

Untersuchungen von FDS-Drucklösern auf unstrukturierten Gittern

Dr. Susanne Kilian

hhpberlin - Ingenieure für Brandschutz

10245 Berlin

Agenda

1

Diskretisierung
Poisson-Gleichung

2

Poisson-Löser

3

Testreihen

4

Fazit

Diskretisierung der Poisson-Gleichung

Strukturierte versus unstrukturierte kartesische Gitter

Druckgleichung in FDS

Elliptische partielle Differentialgleichung Typ „Poisson“

$$\nabla^2 \mathcal{H} = -\frac{\partial(\nabla \cdot \mathbf{u})}{\partial t} - \nabla \cdot \mathbf{F} \quad + \quad \text{Randwerte}$$

Quellterme aus dem vorangehenden Zeitschritt
(Strahlungstransport, Verbrennung, etc.)

- eng gekoppelt mit dem Strömungsfeld
- mindestens 2x pro Zeitschritt zu lösen

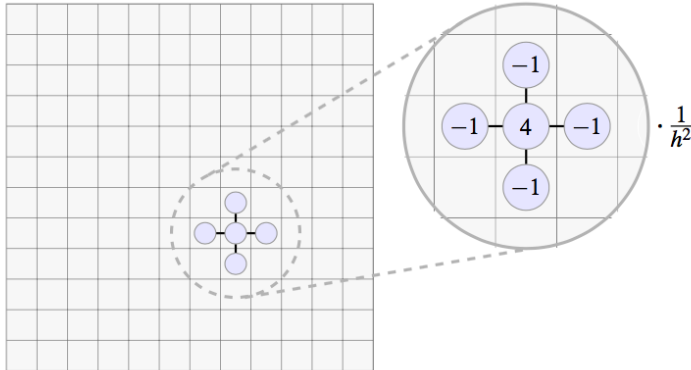
Finite Differenzen Diskretisierung

1

Diskretisierungstern in 2D:

Diskretisierung
Poisson-Gleichung

$$\frac{1}{h^2}(-\mathcal{H}_{i,k-1} - \mathcal{H}_{i-1,k} + 4\mathcal{H}_{i,k} - \mathcal{H}_{i,k+1} - \mathcal{H}_{i+1,k}) = R_{i,k}$$



- zell-zentriert
- spezifiziert die physikalischen Relationen zwischen den Zellen
- führt auf dünn-besetzte Matrix (nur wenige Nicht-Null-Einträge)

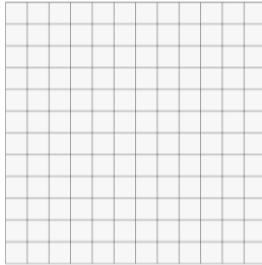
Finite Differenzen Diskretisierung

1

Single-Mesh:

1 globales Gleichungssystem

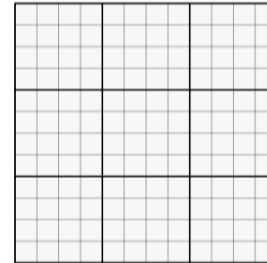
$$Ax = b$$



Multi-Mesh:

