

PERFORMANCEOPTIMIERUNG HYGROTHERMISCHER 3D-SIMULATIONEN DURCH PARAMETEROPTIMIERUNG ITERATIVER GLEICHUNGSSYSTEMLÖSER

A. Nicolai¹ und U. Ruisinger¹

¹Institut für Bauklimatik, TU Dresden, Dresden, Deutschland

E-Mail: andreas.nicolai@tu-dresden.de

KURZFASSUNG

Trotz stetig steigender Rechenleistung moderner Computerhardware kann eine instationäre, dreidimensionale, hygrothermische Simulation immer noch ein langwieriger Vorgang sein. Um die Dauer einer Simulation zu reduzieren, stehen Nutzern verschiedene Einflussmöglichkeiten offen, deren Auswirkungen jedoch nur schwer bzw. aufwändig abzuschätzen sind. Der Artikel analysiert anhand zweier Fallbeispiele verschiedene Möglichkeiten, die Simulationszeiten zu reduzieren. Beginnend mit einer Gittersensitivitätsstudie wird die Skalierung des Aufwandes für verschiedene direkte und iterative Gleichungssystemlöser diskutiert. Weiterhin werden numerische Parameter wie Konvergenzfaktoren und Iterationsgrenzen untersucht. Basierend auf integrierten Laufzeitmessungen und Statistiken für verschiedenen Komponenten des Rechenkerns werden die iterativen Lösungsverfahren GMRES und BiCGStab in Kombination mit einem ILU Vorkonditionier betrachtet. Die Untersuchungen werden anhand der Software DELPHIN 6 durchgeführt. Der Artikel schließt mit einer Empfehlung für eine günstige Ausgangswahl an Algorithmen und Parametern in Abhängigkeit der Systemgröße und einer kompakten Methodik, diese anhand der Solverstatistiken weiter zu optimieren.

Despite continuously enhanced computer hardware, running dynamic 3D hygrothermal simulations may still be a time-consuming process. The duration of simulations may be reduced by user-selected numeric parameters. However, the actual impact of parameters like maximum Krylow subspace, convergence coefficients etc. proves difficult to estimate. In the article different methods of reducing simulation speed are applied to two example cases. Starting with a grid sensitivity study the scaling of effort for different direct and iterative linear equation system solvers is discussed. Further, we investigate parameters influencing the iterative equation system solvers GMRES and BiCGStab in combination with ILU preconditioners. The tests are done with the software DELPHIN 6 and the article concludes with recommendation for suitable initial parameter selection based on system size and presents a short methodology on how to optimise parameters based on collected solver metrics.

EINLEITUNG

Die Modellierung hygrothermischer Prozesse in kapillarporösen Baustoffen ist bereits seit längerem Stand der Technik und bereits standardisiert (DIN, 2007). Im Simulationsprogramm DELPHIN (Nicolai 2007, Nicolai 2020) sind eine Vielzahl von Modellen implementiert, wobei wir uns in diesem Artikel auf das hygrothermische Basismodell mit zwei gekoppelten Bilanzgleichungen für Energie- und Feuchtemassendichten beschränken. Bei der Verwendung der Software DELPHIN im Planungsprozess, vor allem bei der Sanierungsplanung historischer bzw. denkmalgeschützter Gebäude, werden zunehmend auch dreidimensionale Details analysiert, meist in Verbindung mit in Außenwände einbindenden Trägern oder Befestigungselementen.

Die typische Zielgruppe für die hygrothermischen Bauteilsimulation zur Planungsunterstützung sind Bauphysiker, Ingenieure aber auch Studenten und Wissenschaftler. Daher müssen Simulationsergebnisse auf typischen Stand-PCs/Workstations und Notebooks in akzeptablen Zeiten bereitstehen (Nutzung von Rechenzentren an Universitäten sind die Ausnahme). Hinsichtlich der Parallelisierung wird von 4-16 Kernen ausgegangen, bspw. bei einem Workstation-PC mit ThreadRipper 1950x CPU (2018, 16 Kerne, 3.4 Ghz), der für das 3D Beispiel verwendet wurde.

Kurzbeschreibung der relevanten Algorithmen

Das Lösungsverfahren in DELPHIN verwendet die Finite-Volumen-Methode für die Raumdiskretisierung, wobei orthogonale Berechnungsgitter mit variablen Elementdimensionen verwendet werden. Naturgemäß führt die Gittergenerierung bei 3D-Details zu einer großen Anzahl von Lösungsvariablen, welche effiziente numerische Verfahren einschließlich Parallelisierung erfordert.

Das numerische Lösungsverfahren in DELPHIN verwendet viele Teilkomponenten. Für die Zeitintegration wird der CVODE Integrator verwendet (Hindmarsh et al., 2005), einem Integrator mit Zeitschrittadaption basierend auf Fehlerkontrolle und variabler Methodenordnung.

Das implizite Lösungsverfahren (BDF) benötigt für die modifizierte Newton-Iteration Löser für lineare Gleichungssysteme. Hier stehen viele verschiedene zur Verfügung, beginnend mit direkten LU-Faktorisierungsverfahren über iterative Gleichungssystemlöser (Meister, 2008). Die direkten Verfahren sind spezialisiert auf Bandstrukturen bzw. für generische, schwach besetzte Matrizen (KLU, s. Davis & Natarajan, 2010). Für die iterativen Verfahren GMRES, BiCGStab und TFQMR (Hindmarsh et al., 2005) sind zudem Vorkonditionierer notwendig. In diesem Artikel wird ausschließlich die unvollständige LR-Zerlegung (ILU) als Vorkonditionierer (Saad, 2003) betrachtet.

