

Bitgeometrie im Hydro-Code: groessere lokale Versuchsserie

Pascal Bauer

Planung, Implementierung und Auswertung mit ChatGPT

18. September 2026

1 Fragestellung und Abgrenzung

Beschleunigen feste Normalenrichtungen und der Zweibit-Decoder den gesamten Hydro-Gitteraufbau? Spart eine spezialisierte Richtungsprojektion auch im vollstaendigen HLLC-Fluss Zeit? Die Anregung stammt von Pascal Bauer. Die konkrete Versuchsanordnung und Umsetzung stammen von ChatGPT.

Die Referenz speichert Normalen bereits einmal beim Aufbau. Untersucht wird daher nicht das Vermeiden einer Normalenberechnung in jedem Zeitschritt. Die neuen Pfade sind experimentelle C++-Optionen; der Standard bleibt die Referenz. Die Arbeit Version 0.3 bleibt unveraendert.

2 Versuchsplan und Kontrollen

18 adaptive, vollstaendige, konforme Gitter aus 2000, 16000 und 64000 Stuetzpunkten: drei Cluster oder ein dichter Hotspot mit Hintergrund, je drei Seeds (807–809). Die tatsaechlichen Gitter umfassen 7062 bis 217472 Zellen; maximale Blatttiefen je Fall: 24 bis 45. Die Stuetzpunkte erzeugen nur das Gitter, keine Teilchengravitation.

Drei Aufbaupfade: (0) bisherige Mittelpunktrekursion und Kantennormalisierung, (1) dieselben Ecken, aber Orientierung einmal pro Zelle und feste Normalen/Laengen, (2) zusaetzlich Ecken aus dem exakten Zweibit-Decoder. Vollstaendiger Aufbau inklusive Validierung, Nachbarn, Ecksternen und Rekonstruktionsgewichten. Ecken und Kantengeometrie sind Unterzeiten, nicht zum Gesamtwert zu addieren.

Auf jedem Gitter werden vollstaendige Euler-Operatoren zweiter Ordnung ausgewertet: Rekonstruktion, HLLC mit unveraendertem Rusanov-Fallback und konservatives Gather. Je glatter Zustand und schraeger Sod-artiger Sprung, ein und vier Threads. Normale und spezialisierte Projektion werden bei identischer Decoder-Geometrie verglichen. Keine Sortierung der Kanten, keine Gravitation.

Je Kombination fuef Zeitwiederholungen nach Aufwaermen, Varianten und Faelle in deterministisch gemischter Reihenfolge. Kontrollrechnungen und Ein-/Ausgabe ausserhalb der Zeitmessung. Drei Seeds sind die Replikate; Zeitwiederholungen sind keine unabhaengigen physikalischen Versuche. Diagramme zeigen Mediane der fallweisen Mediane und die volle Spannweite der drei Seeds.

2.1 Kontrollergebnisse

Alle 2070 Zeitmessungen sind vorhanden. Maximale komponentenweise Operatorabweichung $|a - b|/(1 + |a| + |b|)$: 3.680×10^{-9} . Maximales normiertes Residuum der integralen Bilanz $|\sum_K A_K \dot{U}_K + F_{\partial\Omega}|/(1 + |F_{\partial\Omega}|)$: 3.750×10^{-15} . CFL-Grenzen und Fallbackzahlen wurden ebenfalls verglichen. Geometrische Ecken, Flaechen und Zentren sind bitgleich.

Zusaetzlich: 16000 HLLC-Paare ueber alle acht Richtungen, einschliesslich kalter auseinanderlaufender Stroemungen und Fallbacks; Vergleich mit allgemeinem Fluss und Gegenrichtung. Ein kurzer zeitabhaengeriger AMR-Test prueft Endzustaende und Erhaltung. Die grosse Messserie selbst wiederholt eingefrorene Zustaende: kein Langzeitstabilitaets- oder Schockkonvergenznachweis.