M lokale Gleichungssysteme

$$A_m x_m = b_m$$

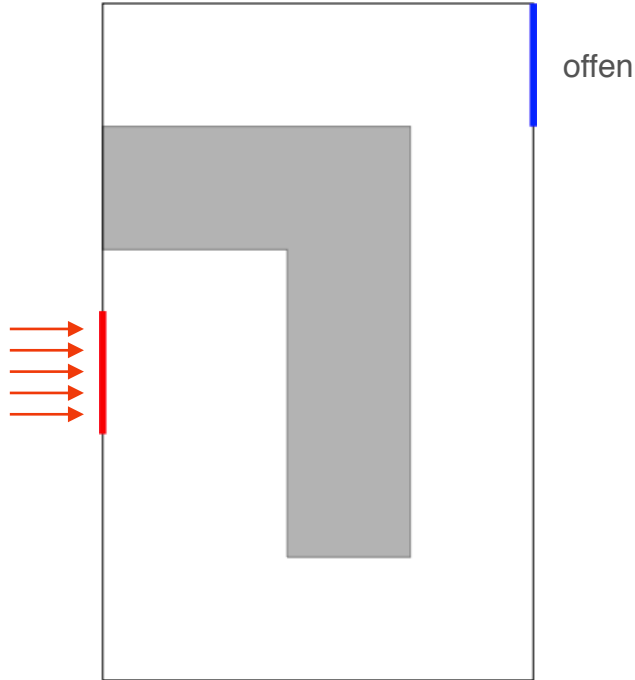


Diskretisierung
Poisson-Gleichung

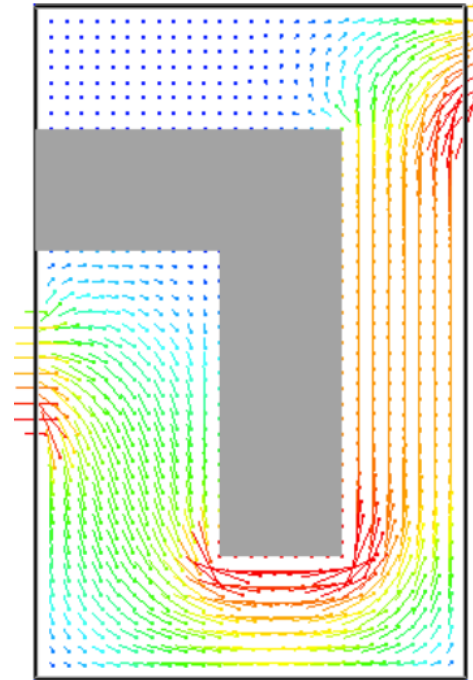
Behandlung interner Objekte

1

Einfaches 2D-Beispiel



Zugehöriges Geschwindigkeitsfeld

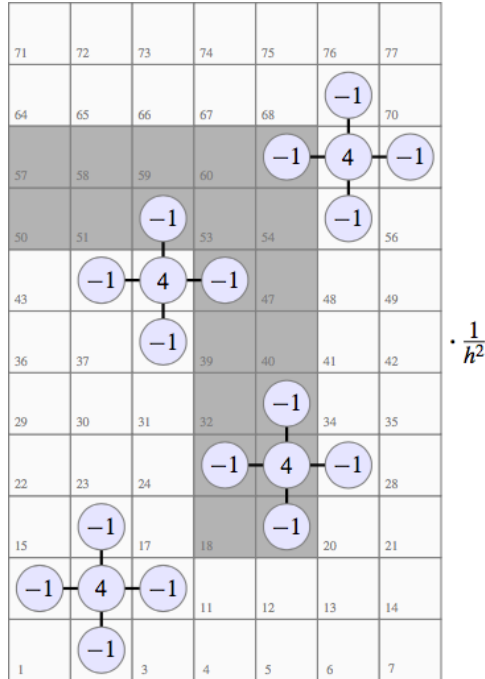


Diskretisierung
Poisson-Gleichung

Strukturierte kartesische Gitter

1

Diskretisierung
Poisson-Gleichung



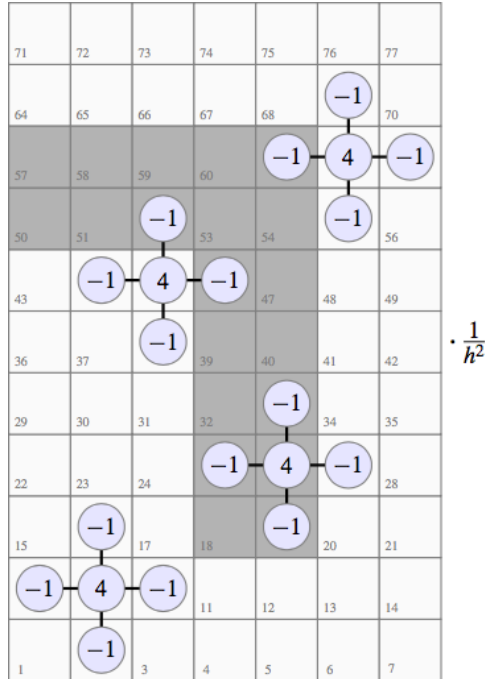
„Gasphase“ und „Solid“-Zellen:

- gleichförmige Matrixsterne unabhängig von inneren Objekten
- Zellen innerhalb der Objekte werden ins Gleichungssystem übernommen

Strukturierte kartesische Gitter

1

Diskretisierung
Poisson-Gleichung



Vorteil:

- sehr reguläre Matrixstruktur (einheitliche Nachbarschaftsverhältnisse zwischen Zellen)
- kann im Lösungsprozess effizient ausgenutzt werden (Beispiel FFT)

Einsatz hoch-optimierter Löser möglich

| | | | | | | |
|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| 71 | 72 | 73 | 74 | 75 | 76 | 77 |
| 64 | 65 | 66 | 67 | 68 | <div>-1</div> | 70 |
| 57 | 58 | 59 | 60 | <div>-1</div> | <div>4</div> | <div>-1</div> |
| 50 | 51 | <div>-1</div> | 53 | 54 | <div>-1</div> | 56 |
| 43 | <div>-1</div> | <div>4</div> | <div>-1</div> | 47 | 48 | 49 |
| 36 | 37 | <div>-1</div> | 39 | 40 | 41 | 42 |
| 29 | 30 | 31 | 32 | <div>-1</div> | 34 | 35 |
| 22 | 23 | 24 | <div>-1</div> | <div>4</div> | <div>-1</div> | 28 |
| 15 | <div>-1</div> | 17 | 18 | <div>-1</div> | 20 | 21 |
| <div>-1</div> | <div>4</div> | <div>-1</div> | 11 | 12 | 13 | 14 |
| <div>-1</div> | | 3 | 4 | 5 | 6 | 7 |

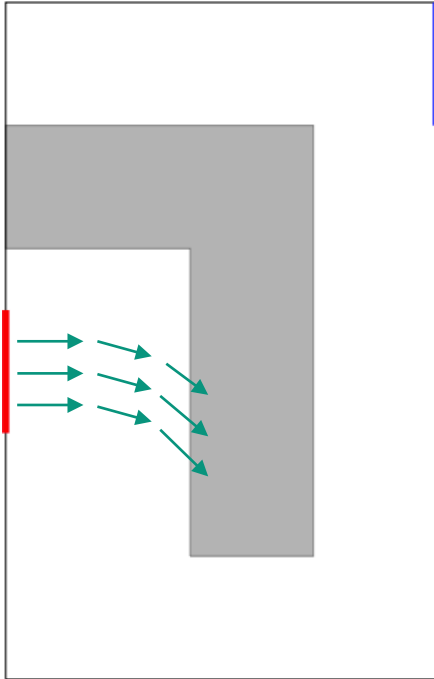
- inkorrekte Behandlung innerer Randwerte
- mögliche Verletzung der Erhaltungseigenschaften (Penetration in innere „Solids“)
- iterative Korrektur an inneren Rändern nötig