Viele dieser Algorithmen und Verfahren können durch Parameter angepasst und aufeinander abgestimmt werden. Der Fokus dieses Artikels liegt auf einer Analyse einiger dieser Parameter und deren Einfluss auf die Simulationsgeschwindigkeit.

GITTERVARIATION

Numerische Fehler in der Bauteilsimulation stammen aus der Approximation der räumlichen Verteilung von Zustandsgrößen und daraus abgeleiteter Gradienten und Flussgrößen, sowie aus der zeitlichen Integration der Erhaltungsgleichungen. Erstere sind stark an das geometrische Berechnungsgitter gekoppelt. Das Thema Zeitintegrationsfehler wird im nächsten Abschnitt diskutiert.

Stand der Technik ist die Verwendung von variablen Gitterbreiten, mit typischerweise 3 wesentlichen Parametern:

- Abmessungen der Elemente an Rändern bzw. Materialgrenzen (Parameter: min),
- größte maximale Elementabmessung (Parameter: max) und
- größtes Vergrößerungsverhältnis (Parameter: stretch für Stretchfaktor) zwischen benachbarten Elementen.

Der Parameter der maximalen Elementabmessung ist typischerweise dann wichtig, wenn starke Gradienten (z.B. durch Schlagregeneintrag oder Kondensatbildung) im Inneren einer Materialschicht auftreten können. Generell ist eine Begrenzung auf 5 bis 10 cm ein sinnvoller Grenzwert.

Wesentlich kritischer ist die Dimension/Dicke eines Elements am Rand bzw. an Materialschichtgrenzen. An den Rändern treten bei realistischer Bewitterung häufig die größten Temperatur- und Feuchtegehaltsgradienten auf. An Materialgrenzen sind unstetige Materialparameter zu berücksichtigen, wobei Fehler bei der Mittellung von Transportkoeffizienten durch kleine Elementabmessungen reduziert werden können. Daher ist die Elementdicke am Rand und an Schichtgrenzen der mit Abstand wirksamste Parameter zur Reduktion von numerischen Fehlern.

Methodik der Gittersensitivitätsstudie

Das Ziel einer Gittersensitivitätsstudie ist letztlich ein möglichst grobes Gitter zu finden, ohne jedoch physikalische Effekte durch numerische Fehler zu verfälschen. Dazu wird folgendes Prozedere empfohlen, wobei jeweils ein Teil der Simulation getestet wird, der möglichst repräsentative Effekte zeigt (bspw. ein Schlagregeneignis mit anschließender Trocknung):

- Variation der Elementabmessung am Rand (bspw. 1, 1,3 und 2 mm), dabei Beibehaltung eines moderaten Vergrößerungsfaktors (z.B. 2) und maximaler Elementbreite (10 cm); Beibehaltung der Größe, bei der gegenüber der nächstkleineren Größe keine signifikanten Abweichungen auftreten.
- Variation des Vergrößerungsfaktors; Beibehaltung des Faktors, welcher gegenüber dem nächstkleineren Faktor keine groben Fehler zeigt.

Die maximale Elementbreite muss unserer Erfahrung nach nur dann beschränkt werden, wenn entweder starke Gradienten im Simulationsfall zu erwarten sind (bspw. Nachrechnung eines Aufsaugversuchs), oder falls starke Gradienten in Bereichen groben Gitters während des Post-Processing beobachtet werden (gradientenbasierte Gitterverfeinerung).

Das Kriterium *“keine groben Fehler”* wird idealerweise durch das Auswerten von Sensorgrößen an kritischen Stellen der Konstruktion ausgewertet.

Auswirkungen auf Simulationsverhalten

Die Veränderung des geometrischen Gitters beeinflusst die Simulation zweifach:

Kleine Volumina in Elementen (vor allem in Eckbereichen bewirkt bspw. die Veränderung von 1,3x1,3 mm² auf 1x1 mm² eine Verkleinerung der Fläche bzw. des Volumens um 69%) führen dazu, dass Ströme oder Quellterme in diesen Elementen zu deutlich beschleunigten Reaktionen der sensiblen Größen führen. Diese erhöhte Dynamik kann verstärkt zu Konvergenzfehlern im Newton-Verfahren bzw. mehr Iterationen führen und zwingt einen fehlerschätzer-basierten Integrator dazu, kleinere Integrationschritte zu wählen. Damit steigt bei kleineren Elementen die Anzahl der zu absolvierenden Schritte und je Schritt die Anzahl an Newton-Iterationen.

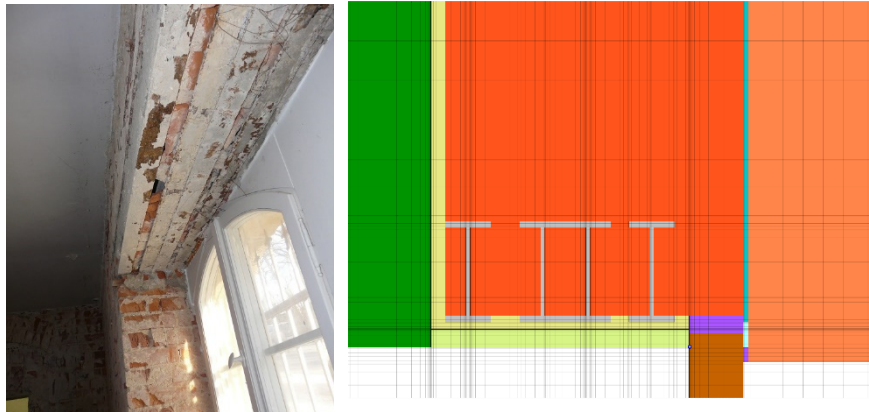


Abbildung 1: Fenstersturz vor Sanierung (links) und Simulationsmodell mit Sanierungsvariante (rechts).