3 Gitteraufbau

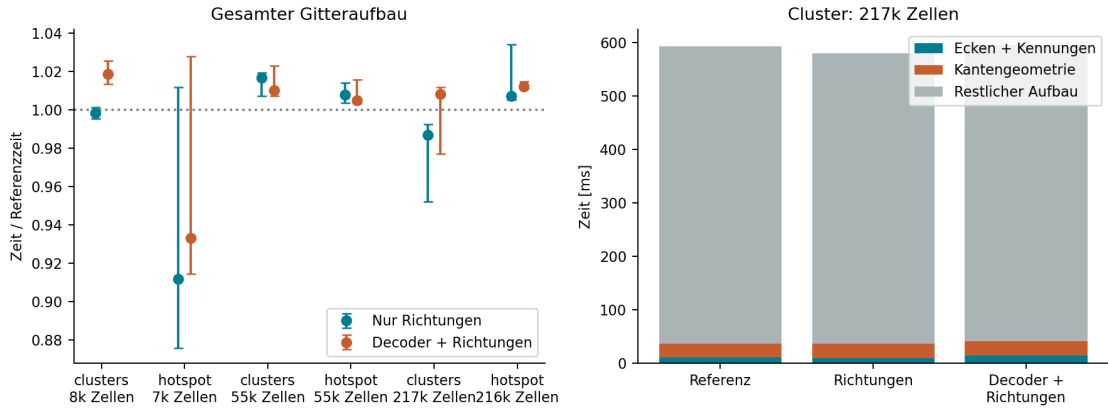


Abbildung 1: Links: Aufbauzeit relativ zur Referenz (kleiner ist besser). Rechts: Aufteilung fuer die groesste Clustergruppe. Unterzeiten sind Teil der Gesamtzeit.

Verteilung	Zellen (Median)	Referenz [ms]	Richt./Ref.	Decoder/Ref.
clusters	7598	14.9	0.998	1.019
hotspot	7279	17.4	0.912	0.933
clusters	55141	128.1	1.017	1.010
hotspot	55005	128.9	1.008	1.005
clusters	217018	593.0	0.987	1.008
hotspot	215727	620.5	1.007	1.012

Bei den beiden groessten Gruppen liegen die Aufbauquotienten zwischen 0.987 und 1.012, also nahe eins. Der groesste scheinbare Gewinn tritt nur im kleinen Hotspot-Fall auf und setzt sich bei groesseren Gittern nicht fort. Rund 94% des Referenzaufbaus der grossen Clustergruppe liegen ausserhalb der gemessenen Ecken- und Kantenphase. Das begrenzt den Gesamtnutzen dieser lokalen Optimierung stark.

Die Groesse eines Vorteils ist am gesamten Aufbau zu beurteilen. Schnellere Formeln fuer Ecken koennen wenig bewirken, wenn Topologie und Rekonstruktionsstencils dominieren. Fuer konstante Gitter wird der Aufbau zudem ueber viele Zeitschritte amortisiert. Diese Messung umfasst nicht den vorgelagerten Aufbau des konformen Baumes aus Stuetzpunkten.