Effizienz-/Genauigkeitsverluste

Strukturierte kartesische Gitter

1

Diskretisierung
Poisson-Gleichung



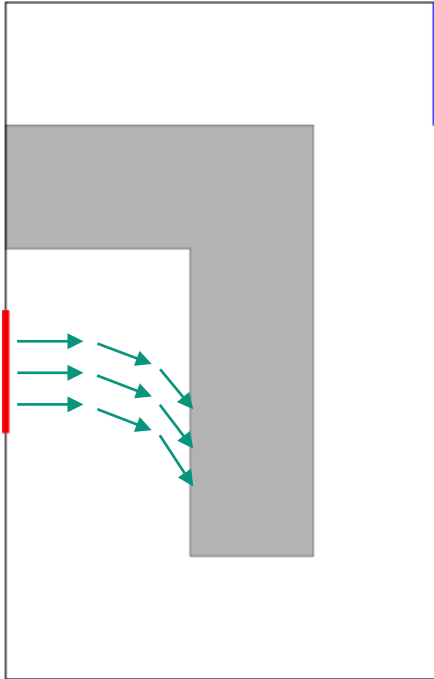
FFT(tol): „Direct Forcing Method“

- „Eindringen“ in innere Objekte soll verhindert werden

Strukturierte kartesische Gitter

1

Diskretisierung
Poisson-Gleichung



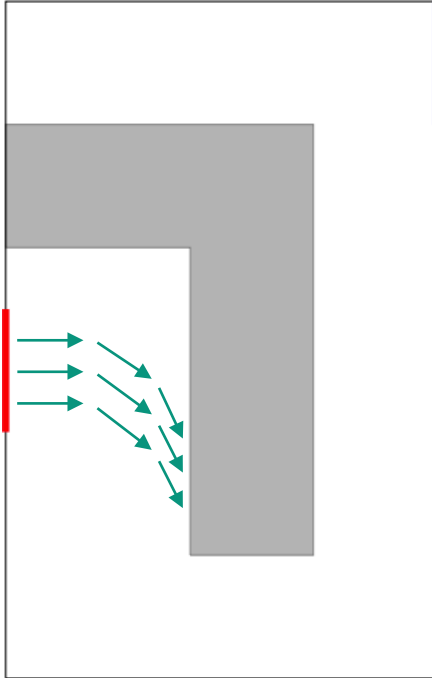
FFT(tol): „Direct Forcing Method“

- „Eindringen“ in innere Objekte soll verhindert werden
- Normalableitung der Geschwindigkeit muss gegen Null gehen

Strukturierte kartesische Gitter

1

Diskretisierung
Poisson-Gleichung



FFT(tol): „Direct Forcing Method“

- „Eindringen“ in innere Objekte soll verhindert werden
- Normalableitung der Geschwindigkeit muss gegen Null gehen
- es wird so lange iteriert bis vorgegebene Toleranz **tol** unterschritten ist

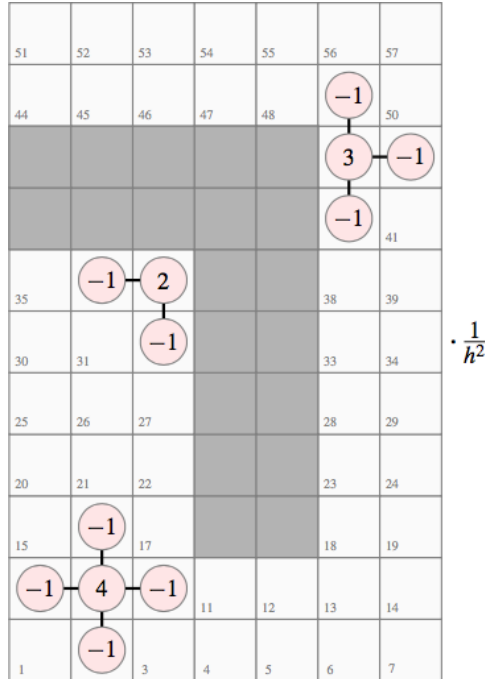
Eingabe in FDS-Geometrie, z.B.:

&PRES VELOCITY_TOLERANCE = 0.001

Unstrukturierte kartesische Gitter

1

Diskretisierung
Poisson-Gleichung



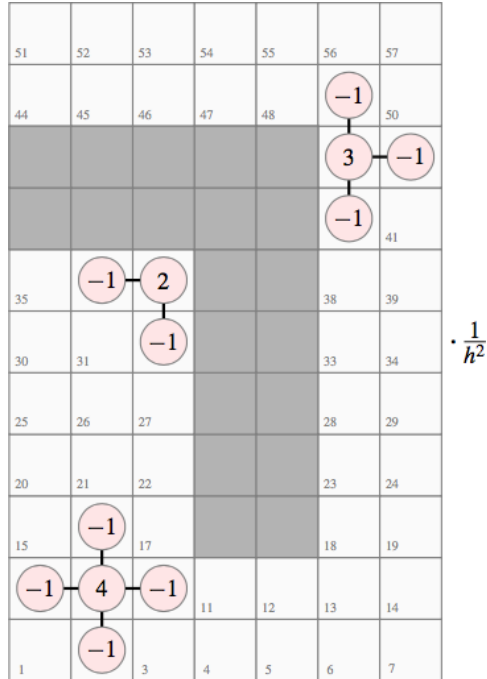
Nur „Gasphase“-Zellen:

- unterschiedliche Matrixsterne unter Aussparung innerer Objekten
- Zellen innerhalb der Objekte werden **nicht** ins Gleichungssystem übernommen

Unstrukturierte kartesische Gitter

1

Diskretisierung
Poisson-Gleichung



Vorteil:

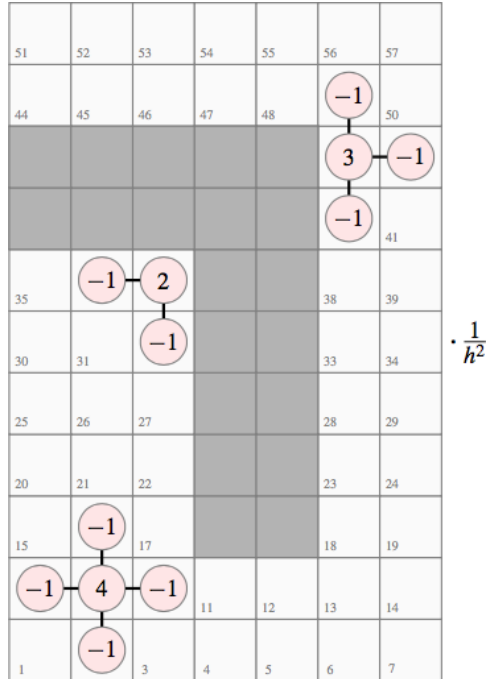
- Korrekte Setzung der Randwerte
(**homogene Neumann-Bedingung**)
- geringere Anzahl an Unbekannten

