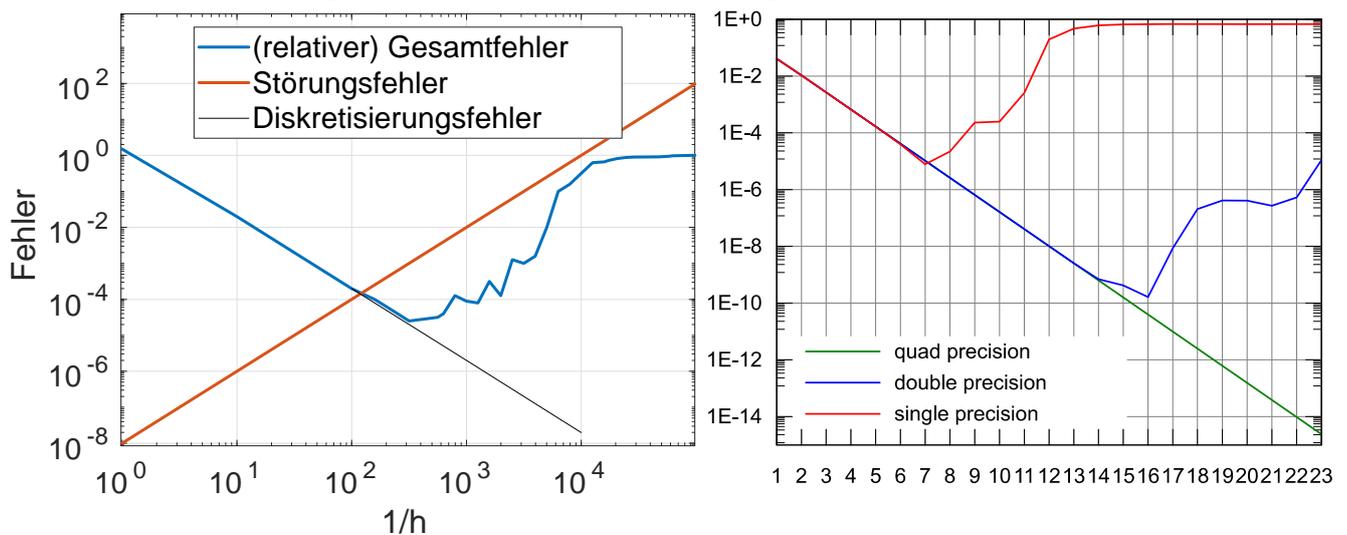


Abb.3: Links: Illustrative Darstellung der Gesamtfehler sowie der Störungs- und Diskretisierungsfehler in SP für einen bilinearen FEM-Ansatz für das Poisson-Problem; rechts: Realer Verlauf der \mathcal{L}^2 -Gesamtfehler für einfache, doppelte und vierfache Rechengenauigkeiten bei Anwendung von Standard-FEM in 1D in Abhängigkeit vom Verfeinerungslevel (Level 10 entspricht $h = 1/1024$).

Um beide Fehlerarten zu äquilibrieren, zeigt eine einfache Rechnung, dass die Gitterweite h (asymptotisch) nicht kleiner als $\mathcal{O}(\text{TOL}^{1/4})$ gewählt werden darf. Das führt aber für SP und insbesondere für HP zu sehr groben Gitterweiten und macht damit die Lösung von sehr großen Problemen, für die eigentlich (parallele) Hochleistungsrechner mit solcher Beschleunigerhardware benötigt werden, unmöglich.

Die zentrale Idee des „prehandling“ (bzw. der „Vorbehandlung“) besteht jetzt darin, die Steifigkeitsmatrix A_h explizit in eine modifizierte Matrix \tilde{A}_h zu transformieren, so dass – bei exakter Rechengenauigkeit – die resultierenden Gleichungssysteme zu identischen Ergebnissen führen und dabei folgende drei Bedingungen erfüllt sind:

1. starke Reduktion der Konditionszahl: $\text{cond}(\tilde{A}_h) \ll \text{cond}(A_h)$
2. ähnliche „sparsity“ (Dünnbesetztheit) der transformierten Matrix \tilde{A}_h
3. effiziente Transformation (z.B. in $\mathcal{O}(N \log N)$, wenn N die Problemgröße ist)

Ein Kandidat für eine solche Transformation und damit für „prehandling“ – zumindest in 2D – ist die genannte Methode der hierarchischen Finiten Elemente, die zu Konditionszahlen der Größe $\text{cond}(\tilde{A}_h) = \mathcal{O}\left(\left(\log \frac{1}{h}\right)^2\right)$ [9] führt und damit die Verwendung von „single“ und sogar „half precision“ erlaubt (siehe Abb. 4). Konkret erhalten wir für die untersuchten Testkonfigurationen Konditionszahlen in der (maximalen) Größenordnung $\text{cond}(\tilde{A}_h) \approx 50 - 100$, die für (relative) Genauigkeiten im Bereich von ca. 1% ausreichen, was aber für komplexe technische Simulationen in den meisten Fällen eine realistische Forderung darstellt.