- Kleinere und damit mehr Elemente vergrößern die Anzahl der Unbekannten. Der Aufwand für die meisten numerischen Algorithmen zur Lösung von Gleichungssystemen skalieren in höherer Potenz mit der Anzahl der Unbekannten, bspw. hat die LU-Zerlegung einer Matrix die Komplexität $O(n^3)$ (auch die einer Bandmatrix), iterative Verfahren skalieren je nach Verfahren mit $O(n^2)$ bzw. schwach $O(n^3)$. Damit vergrößert sich der Aufwand je Newton-Iteration zum Teil deutlich.

Die Kombination aus beiden Effekten zeigt, dass eine Gittersensitivitätsstudie *maßgeblich* zu einer Optimierung der Simulationsgeschwindigkeit beitragen kann. Nebenbei wird verhindert, dass numerische Fehler aus der Gitterwahl unbemerkt bleiben.

Beispiel für Gitteroptimierung

Anhand eines einfachen 2D-Beispiels soll eine Gitteranalyse und die Auswirkung auf die Performance gezeigt werden. Das in Abbildung 1 gezeigte Konstruktionsdetail wird in DELPHIN als 2D-Schnitt durch den Fenstersturz modelliert. Es werden exemplarisch drei Gittervarianten getestet. Tabelle 1 enthält Details zur Gittererstellung.

Die Gittersensitivitätsstudie wird mit einem direkten Gleichungssystemlöser (KLU) durchgeführt. Als Vergleichsgröße zur Bewertung der Gittervarianten wird die Gesamtfeuchtemasse in der Konstruktion betrachtet (Temperatur und Luftfeuchte in der Laibungsecke wären ebenfalls gute Größen). Abbildung 2 zeigt einen Ausschnitt der Simulationsergebnisse. Die Varianten mit normalem und feinem Gitter liefern offensichtlich sehr ähnliche Ergebnisse, sodass die Gitterdetailstufe bei der Variante „Normal“ ausreicht.

Tabelle 1:

Gittervarianten beim 2D-Fenstersturzdetail; letzte Spalte zeigt Elementzahl, Anzahl der Unbekannten (2 Bilanzgleichungen je Element) und Bandbreite (BW) der Jacobimatrix.

GITTER	MIN [mm]	STRETCH	MAX [mm]	ELEM./ VARS (BW)
grob	2,0	2,0	100	5804 / 11608 (187)
normal	1,3	1,75	100	9019 / 18038 (231)
fein	1,0	1,3	50	22663 / 45326 (371)

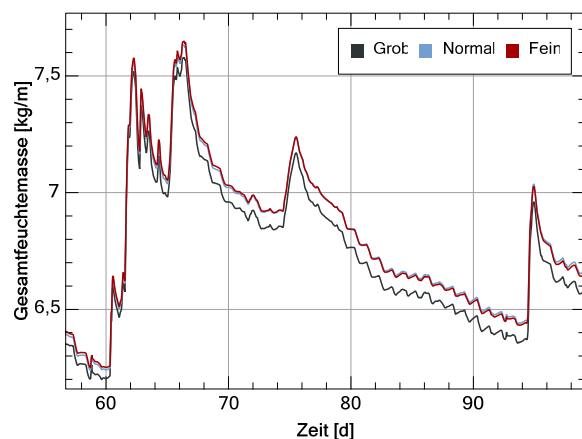


Abbildung 2: Vergleich der Ergebnisse bei der Sensitivitätsstudie. Die Ergebnisse der Varianten „Normal“ und „Fein“ liegen nahezu übereinander, beim groben Gitter gibt es sichtbare Abweichungen.

Um den Einfluss des Gitters auf die Ausführungszeiten zu verdeutlichen, zeigt Tabelle 2 eine Zusammenfassung der Laufzeiten in Minuten (für die ersten 120 Tage der Simulation) und für zwei typische Vertreter der direkten Gleichungssystemlöser (Band und KLU). Bei gleichem Gitter sind die Ergebnisse der Band-Matrix und KLU-Varianten im Rahmen der geforderten Genauigkeit nahezu identisch, und auch

das Konvergenzverhalten und die Anzahl der Integrationsschritte sind sehr ähnlich.

Tabelle 2:

Laufzeiten [min] der Simulationen und Zeitaufwand für Teilmodule des Rechenkerns für die direkten LGS-Löser Band und KLU (Jac - Jacobi-Matrix Berechnung, Fact - Faktorisierung der Matrix, Solve - Rückwärtsauflösen).

VARIANTE	LAUFZEIT	JAC	FACT	SOLVE
Band/grob	20,5	4,5	7,3	5,7
Band/normal	53,8	14,4	21,1	12,0
Band/fein	375,9	148,7	155,6	45,8
KLU/grob	13,2	0,4	7,8	2,9
KLU/normal	32,2	1,2	20,6	5,9
KLU/fein	167,4	6,4	125,0	20,4

Bei beiden LGS-Lösern wird die oben angesprochene Skalierung der Laufzeiten sehr gut deutlich, vor allem beim Teilaufwand für das Faktorisieren der Matrix. Gleichzeitig zeigt sich auch, dass traditionelle Band-Matrix-Löser den direkten Lösungsverfahren für schwach besetzte Matrizen (wie z.B. KLU) selbst bei groben Gittern und kleineren Bandbreiten unterlegen sind. Bei der Übersicht über die Laufzeitanteile in Tabelle 2 fällt weiterhin auf, dass gerade die zeitaufwändigsten Operationen Faktorisierung und Rückwärts-Auflösen strikt sequentielle Algorithmen sind, welche sich nicht parallelisieren lassen. Für eine weitere Performancesteigerung sind deshalb iterative Gleichungssystemlöser nötig.