Höhere Genauigkeit,
keine zusätzliche Korrektur

Unstrukturierte kartesische Gitter

1

Diskretisierung
Poisson-Gleichung



Nachteil:

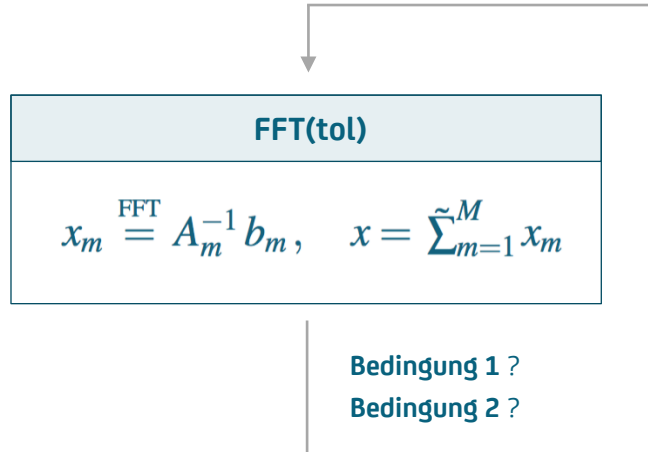
- Verlust der regulären Matrixstruktur
- keine spezialisierten Löser mehr anwendbar (Beispiel: FFT)

Einsatz optimierter Löser schwieriger

Lösung der Poisson-Gleichung

Präsentation verschiedener Lösungsansätze

Fast Fourier Transformation: FFT(tol) mit Geschwindigkeits-Korrektur



Bedingung 1: Innere Objekte

Geschwindigkeits-Normalkomponenten < **tol**

Bedingung 2: Innere Meshgrenzen

Differenz Geschwindigkeits-Normalkomponenten < **tol**

- ein einzelner FFT-Durchlauf ist hoch-effizient und robust
- aber nur für strukturierte Gitter anwendbar !

Parallele LU-Zerlegung:

MKL Interface Intel MKL Pardiso

MKL - Initialisierung

$$LU = \sum_{m=1}^M A_m$$

Initialisierung

- zunächst „Reordering“ der Matrixstruktur
- verteilte Berechnung der globalen LU-Zerlegung

MKL - Poisson Lösung

$$Ly = b, \quad Ux = y$$

Poisson-Lösung pro Zeitschritt:

- Einfache Lösung des Poisson-Problems durch lokales Vorwärts-/Rückwärts-Einsetzen

- ebenfalls als sehr effizient und robust angepriesen
- für strukturierte und unstrukturierte Gitter anwendbar

ScaRC-Varianten: Block-CG und -GMG Verfahren

ScaRC-CG / ScaRC-GMG

Vorkonditionierung/Glättung:

Block-SSOR, Block-MKL

Grobgrid-Lösung:

CG, MKL

Konjugierte Gradientenverfahren (CG):

- Lösung eines äquivalenten Minimierungs-Problems

Geometrisches Mehrgitterverfahren (GMG):

- Verwendung einer kompletten Gitterhierarchie mit exakter Lösung auf größtem Gitter

Blockweise Durchführung mit 1 Zelle Überlapp

- meist gute Konvergenz, aber abhängig von Verfahrensparametern
- für strukturierte und inzwischen auch für unstrukturierte Gitter anwendbar

Numerische Testreihen

Vergleich für verschiedene Geometrien

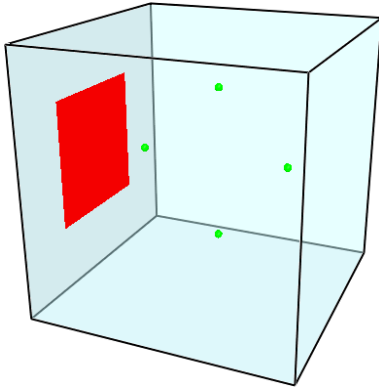
Basis-Geometrien

3

Testreihen

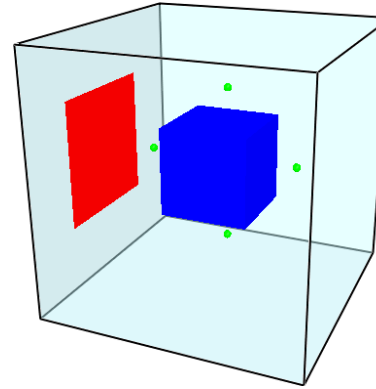
Cube⁻

Würfel ohne Hindernis



Cube⁺

Würfel mit Hindernis



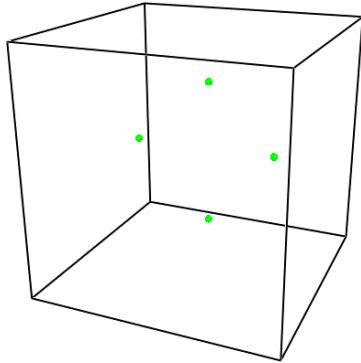
Gitterweiten:
 $24^3, 48^3, 96^3$
für gesamten Würfel

- zyklische Einströmung von links via Ramp, offene Ausströmung rechts
- Vergleich FFT(tol) auf strukturiertem Gitter versus MKL und ScaRC auf unstrukturiertem Gitter