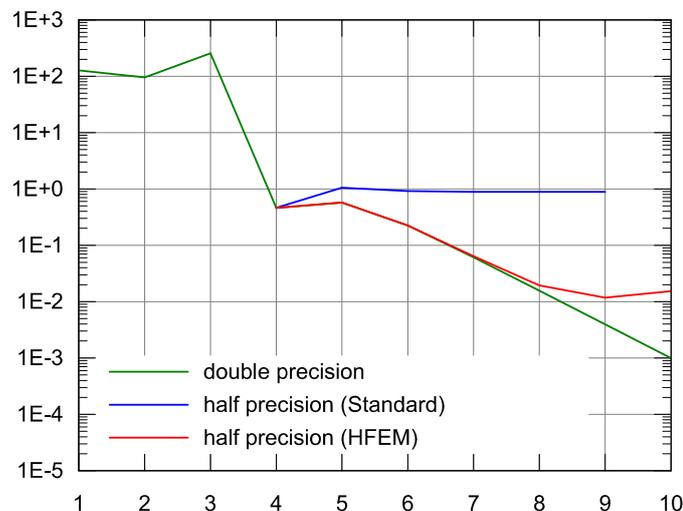


Abb. 4: Vergleich der \mathcal{L}^2 -Fehler bei Verwendung von Standard-FEM (blau) und Prehandling mittels HFEM (rot) jeweils in HP sowie der Referenzlösung in DP (grün, in diesem Bereich unabhängig von Prehandling) in 2D in Abhängigkeit vom Verfeinerungslevel (Level 10 entspricht wieder $h = 1/1024$) für eine stark oszillierende exakte Lösung. Ohne Prehandling weicht der Fehler ab Level 4 ab, mit Prehandling dagegen ab Level 8 bzw. 9.

Ausführliche numerische Analysen und Tests in [8] belegen, dass damit tatsächlich eine geeignete Technik gefunden werden konnte, um zumindest für FEM-Diskretisierungen von Poisson-Problemen in 2D auf allgemeinen Rechengittern (d.h. Gittern, die von einem unstrukturierten Grobgitter ausgehend durch sukzessive hierarchische Verfeinerung entstehen) qualitativ und auch quantitativ genaue FEM-Lösungen in SP und sogar in HP zu berechnen.

IV) Ein direkter HFEM-Poisson-Löser

Ein Nebeneffekt der durch das „prehandling“ deutlich verringerten Konditionszahl der Steifigkeitsmatrix ist, dass iterative Methoden vom Typ Krylov-Raum-Verfahren (CG [10]) schon sehr effiziente Zugänge darstellen. Analysiert man allerdings die Struktur der entstehenden Matrizen genauer und berücksichtigt dabei die besonderen Eigenschaften der hierarchischen FEM, so lassen sich im Rahmen eines Schurkomplement-Ansatzes neben iterativen Varianten sogar effiziente direkte Lösungsverfahren konzipieren. Deren Aufwand lässt sich wie $\mathcal{O}(N^{3/2})$ charakterisieren, wobei N die Problemgröße darstellt und sich bei linearen bzw. bilinearen Finiten Elementen in 2D wie $\mathcal{O}(h^{-2})$ verhält. Die Grundlage dieses Verfahrens [11] ist eine spezielle Unterteilung der Knoten, die im direkten Zusammenhang mit der Gitterhierarchie steht. Konkret ergeben sich für den resultierenden Löser für das äquidistant diskretisierte Einheitsquadrat ca. $12N^{3/2}$ als notwendige Anzahl an arithmetischen Operationen (FLOPs).