ZEITINTEGRAL-FEHLERKONTROLLE

Moderne Zeitintegrationsverfahren verwenden Methoden zur Adaption des Zeitschritts, um die geforderte Genauigkeit einzuhalten (vgl. Kap. 8 in Press, 2002). Der Zeitintegrator CVODE in DELPHIN passt sowohl die Zeitschrittlänge als auch Methodenordnung entsprechend eines Fehlerschätzers an. Innerhalb des impliziten Verfahrens wird

die vom Newton-Verfahren gelieferte Lösung dafür verwendet. Daher muss die Newton-Iteration *genau genug* konvergieren, um hinreichend genaue Lösungsgrößen zu liefern.

Sowohl das Newton-Verfahren als auch der Fehlerschätzer basieren auf den gleichen Fehlernormen, wobei für die Konvergenz des Newton-Verfahrens das Kriterium um einen Faktor verschärft wird (üblicherweise 0,1), um eine ausreichend genaue Newton-Lösung zu erzwingen. Dieser Faktor bzw. *Koeffizient für Konvergenz des nichtlinearen Verfahrens* (NonlinConvCoef) kann ein wichtiger Optimierungsparameter sein. Ist der Koeffizient zu groß, führt das zwar zu wenig Newton-Konvergenzfehlern, dafür unter Umständen zu mehr Fehlerschätzerfehlern. Im Gegensatz dazu könnte bei einem kleinen Koeffizienten das Newton-Verfahren häufiger scheitern, dafür würden aber im Schnitt größere Zeitschritte gewählt werden und damit insgesamt weniger Newton-Verfahren ausgeführt werden. Letzteres ist auch der Grund, warum eine Verschärfung der Konvergenzforderungen an das Newton-Verfahren die Simulationsgeschwindigkeit steigern kann, obwohl dies bei anderen Simulationsprogrammen *ohne Fehlerschätzer* eher nachteilig ist.

Bei den meisten Simulationen ist der voreingestellte Faktor von 0,1 ausreichend. Beim Auftreten vieler Fehlerschätzerüberschreitungen (Abb. 3, Zeile 7) kann man diesen Koeffizienten auf 0,08 oder noch kleinere Werte reduzieren. Sollten dann in der Solver-Statistikausgabe sehr viele Newton-Konvergenzfehler ausgewiesen werden (Abb. 3, Zeile 6), so ist der Koeffizient zu klein gewählt.

Durch Anpassung dieses Koeffizienten sind in Verbindung mit iterativen Gleichungssystemlösern mitunter größere Performancesteigerungen möglich (siehe spätere 3D-Beispiele).

```

Solver statistics
-----
1  Wall clock time = 0.833 h
-----
2  Framework: Output writing = 0.006 h ( 0.69 %)
3  Framework: Step-completed calculations = 0.000 h ( 0.00 %)
4  Integrator: Steps = 9406
5  Integrator: Newton iterations = 17111
6  Integrator: Newton convergence failures = 208
7  Integrator: Error test failures = 35
8  Integrator: Function evaluation (Newton) = 0.183 h (21.93 %) 17112
9  Integrator: LES setup = 0.247 h (29.69 %) 1380
10 Integrator: LES solve = 0.372 h (44.65 %) 17111
11 LES: Linear iterations = 34916
12 LES: Linear convergence failures = 1475
13 LES: Function evaluations (LS solve) = 0.000 h ( 0.00 %)
14 LES: Function evaluations (Jacobian gen.) = 0.140 h (16.77 %) 16666
15 LES: Jacobian matrix assembly time = 0.174 h (20.84 %)
16 LES: Matrix-vector multiplications = 0.035 h ( 4.23 %) 69832
17 LES: Function evaluations (precond. setup) = 0.000 h ( 0.00 %) 0
18 LES: Preconditioner setup = 0.074 h ( 8.86 %) 641
19 LES: Preconditioner solves = 0.251 h (30.15 %) 86899
-----
    
```

Abbildung 3: Statistiken und Performancemetriken bei einem Solverlauf mit iterativen Gleichungssystemlösern.