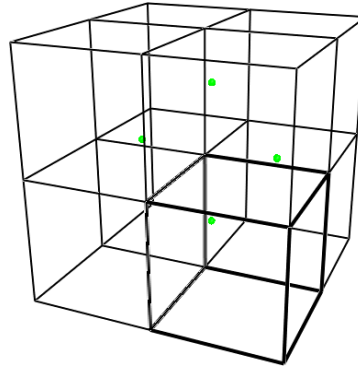
Mesh-Zerlegungen

3

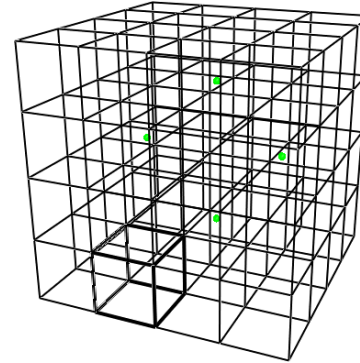
1-Mesh



8-Mesh



64-Mesh



Testreihen

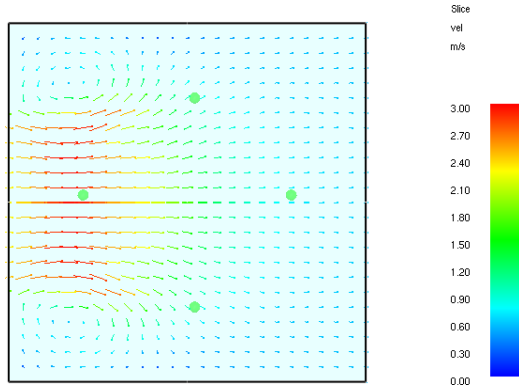
- Notationen: **Cube⁻(M)** und **Cube⁺(M)** für jeweilige M-Mesh-Zerlegung
- Anzahl an Zellen pro Mesh wird weniger mit wachsender Anzahl an Meshes

Genauigkeits-Check

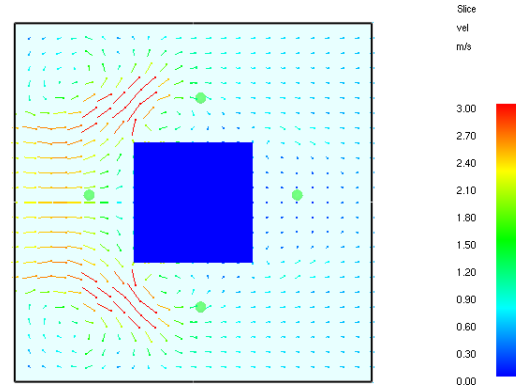
3

Testreihen

Cube⁻



Cube⁺



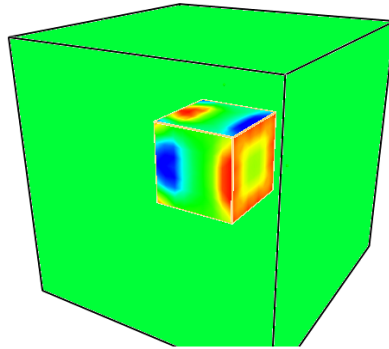
- Messung des Volumenstrom-Differenzen „Einströmung versus Ausströmung“
- Messung der Druckverläufe an gekennzeichneten Messstellen

Cube⁺(1): Penetration des Hindernisses

Testreihen

FFT(10⁻²)

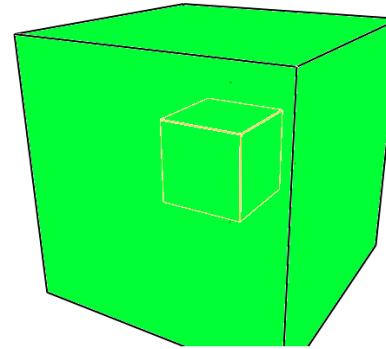
24³ Zellen



Time: 0.516

FFT(10⁻⁶)

24³ Zellen



Time: 0.516

& BNDF QUANTITY='VELOCITY ERROR' /

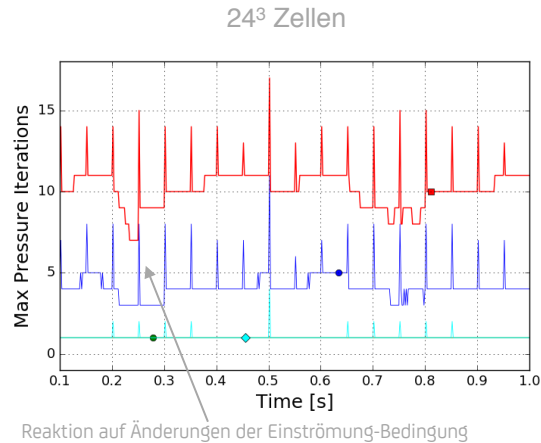
- Geschwindigkeits-Korrektur reduziert den Fehler entlang des Hindernisses (beiden Fällen liegt gleiche Skala zugrunde)