Diese Anzahl an Operationen ist natürlich nicht optimal, da keine lineare Komplexität, also $\mathcal{O}(N)$, erreicht wird. Aber dafür beruht der zugehörige Algorithmus auf „dense“ Matrix-Vektor bzw. sogar Matrix-Matrix-Multiplikationen (im Falle von Problemen mit vielen rechten Seiten), die – wie oben gezeigt – auf der V100 so implementiert werden können, dass sie tatsächlich die Tensorcores ausnutzen, wie auch die nachfolgenden Studien illustrieren werden.

V) Numerische Resultate und Vergleiche mit Standard-FEM-Lösern auf Standard-Hardware

In den folgenden Grafiken in Abb. 5 zeigen wir exemplarisch für die drei Gitterweiten $h = 1/1024$, $1/512$ und $1/256$ (was im Zusammenhang mit (bi)linearen FEM für das Poisson-Problem auf dem äquidistant verfeinerten Einheitsquadrat zu $N \approx 1$ Million Unbekannte (für $h = 1/1024$) führt) die gemessenen GFlop/s Raten für die verschiedenen Genauigkeiten (also in HP, SP und DP). Zusätzlich geben wir in Abb. 6 auch die (durchschnittliche) Zahl an Millionen von Freiheitsgraden an, die in einer Sekunde behandelt (d.h. „gelöst“) werden können, um einen direkten Vergleich mit (geometrischen) FEM-Mehrgitterlösern zu erlauben. Dabei zeigen wir jeweils die Resultate für genau eine rechte Seite, d.h. wir lösen $Ax = b$ mit Lösungsvektor x und rechter Seite Vektor b der Dimension N , und für „viele“ (= M) rechte Seiten, d.h. $AX = B$ mit X und B als Matrix von Lösungen bzw. rechten Seiten der Dimension $N \times M$.

Gerade der Fall $M > 1$, d.h. wir lösen die Poisson-Probleme auf dem gleichen Rechengitter, aber gleichzeitig für M rechte Seiten, erlaubt zahlreiche Anwendungen von vollbesetzten Matrix-Matrix-Multiplikationen, die – wie vorher gezeigt – die höchsten Rechenleistungen erzielen können. Eine illustrierende Motivation, wo solche Probleme auftreten bzw. wie ebenfalls im Zuge von Hardware-orientierter Numerik entsprechende neue Algorithmen hergeleitet werden können, werden wir im abschließenden Kapitel nachreichen.

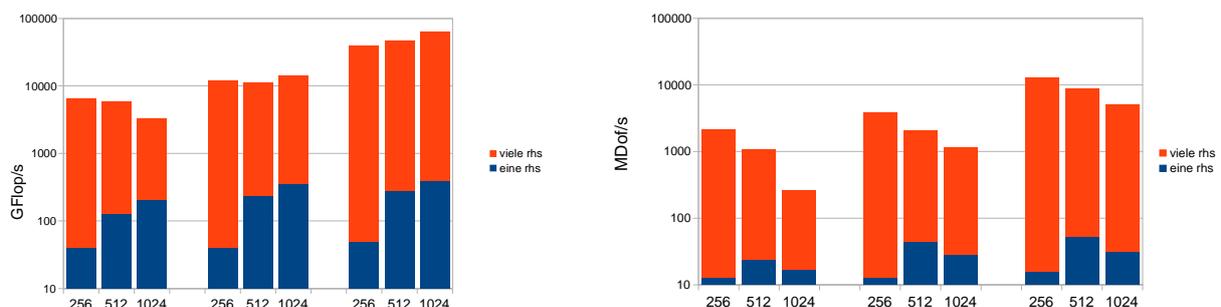


Abb.5: GFlop/s Raten (links) und „Anzahl Millionen von Freiheitsgrade/s“ (rechts) für die neue HFEM-basierte Methode für DP (jeweils links), SP (Mitte) und HP mit Tensorcores (rechts) auf der V100.