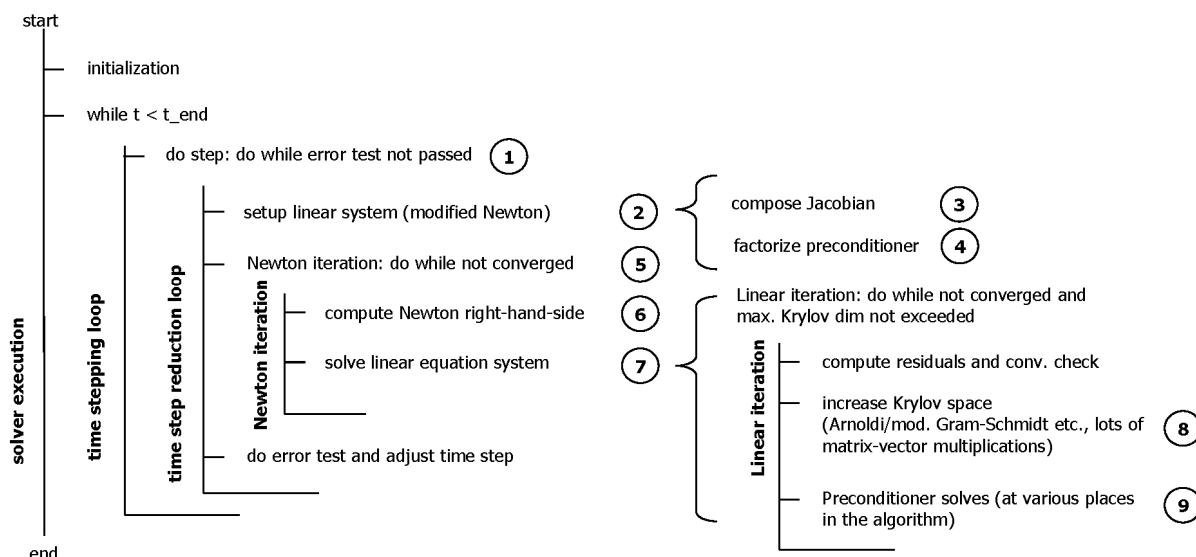


Abbildung 4: Schematik der Module und Schleifenstrukturen des numerischen Rechenkerns bei Verwendung iterativer Gleichungssystemlöser.

ITERATIVE GLEICHUNGSSYSTEMLÖSER

Anstelle der direkten LGS-Löser werden nun iterative Lösungsverfahren aus der Klasse der Krylow-Unterraumverfahren verwendet, wobei in diesem Artikel lediglich die Verfahren GMRES und BiCGStab (Meister 2007, Saad 2003) verwendet werden. Beide sind für asymmetrische Matrizen geeignet, wobei die Matrizen in DELPHIN zwar assymetrisch sind, aber grundsätzlich eine symmetrische *Besetzung* aufweisen (welches hinsichtlich der Speicherstruktur und Matrizenoperationen ausgenutzt wird).

Mit den Iterationsschleifen des linearen Gleichungssystemlösers sind insgesamt 4 Schleifen geschachtelt, was in Abbildung 4 verdeutlicht wird.

Während der Simulationsausführung protokolliert DELPHIN die Anzahl der Aufrufe einzelner Module und die Zeit, welche dafür verwendet wird. Ein Beispiel für die generierte Ausgabe der Solverstatistiken und Metriken ist in Abbildung 3 gezeigt. Diese ist hilfreich für die Optimierung der einzelnen Numerikparameter. Relevant für die Auswertung sind folgende Zeilen:

- Z1. Gesamtlaufzeit der Simulation
- Z4. Gesamtzahl der Integrationsschritte
- Z5. Anzahl der Newton-Iterationen
- Z6. Newton-Konvergenzfehler
- Z7. Fehlerschätzerüberschreitungen
- Z8. Auswertungen der Systemfunktion während der Newton-Iterationen
- Z9. Vorbereitung des linearen Gleichungssystems
- Z10. Lösen des linearen Gleichungssystems

Der Zeitaufwand in den Zeilen 2, 3, 8, 9 und 10 summiert sich in etwa zur Gesamtlaufzeit der Simulation (abzüglich eines kleinen nicht erfassten Aufwands in kleineren Funktionen). Der Aufwand für

das Vorbereiten des linearen Gleichungssystems ist weiterhin aufgeschlüsselt in den Zeilen:

- Z14. Funktionsauswertungen zur Berechnung der Jacobimatrix
- Z15. Gesamtaufwand für die Berechnung der Jacobimatrix (enthält Z14)
- Z18. Berechnung und im Fall der ILU die Zerlegung der Vorkonditionierer-Matrix

sodass sich Z15 und Z18 zusammen ungefähr zum Zeitaufwand in Zeile 9 summieren. In Abbildung 4 entspricht das (2), (3) und (4).

Der Aufwand für das Lösen des linearen Gleichungssystems (Summe in Zeile 10) schlüsselt sich auf in:

- Z16. Matrix-Vektor-Multiplikationen
- Z19. Anwendungen des Vorkonditionierers

sodass Z16 und Z19 (und ein nicht erfasster Zusatzaufwand des eigentlichen Iterationsverfahrens) zusammen den Zeitaufwand in Zeile 10 ergeben. Dies entspricht in Abbildung 4 den Punkten (7), (8) und (9), d. h. der innersten Schleife des Algorithmus.

Hinsichtlich der Parallelisierung des Rechenkerns sollte noch erwähnt werden, dass die Operationen des ILU-Vorkonditionierers (Zeilen 18 und 19) nicht parallelisiert werden können. Für die eigentliche Performance-Optimierung sind jedoch nicht die absoluten/relativen Zeiten wichtig, sondern vielmehr die Ausführungsanzahl der einzelnen Operationen, bzw. die Fehleranzahl (siehe Zeilen 6, 7 und 12).

Konvergenzforderungen

Die Verwendung eines iterativen Gleichungssystemlösers innerhalb einer Newton-Iteration weist einige Besonderheiten auf. Zum einen sind die Änderungen der Lösungsgrößen von einem

Newton-Schritt zum nächsten typischerweise klein und werden mit zunehmender Konvergenz des Newton-Verfahrens sehr klein. Dadurch werden üblicherweise nur wenige Iterationen benötigt. Entscheidend für die Konvergenz des Newton-Verfahrens ist jedoch, dass die im iterativen Verfahren berechnete Lösung (=Korrekturvektor für den Newton-Schritt) *genau genug* ist, um Newton-Konvergenz zu ermöglichen.