Cube⁺(1): FFT(tol), tol = 10^{-2} , 10^{-4} , 10^{-6}

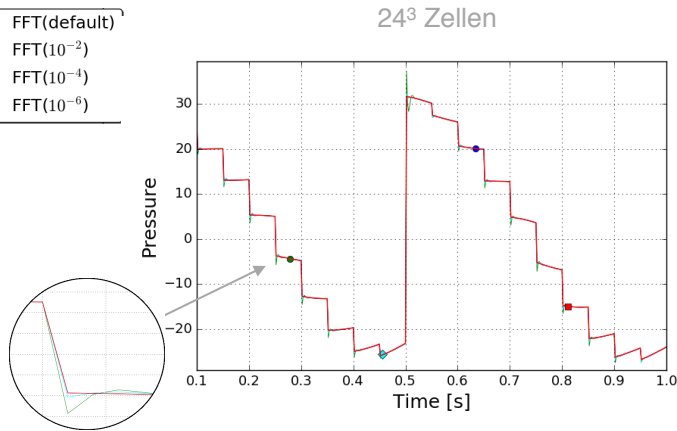
3

Testreihen

Anzahl Poisson-Lösungen pro Zeitschritt



Geschwindigkeits-Fehler entlang Hindernis



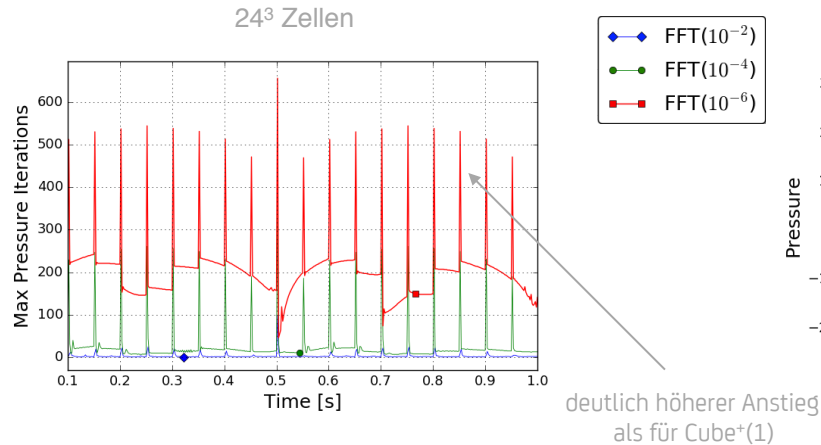
- mehr Poisson-Lösungen pro Zeitschritt erforderlich (~ 4 für tol= 10^{-4} , ~ 10 für tol= 10^{-6})
- zum Vergleich: nur 1 Poisson-Lösung pro Zeitschritt für **Cube⁻(1)**, unabhängig von tol

Cube⁺(8): FFT(tol), tol = 10^{-2} , 10^{-4} , 10^{-6}

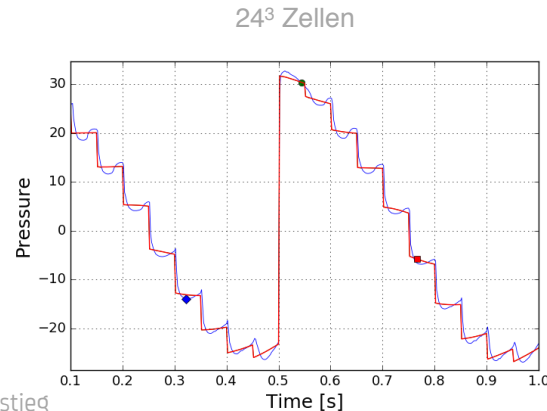
3

Testreihen

Anzahl Poisson-Lösungen pro Zeitschritt



Druckverlauf rechte Messstelle



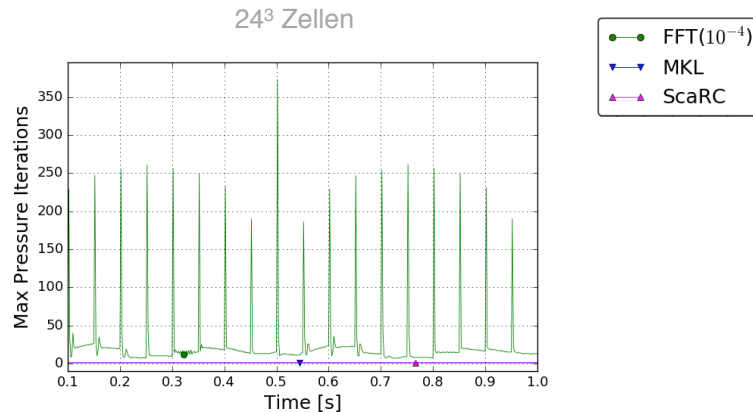
- Geschwindigkeits-Korrektur an Mesh-Grenzen deutlich aufwändiger als entlang Hindernis
- Anzahl der Poisson-Lösungen pro Zeitschritt: ~ 25 für tol= 10^{-4} , ~ 200 für tol= 10^{-6}