Tatsächlich belegen unsere Studien auf der V100, dass für „große“ Probleme, d.h. in unserem Fall für $h = 1/1024$ mit ca. 1 Million Unbekannten im Ort bei gleichzeitig vielen (= M) rechten Seiten, die Tensorcores immer besser ausgenutzt werden können, was letztendlich zu ca. 60 TFLOP/s „realer“ Rechenleistung führt. Und unsere Versuche, die internen V100 Strukturen zusammen mit speziellen NVIDIA Routinen besser auszunutzen, zeigen, dass wir damit deutlich über dem erwarteten Geschwindigkeitsgewinn beim Übergang von „single“ zu „half precision“ liegen, was bedeutet, dass wir mit diesem Zugang tatsächlich die Tensorcores ausnutzen können. Allerdings sind das bisher „nur“ TFLOP/s Raten, die nichts darüber aussagen, wie effizient dieser neue Zugang tatsächlich ist, insbesondere im direkten Vergleich zu einem Standard (geometrischen) FEM-Mehrgitterlöser, der auf den oben untersuchten „sparse“ MV-Anwendungen (natürlich in „double precision“ aufgrund der schlechten Konditionierung) beruht: Während ein typisches (optimiertes) Mehrgitterverfahren eine „optimale“ numerische Komplexität aufweist, d.h. $\mathcal{O}(N)$ arithmetische Operationen (FLOPs) zur Lösung benötigt, kostet der Lösungsvorgang mit der neuen vorgestellten Methode ca. $12N^{3/2}$ FLOPs, was nicht optimal ist für immer größer werdende Problemgrößen N .

Im konkreten Fall betrachten wir allerdings Probleme der Größe $N \approx 1$ Million, was für viele 2D-Problemstellungen schon eher zu den „größeren“ Problemen gehört; tatsächlich würde ein analoger Zugang in 3D bei der untersuchten Gitterweite $h = 1/1024$ zu ca. 1 Milliarde Unbekannten führen. Wenn man gleichzeitig berücksichtigt, dass für obige (optimierte) Mehrgitterverfahren eine Faustregel lautet, dass auf solchen gleichförmigen Gittern ca. $1.000N$ arithmetische Operationen ausreichen, um das zugehörige Problem (hinreichend genau) zu lösen, vergleichen wir hier die Faktoren 1.000 (Standard MG) und $12N^{1/2} \approx 12.000$ (HFEM): Der neue Zugang benötigt damit zwar ca. 12-mal mehr arithmetische Operationen, aber die Hoffnung ist – und dies wird auch durch die folgenden Ergebnisse belegt – dass durch die bessere Ausnutzung der Hardwareleistung dennoch insgesamt eine deutlich höhere Effizienz, d.h. deutlich geringere Rechenzeiten, erreicht wird.

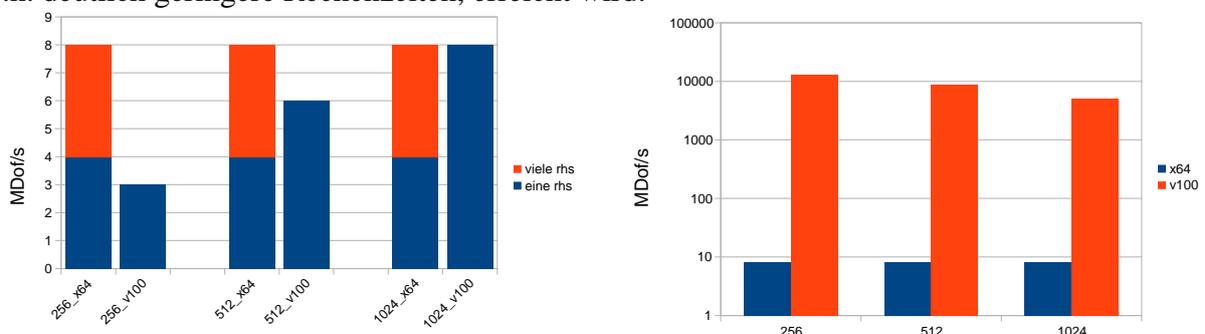


Abb.6: „Anzahl Millionen von Freiheitsgrade/s“ (links) für den FEM-MG-Zugang in DP und Vergleich dieses „Standardzugangs“ (FEM, MG, DP, x64, viele rechte Seiten) mit der neuen Methode (HFEM, direkt, HP, V100).

Konkret zeigen diese Studien, dass der neue Zugang im Falle von einer einzigen rechten Seite, also $Ax = b$, im Vergleich zu unserem (in FEAT3 [12] bereits hochoptimierten) Standard-Mehrgitter-Löser auf der V100 bis zu viermal und auf der AMD CPU bis zu achtmal schneller sein kann (man beachte: Der neue Löser für HFEM in HP wird mit dem FEM-Zugang in DP verglichen). Und im Falle von vielen rechten Seiten erreichen wir weitere deutliche Effizienzsteigerungen auf der V100, da wir sowohl „lower precision“ als auch die Tensorcores ausnutzen können. Und um es auf die Spitze zu treiben: Der neue Zugang ist auf einer V100 Grafikkarte bei vielen rechten Seiten – und bei nahezu gleicher Genauigkeit – um den Faktor 600 schneller als unser bisheriges „Arbeitspferd“, ein geometrischer FEM-Mehrgitterlöser in „double precision“ und CSR-Speichertechnik auf einer (multicore) AMD EPYC 7542 CPU!