Als Konvergenzkriterium für die Lösung des linearen Gleichungssystems (LGS) wird die gewichtete Quadratsummen-Wurzelnorm des Residualvektors mit einem Grenzwert verglichen. Der Grenzwert bestimmt sich aus den Genauigkeitsforderungen des Newton-Algorithmus. Zusätzlich wird der Grenzwert durch einen *linearen Konvergenzkoeffizienten* (LinIterConvCoef) < 1 abgemindert. Dadurch wird gefordert, dass das Residuum genauer bestimmt wird, als die vom Newton-Verfahren geforderte Genauigkeit.

Ist die Genauigkeitsforderung an das lineare Verfahren zu hoch, kann dies zu sehr vielen Iterationen führen, bzw. das Verfahren konvergiert aufgrund der begrenzten Zahlengenauigkeit bei Computerzahlen überhaupt nicht. Deshalb wird die Anzahl der Iterationen und damit die *Dimension des Krylow-Unterraums* durch einen Parameter begrenzt. Die Iteration wird beim Erreichen dieser Grenze abgebrochen, ein linearer Konvergenzfehler verzeichnet (siehe Abb. 3, Zeile 12) und die bis dahin berechnete Lösung dem Newton-Verfahren übergeben. Dies wirkt sich mitunter negativ auf die Konvergenz des Newton-Verfahrens aus. Daher ist eine große Anzahl an Konvergenzfehlern im linearen Gleichungssystemlöser immer auch ein Indikator für verschenkte Performance. Im Gegenzug sollte man den Krylow-Unterraum nicht zu groß wählen, da vor allem beim GMRES-Verfahren der Aufwand mit der Größe des Unterraums steigt.

Für die effiziente Verwendung eines iterativen Gleichungssystemlösers ist ein Vorkonditionierer notwendig, wobei in diesem Artikel ausschließlich die unvollständige LR-Zerlegung der Matrix verwendet wird (ILU). Die ILU-Faktorisierung und Anwendung ist allerdings ein sequentieller Algorithmus und kann nicht parallelisiert werden

Tabelle 3 zeigt für die Variante mit „normalem“ Gitter den Vergleich der Laufzeiten bei den direkten Verfahren und den iterativen Lösern. Letztere wurden jeweils mit verschiedenen Krylow-Unterraum-Begrenzungen ausgeführt. Wie erwartet nimmt die Anzahl der Konvergenzfehler im LGS-Löser bei größeren Unterräumen (mehr Iterationen) ab, welches sich auch günstig auf die Newton-Iterationen auswirkt.

Tabelle 3:

Laufzeiten [min] und Statistiken für verschiedene LGS-Löser (Steps - Anzahl der Integrationsschritte, NLC - Konvergenzfehler im linearen Gleichungssystemlöser, NC - Konvergenzfehler des Newton-Verfahrens, ERR - Fehlerschätzerüberschreitungen). Fett gedruckt sind die jeweils besten Varianten für die beiden Verfahren.

VARIANTE	ZEIT	STEPS	NLC	NC	ERR
Band	53,8	61999	-	415	2696
KLU	32,2	63360	-	424	2752
GMRES(10)	17,5	70034	36403	3926	1280
GMRES(15)	12,9	62421	6314	652	2487
GMRES(20)	13,1	61419	916	416	2630
GMRES(25)	12,8	61881	148	438	2629
GMRES(30)	12,6	61025	31	413	2644
GMRES(40)	13,0	61494	0	417	2683
BiCGStab(10)	13,8	62369	8807	729	2445
BiCGStab(15)	13,0	61577	1104	423	2657
BiCGStab(20)	13,2	62061	116	420	2617
BiCGStab(30)	13,9	61300	4	402	2704

Interessant ist die Variante GMRES(10), bei der die Anzahl der linearen Iterationen eindeutig zu stark begrenzt ist. Dies führt zu sehr vielen vorzeitigen Verfahrensabbrüchen im LGS-Löser, und entsprechend starken Konvergenzproblemen im Newton-Verfahren. Da ein Newton-Konvergenzfehler zur Wiederholung der Integrationsschritte mit kleineren Schrittgrößen führt, sind diese bei diesem Fall häufig *konvergenzdominiert*, statt durch den Integrationsfehlerschätzer begrenzt zu werden. Dies drückt sich durch eine entsprechend geringere Anzahl an Fehlerschätzerfehlern (ERR) aus.

Bei allen anderen Fällen liegen die Zeitschritte in etwa in der gleichen Größenordnung, welches an der ungefähr ähnlichen Gesamtzahl an Integrations-schritten ablesbar ist. Es fällt auch auf, dass die Güte der Lösung bei den iterativen Verfahren den der exakten direkten Verfahren entspricht. Das Optimum beim GMRES liegt bei 30 Iterationen und beim BiCGStab bei 15 Iterationen, ist also eindeutig verfahrensabhängig. Wird die maximale Krylow-Unterraumgröße zu groß gewählt, leidet die Performance.

Aus der 2D-Studie und unseren Erfahrungswerten aus vielen anderen Simulationen können wir bereits folgende Schlussfolgerungen ziehen:

- Die Veränderung des Gitters hat den weitaus größten Einfluss auf die Simulationsgeschwindigkeit, vor allem bei der Verwendung von direkten Verfahren.
- Bei 2D-Problemen ist der KLU-Löser grundsätzlich dem traditionellen Band-Matrix-Löser überlegen.