Cube⁺(8): Alle Verfahren im Überblick

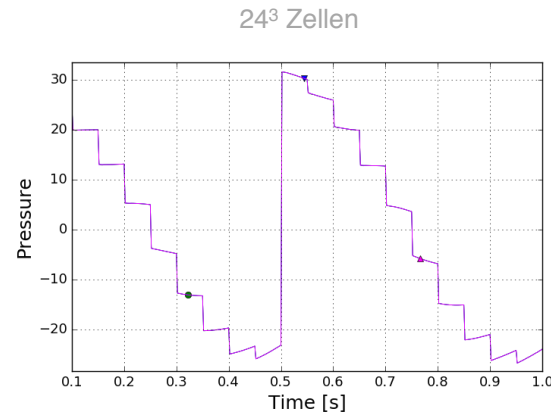
3

Testreihen

Anzahl Poisson-Lösungen pro Zeitschritt



Druckverlauf rechte Messstelle

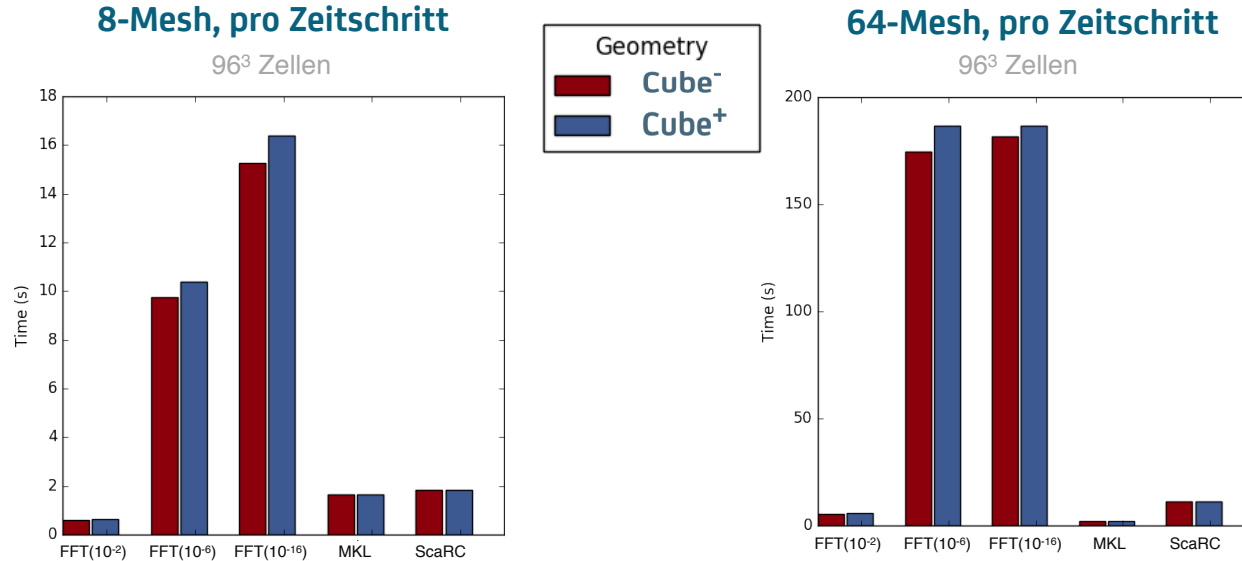


- Konsistenter Verlauf für alle Verfahren (für FFT(tol) ab tol=10⁻⁴)
- FFT(10⁻⁴) benötigt ~ 25 Poisson-Lösungen pro Zeitschritt, MKL/ScaRC genau je 1

Cube⁻ versus Cube⁺: Dauer 1 Poisson-Lösung

3

Testreihen



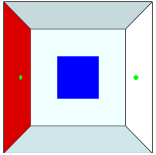
- jeweils deutlicher Anstieg der FFT-Laufzeiten für feiner werdende Toleranz
- Laufzeiten für MKL und ScaRC für beide Geometrien gleich, ScaRC für 8-Mesh konkurrenzfähig

Pipe⁺: Kanal mit Hindernissen

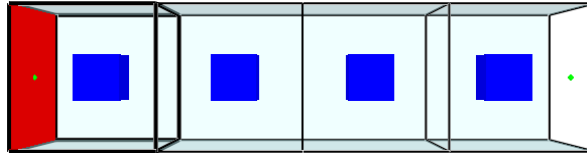
3

Testreihen

Pipe⁺(1)



Pipe⁺(4)



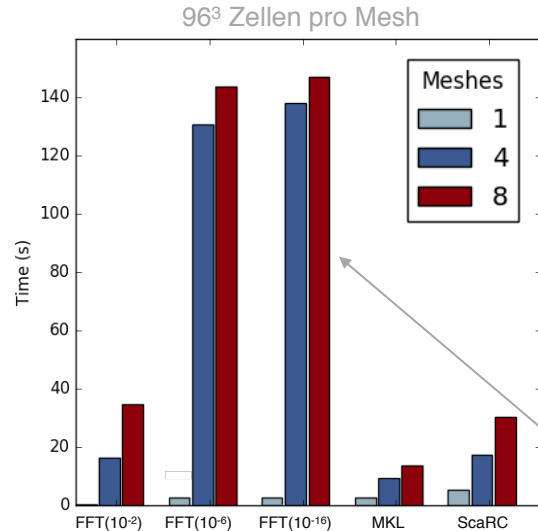
Pipe⁺(8)



- zyklische Einströmung links, offener Rand rechts, Messung der Volumenströme
- Gitterweiten 24^3 , 48^3 , 72^3 , 96^3 pro Mesh (Zell-Anzahl wächst proportional zu Mesh-Anzahl)

Pipe⁺ (M), M=1, 4, 8: Zeitvergleich Löser

Zeit für 1 Poisson-Lösung pro Zeitschritt



- FFT(tol) ist für 1-Mesh nicht zu schlagen (trotz ~10 Poisson-Lösungen pro Zeitschritt), jedoch erheblich schlechter ab 4 Meshes
- MKL und ScaRC skalieren deutlich besser
- MKL ist mehr als doppelt so schnell wie ScaRC z.B.: Pipe+(8): MKL~13.7 s versus ScaRC~30.2 s

- grundsätzlich schlechte Skalierung auf Rechner (Intel(R) Xeon, 2.67GHz, 8 CPUs zu je 8 Kernen)

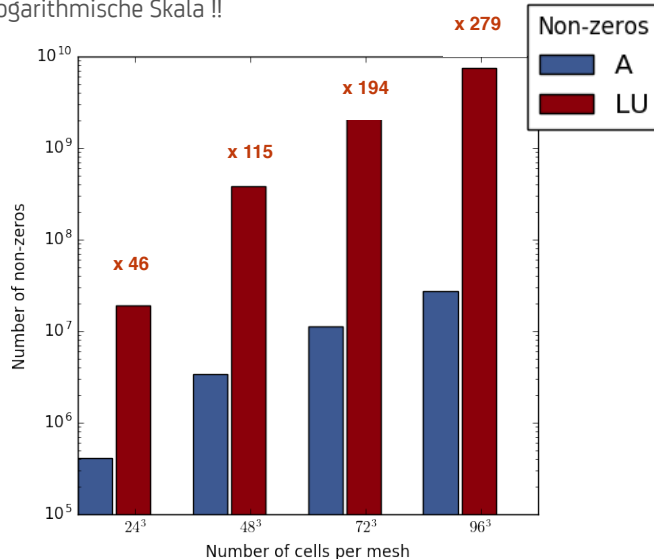
Testreihen

Pipe⁺(8): Kosten MKL-Methode

3

Testreihen

Logarithmische Skala !!