4 Vollstaendiger Euler-Operator

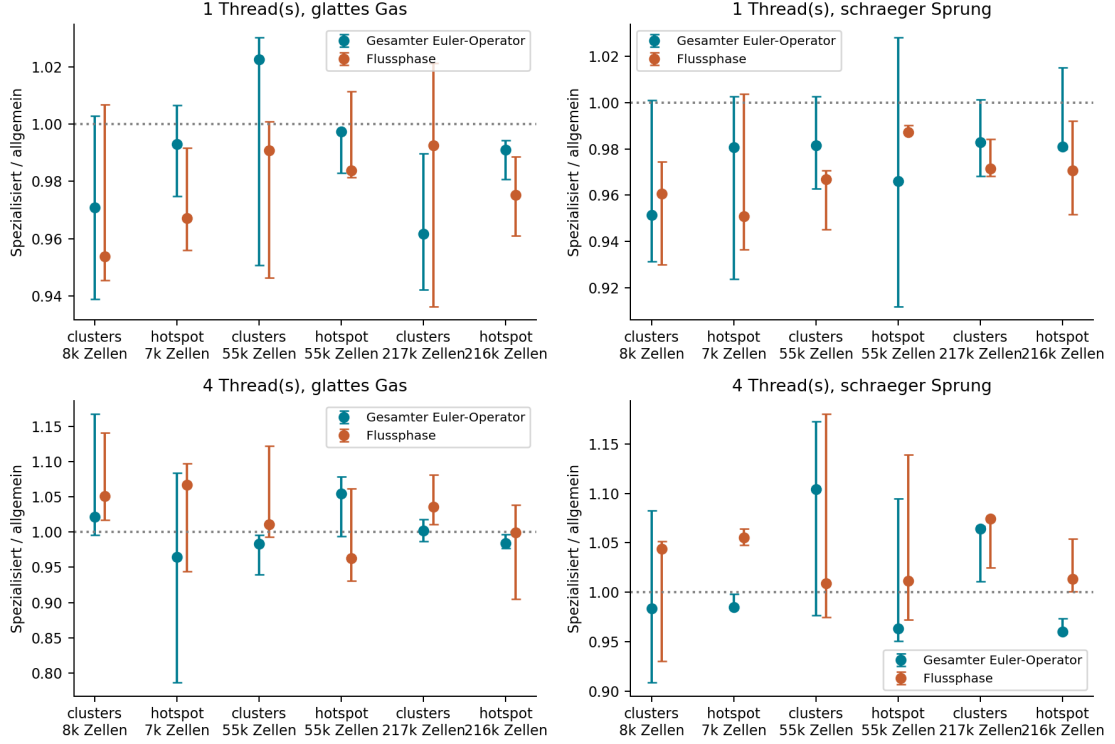


Abbildung 2: Spezialisierte Projektion gegen allgemeines Skalarprodukt auf derselben Bitdecoder-Geometrie. Werte unter eins sind guenstig. Die Flussphase enthaelt auch Seitenrekonstruktion und Allokationen, ist kein isolierter Riemann-Kernel.

Eine Fallunterscheidung spart arithmetische Operationen, kann aber Verzweigungen und andere Optimierungen beeinflussen. Aus diesen Wandzeiten allein laesst sich keine Mikroarchitektur-Ursache beweisen. Insbesondere ist ein kleiner Unterschied innerhalb der Seed-Spannweite kein belastbarer Gewinn.

Im Ein-Thread-Fall ist die spezialisierte Flussphase in allen sechs Gruppen und beiden Gaszuständen im Median etwas schneller: Quotienten etwa 0.951 bis 0.993. Das ist wesentlich weniger dramatisch als der isolierte Projektions-Mikrobenchmark vermuten liess und zeigt, weshalb der komplette Operator gemessen werden muss. Mit vier Threads ist das Bild gemischt (Flussquotienten etwa 0.963 bis 1.074); die Streuung ist teilweise deutlich groesser als der Effekt.

5 Einordnung und Reproduzierbarkeit

Die Richtungsvariante speichert ein zusätzliches Byte je Kante in einem separaten Vektor; dessen Kapazitätsreserve kann zusätzlichen Speicher benötigen. Die Referenz allokiert diesen Vektor nicht. Ein vorgelagerter Erkundungslauf mit grösserem Kantendatensatz liegt separat unter `results` und geht nicht in diese Auswertung ein.

Alle sieben Release-Regressionstests sowie beide Hydrotests unter `AddressSanitizer` und `UndefinedBehaviorSanitizer` bestanden. Die Leak-Erkennung war dabei deaktiviert; daraus folgt kein gesonderter Nachweis der Speicherleckfreiheit.

Entscheidung: keine Standardumstellung. Die Bitgeometrie ist funktional brauchbar, doch der grössere Versuch zeigt keinen durchgängigen Gewinn im Gesamtaufbau. Für die Richtungsprojektion gibt es einen kleinen positiven Hinweis bei einem Thread, aber keinen stabilen Vorteil über alle getesteten Betriebsarten. Die experimentellen Optionen bleiben für gezielte Folgeprüfungen erhalten.

Nächste technische Priorität: den restlichen Gitteraufbau feiner messen: Mengenaufbau und Validierung, Nachbarsuche, Ecksterne sowie Least-Squares-Gewichte. Erst eine solche Trennung zeigt, ob Wiederverwendung unveränderter Bereiche, kompaktere Datenstrukturen oder Parallelisierung den grösseren Hebel bieten. Das ist ein Vorschlag aus den Messdaten, noch kein bereits bewiesener Optimierungserfolg.

Lokaler Rechner: Intel Core Ultra 7 255U, GCC 13.3.0, Release (-O3, kein fast-math, FP-Kontraktion aus). OpenMP: dynamische Teams aus, spread/core-Platzierung, PASSIVE. Dauer der finalen Serie: 166.6 Sekunden. Keine gleichzeitigen Benchmarkprozesse. Keine feste Taktfrequenz, keine Temperaturkontrolle, hybride Kerne und Hintergrundlast möglich. Ein-/Vier-Thread-Zeiten sind hier keine umfassende Skalierungsstudie.

Vor einer Standardumstellung wären ein reproduzierbarer Gesamtnutzen auf weiteren Rechnern und vollständige zeitabhängige Hydrovergleiche erforderlich. Vielversprechende kleine Teilzeiten allein rechtfertigen weder acht getrennte Solver noch eine allgemeine Beschleunigungsbehauptung. Sinnvolle nächste Optimierungsziele ergeben sich aus dem dominierenden Rest des Gitteraufbaus.

Rohdaten und atomarer Laufstatus stehen unter `results_final`; `summary.json` enthält Quotienten je Seed. Quellarchiv, Binaerdatei, Compilerflags, Tests und SHA-256-Prüfsummen werden neben dem Bericht gesichert. Der Bericht verwendet die bereitgestellte LaTeX-Vorlage.