Natürlich muss man an dieser Stelle kritisch anmerken, dass diese Studien nur für ein äquidistant verfeinertes Einheitsquadrat durchgeführt wurden, und das mit einem „black box“ FEM-Mehrgitterzugang für allgemeine, d.h. insbesondere semi-strukturierte Grobgitter, die hierarchisch verfeinert werden. Der neu vorgestellte Zugang kann aber ebenfalls auf solchen Rechengittern verwendet werden, bei denen das „Grogitter“ unstrukturiert ist, weitere Verfeinerungslevel aber durch uniforme Verfeinerungen erhalten werden. Ein typisches Beispiel (z.B. für eine „flow around a square“ Konfiguration) ist in der nachfolgenden Abb. 7 zu finden, für die wir ebenfalls im Rahmen dieser „proof-of-concept“ Studien mit linearen FEM den beschriebenen Zugang analysieren konnten.

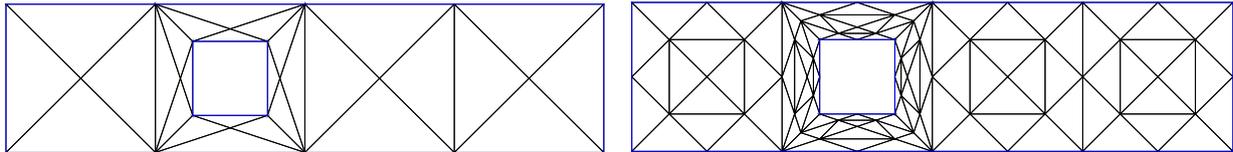


Abb. 7: Darstellung des größten Gitters einer „flow around a square“ Konfiguration und des Gitters nach einem uniformen Verfeinerungsschritt.

Ein weiterer Vorteil des neuen Zugangs, zumindest der beschriebenen Variante eines „direkten“ Löser, besteht darin, dass das Löserverhalten mehr oder weniger unabhängig von sog. „aspect ratios“, d.h. Gitterverzerrungen, Elementgrößensprüngen, etc. ist. Im Gegensatz dazu reagieren die verwendeten (iterativen) geometrischen Mehrgitterlöser bzgl. ihres Konvergenzverhaltens tatsächlich sehr sensitiv (und zwar deutlich schlechter!) auf solche Gitterstrukturen.

VI) Zusammenfassung und Ausblick

Ziel dieser aktuellen Studie (Stand: November 2020) war es, exemplarisch zu demonstrieren, wie mit dem Wissen über moderne „Spezialhardware“ neue (Hardware-orientierte) numerische Techniken und Algorithmen entwickelt werden können. Diese sind in der Lage, eine hohe numerische Effizienz mit einer gleichzeitig sehr hohen „computational“ Performance auf eben solchen Computersystemen, hier insbesondere Beschleunigerhardware vom Typ NVIDIA Volta V100, zu verknüpfen. Wird zusätzlich noch Parallelität bzgl. vieler solcher Komponenten ausgenutzt, können für entsprechende Anwendungen in der numerischen Kontinuumsmechanik, die solche „Basisbausteine“ verwenden, die Simulationszeiten um mehrere Größenordnungen beschleunigt werden.

Natürlich sind das „nur“ erste (Vor-)Studien für Poisson-Probleme; diese stellen aber beispielsweise einen der zentralen Bausteine innerhalb von zeitabhängigen Strömungsproblemen auf der Basis der inkompressiblen Navier-Stokes Gleichungen dar: Im Rahmen von (diskreten) Projektionsmethoden als typische Operator-Splitting-Zugänge muss in jedem Zeitschritt ein Poisson-Problem für den Druck gelöst werden, bei dem tatsächlich oft nur die rechte Seite von Zeitschritt zu Zeitschritt variiert, wenn das (Orts-)Rechengitter gleich bleibt. Im Zusammenspiel mit „zeitsimultanen“ Zugängen [13] untersuchen wir gerade Varianten der obigen Projektionsmethoden, die es erlauben, **alle** Druck-Poisson-Probleme, d.h. für alle Zeitschritte gleichzeitig, auf einmal zu lösen: Dies entspricht gerade dem beschriebenen Fall von $AX = B$, d.h. gleiche Steifigkeitsmatrix A , aber für eine Matrix an rechten Seiten B und entsprechenden Lösungsvektoren X , für die die beschriebenen sehr hohen Rechenleistungen erreicht werden konnten.