Speicherbedarf:

- sehr hoher Speicherbedarf durch „Fill-in“:
LU hat viel mehr Nicht-Null-Einträge als A

Laufzeit: Beispiel 8 Meshes mit je 96^3 Zellen

- Initialisierung: 2192.7 s
- eine Poisson-Lösung: 13.7 s

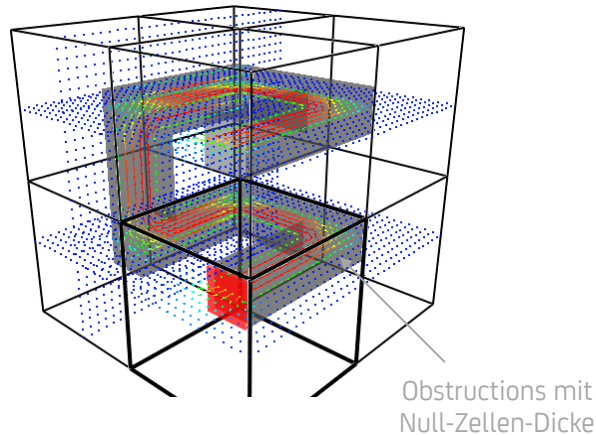
FFT und ScaRC können auf gleichen Ressourcen feinere Probleme rechnen

(Bsp: 288^3 pro Mesh für FFT und ScaRC noch rechenbar, für MKL bereits 240^3 nicht mehr)

DUCT_FLOW: Kanaldurchströmung

3

aus FDS-Verification-Guide:



8 Meshes, 128^3 Zellen

| Methode | Zeit pro Poisson-Lösung |
|------------------|-------------------------|
| FFT(10^{-4}) | 41.3 s |
| MKL | 4.4 s |
| ScaRC | 7.5 s |

Testreihen

- Vergleich FFT(tol) auf strukturiertem Gitter versus MKL und ScaRC auf unstrukturiertem Gitter

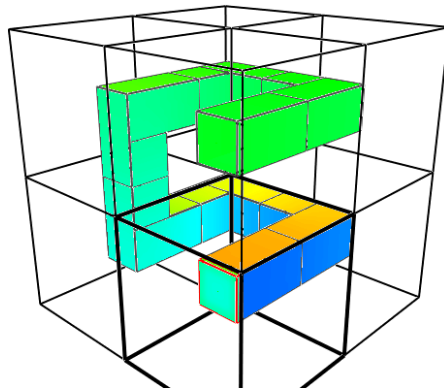
DUCT_FLOW: Fehler an Kanalwänden

3

Testreihen

FFT(10-4), 8-Mesh

64^3 Zellen

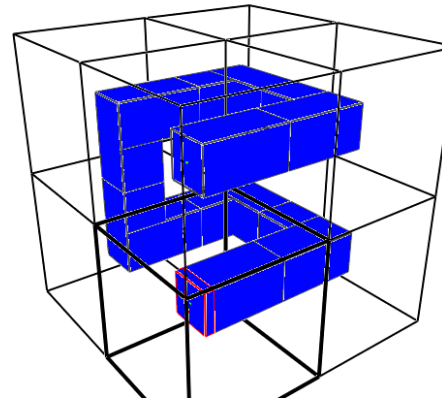


Endry
verr
m/s
 $\cdot 10^{-3}$

1.85
1.50
1.15
0.80
0.45
0.10
-0.25
-0.60
-0.95
-1.30
-1.65

MKL / ScaRC, 8-Mesh

64^3 Zellen



Endry
verr
m/s

0.00
0.00
0.00
0.00
0.00
0.00
0.00
0.00
0.00
0.00
0.00

- FFT konvergiert ausgesprochen langsam: in 1000 Schritten nur $\text{tol}=10^{-4}$
- MKL und ScaRC liefern Null-Fehler an Kanalwänden unabhängig von Mesh-Anzahl in je 1 Schritt

Fazit

Zusammenfassung

4

Ausblick

| Methode | Diskretisierung | | zusätzliche Geschwindigkeits- Korrektur | zusätzlicher Speicherbedarf | Laufzeit-Effizienz |
|----------|-----------------|----------------|---|--------------------------------|---|
| | strukturiert | unstrukturiert | | | |
| FFT(tol) | ja | nein | ja | gering | für Single-Mesh sehr hoch, aber schlechte Skalierbarkeit |
| MKL | ja | ja | nein | hoch | teure Initialisierung, gute bis moderate Skalierbarkeit |
| ScaRC | ja | ja | nein | gering | moderate Skalierbarkeit, noch Optimierungspotential |



hhpberlin
Ingenieure für Brandschutz GmbH

Hauptsitz
Rotherstraße 19 · 10245 Berlin

Amtsgericht
Berlin-Charlottenburg
Register-Nr.: HRB 78 927
Ust-ID Nr.: DE217656065

Geschäftsführung:
Dipl.-Ing. Karsten Foth
Dipl.-Inf. BW [VWA] Stefan Truthän

Beirat:
Dipl.-Ing. Margot Ehrlicher
Prof. Dr.-Ing. Dietmar Hosser
Dr.-Ing. Karl-Heinz Schubert

MKL: Einfluss OpenMP-Threads

Vergleich der Zeiten für 1, 2 und 4 OpenMP-Threads

| NUM_OMP_THREADS | MKL(Init) | | MKL-Solve |
|-----------------|-------------|----------------|-----------|
| | Permutation | Faktorisierung | |
| 1 | 19.38 s | 454.25 s | 4.56 |
| 2 | 19.32 s | 453.02 s | 4.44 |
| 4 | 19.29 s | 452.36 s | 4.41 |

- Weniger Effekt als erwartet, keine großen Unterschied
- Unterschiedliche Ergebnisse auf verschiedenen Rechnersystemen?