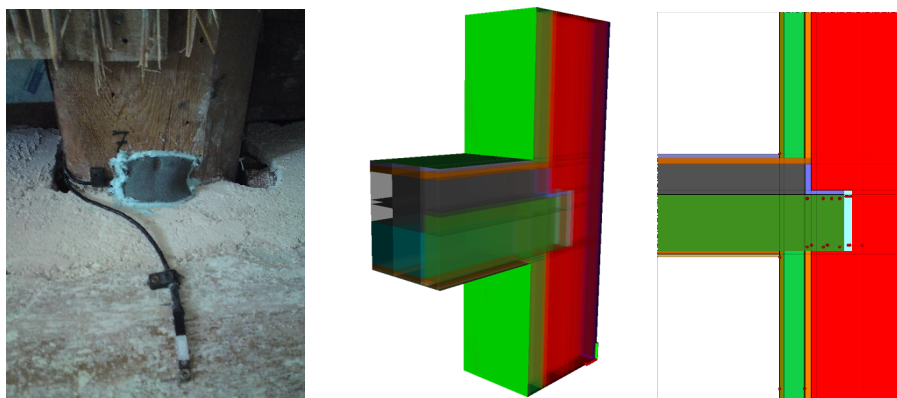


Abbildung 5: Balken (instrumentiert, von unten fotografiert) vor Anbringen der Dämmung (links), 3D Simulationsmodell mit Symmetrieschnitt durch den Deckenbalken (mitte), 2D-Modellansicht der Schnittebene ohne Berechnungsgitter und mit Punkten der Ergebnisausgabe (rechts).

- Bereits bei Modellgrößen von ca. 5000 Elementen sind die iterativen Gleichungssystemlöser den direkten Verfahren überlegen (z.B. benötigt GMRES(30) nur 5,2 min beim groben Gitter verglichen mit 13,2 min bei Verwendung von KLU).
- Die optimale Maximalgröße des Krylow-Unterraums hängt vom Lösungsverfahren und physikalischen Problem ab und sollte individuell optimiert werden.

3D-MODELLE

Die Optimierung von 3D-Modellen erfolgt prinzipiell in ähnlicher Weise. Aufgrund der langen Simulationszeiten (nach bereits langen Initialisierungszeiten) ist es in der Ingenieurspraxis nicht zielführend, mehrere Parametervarianten über einen längeren Simulationszeitraum auszutesten. Die Einsparung in der endgültigen Simulation würden durch den anfänglichen Mehraufwand überdeckt werden. Wir haben für ein 3D-Fallbeispiel verschiedene Parametervarianten getestet. Bei diesem Beispiel durchstößt ein Deckenbalken eine Dämmschicht (siehe Abb. 5). Zur besseren Übersicht ist nicht das endgültig verwendete Diskretisierungsraster gezeigt, sondern lediglich die Schichten, welche sich aus dem Geometriaufbau ergeben.

Ausgehend von einer vorgelagerten Gittersensitivitätsstudie wurde ein Gitter mit den Generierungsoptionen $min=2mm$, $stretch=2$, $max=200mm$ gewählt (381300 Lösungsvariablen), welches hier einen guten Kompromiss zwischen erträglicher Rechenzeit und Genauigkeit bildet.

Variation: Iterationsgrenze beim linearen Verfahren

Basierend auf den Erfahrungen mit dem 2D-Fallbeispiel werden Variantenstudien unter Veränderung der Maximalgröße des Krylow-Unterraums durchgeführt. Dabei werden die beiden iterativen Solver GMRES und BiCGStab verwendet.

Die physikalischen Ergebnisse aller Varianten (auch der späteren, längeren Simulationen) waren durch die Integrationsfehlerkontrolle nahezu identisch, weswegen auf eine Auswertung der Ergebnisse verzichtet werden kann.

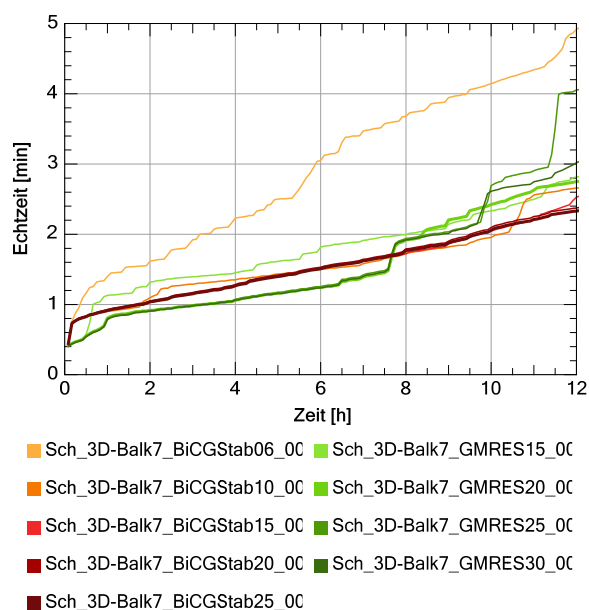
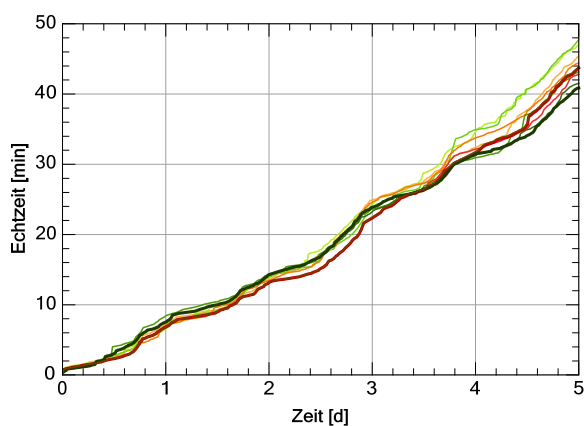