Und natürlich gibt es noch weiteren Forschungsbedarf hinsichtlich „prehandling“ in 3D und für andere Differentialoperatoren bzw. FEM-Ansätze, aber auch bzgl. iterativer Varianten des

HFEM-Schurkomplement Ansatzes, insbesondere auf anderer Beschleunigerhardware: Unsere vorgestellten Studien zu den tatsächlich möglichen Effizienzsteigerungen von mathematischen Simulationswerkzeugen, gerade in der numerischen Kontinuumsmechanik, erscheinen noch relevanter angesichts der Tatsache, dass das Rechnersystem „JUWELS“ in Jülich [14] mit dem neuen Boostermodul (basierend auf einem Nachfolgermodell der betrachteten V100 Grafikkarten) in der neuesten Top500-Liste auf Platz 7 steht und somit wohl der derzeit schnellste europäische Rechner ist. Das entsprechende Nachfolgermodell, NVIDIA Ampere A100, bietet Tensorcores in „half precision“ mit mehr als 300 TFLOP/s: Zusammen mit „prehandling“ Techniken wird es daher möglich sein, hocheffiziente FEM-Simulationen von PDEs auf moderner Beschleunigerhardware auszuführen, die das Peak Performance Potential der Hardware (nahezu) voll ausnutzen.

VII) Literatur

- [1] V100 SXM2 <https://images.nvidia.com/content/technologies/volta/pdf/volta-v100-datasheet-update-us-1165301-r5.pdf>
- [2] half precision https://en.wikipedia.org/wiki/Half-precision_floating-point_format
- [3] <https://blogs.nvidia.com/blog/2019/11/15/whats-the-difference-between-single-double-multi-and-mixed-precision-computing/>
- [4] AMD Epyc 7542, 32 Kerne <https://www.amd.com/en/products/cpu/amd-epyc-7452>
- [5] Saad, Y.: Iterative Methods for Sparse Linear Systems; <https://doi.org/10.1137/1.9780898718003>
- [6] Tensorcores <https://www.nvidia.com/de-de/data-center/tensor-cores/>
- [7] Ruda, D., Turek, S., Zajac, P. & Ribbrock, D. (2019). The Concept of Prehandling as Direct Preconditioning for Poisson-like Problems. Ergebnisberichte des Instituts für Angewandte Mathematik, No.620, Dezember 2019, <http://dx.doi.org/10.17877/DE290R-20384>
- [8] Ruda, D. (2020). Numerische Studien zur „Vorbehandlung“ (prehandling) von Poisson-artigen Problemen durch die hierarchische Finite-Elemente-Methode. Masterarbeit, TU Dortmund
- [9] Yserentant, H. (1986). On the Multi-Level Splitting of Finite Element Spaces. Numerische Mathematik, Vol. 49, S. 379-412, <https://doi.org/10.1007/BF01389538>
- [10] Meister, A.: Numerik linearer Gleichungssysteme; <https://doi.org/10.1007/978-3-658-07200-1>
- [11] Ruda, D., Turek, S., Ribbrock, D. Zajac, P. & Poelstra, H. (2021). Very fast FEM Poisson solvers on lower-precision accelerator hardware: A „proof-of-concept“ study for NVIDIA Volta V100 (in Vorbereitung)
- [12] Ruelmann, H.; Geveler, M.; Ribbrock, D.; Zajac, P. & Turek, S. (2019) Basic Machine Learning Approaches for the Acceleration of PDE Simulations and Realization in the FEAT3 Software, Ergebnisberichte des Instituts für Angewandte Mathematik Nummer 618, Fakultät für Mathematik, TU Dortmund, 618
- [13] Dünnebacke, J.; Turek, S.; Lohmann, Chr.; Sokolov, A. & Zajac, P. (2020). Increased space-parallelism via time-simultaneous Newton-multigrid methods for nonstationary nonlinear PDE problems (Special Issue PACO 2019, eingereicht)
- [14] <https://www.fz-juelich.de/SharedDocs/Pressemitteilungen/UK/EN/2020/2020-11-16-juwels-booster.html?nn=362372>