Abbildung 6: Simulationengeschwindigkeitsvergleich bei veränderten Iterationszahlengrenzen für die linearen Gleichungssystemlöser. Rechendauer (Echtzeit) ist aufgetragen über Simulationszeit, niedrigere Kurven bedeuten schnellere Simulation. GMRES(20) und BiCGStab(25) sind die jeweils besten Varianten (dicke Linien)

Zunächst wurden folgende numerische Parameter verwendet: $reltol = 5e-5$, $NonlinConvCoef = 0.1$, $LinIterConvCoef = 0.08$. Die letzteren Parameter sind die oben besprochenen Konvergenzkoeffizienten für das Newton-Verfahren und das lineare Krylow-Unterraum-Iterationsverfahren. Die Simulationen wurden zunächst jeweils für einen Simulationszeitraum von 12 h (ca. 4 min Echtzeit je Simulation) ausgeführt.

Abbildung 6 zeigt ein Echtzeit-Simulationszeit-Diagramm, bei dem der zeitliche Verlauf der Simulationsgeschwindigkeit gut ablesbar ist. Die Varianten GMRES(20) mit MaxKrylowDim = 20 und BiCGStab(25) (d.h. max. 25 lineare Iterationen) waren dabei die schnellsten. Bei allen Varianten mit mehr als 20 Iterationen traten nur noch sehr wenige Konvergenzfehler im LGS auf. Eine weitere Erhöhung der Iterationsgrenze führte bis zum Ende der 12 h eher zu einer Verlangsamung. Die Variante BiCGStab(6) zeigt eindrucksvoll an der deutlich längeren Simulationszeit (Faktor 2), welchen Einfluss eine zu starke Begrenzung der Iterationen hat.

Die ersten Stunden der Simulation sind jedoch häufig noch nicht wirklich aussagekräftig, da sie vor allem bei massiven Konstruktionen durch die Anfangsbedingungen dominiert werden. Die Simulationen wurden daher bis zu 5 Tagen Simulationszeit fortgeführt (Abb. 7).



- Sch_3D-Balk7_GMRES15_0C ■ Sch_3D-Balk7_BiCGStab15_0C
- Sch_3D-Balk7_GMRES20_0C ■ Sch_3D-Balk7_BiCGStab20_0C
- Sch_3D-Balk7_GMRES25_0C ■ Sch_3D-Balk7_BiCGStab25_0C
- Sch_3D-Balk7_GMRES30_0C ■ Sch_3D-Balk7_BiCGStab30_0C
- Sch_3D-Balk7_GMRES35_0C ■ Sch_3D-Balk7_BiCGStab35_0C

Abbildung 7: Geschwindigkeitsvergleich in den ersten 5 Tagen. GMRES(35) und BiCGStab(30) sind die jeweils besten Varianten (dicke Linien).

Zunächst fällt auf, dass die Unterschiede in der Simulationszeit insgesamt doch recht gering sind. Die Varianten mit den eindeutig zu kleinen Iterationsgrenzen (MaxKrylowDim < 15) sind im Diagramm nicht dargestellt. Bei den Varianten mit Krylow-Unterraum > 25 treten kaum noch Konvergenzfehler beim linearen Verfahren auf. Eine weitere Erhöhung der Iterationsgrenze brachte keine weitere Leistungssteigerung, z.B. war die Variante BiCGStab(35) leicht langsamer als BiCGStab(30). Auch besteht das Risiko, die mit zunehmender Iterationszahl akkumulierenden numerischen Fehler im GMRES- oder BiCGStab-Verfahren in das Newton-Verfahren einzubringen. Dieses konvergiert dann zwar, jedoch auf ein verfälschtes Ergebnis,

welches dann zu einer erhöhten Zahl von Fehlerschätzerüberschreitungen führt.

Variation: Konvergenzkoeffizient für das Newton-Verfahren

Dies lässt sich durch Anpassung des bereits beschriebenen Konvergenzkoeffizienten für die nichtlineare Iteration verbessern. Abbildung 8 zeigt den Vergleich der Varianten GMRES(30), mit unverändertem LinIterConvCoef = 0.08, aber verschiedenen Werten für NonLinConvCoef = 0,01, 0,02, 0,04, 0,06, 0,08 und 0,1 (bisherige Varianten).

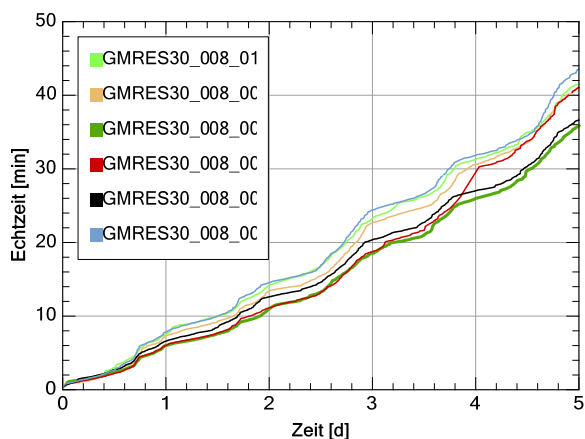


Abbildung 8: Geschwindigkeitsvergleich bei den Varianten mit verändertem Konvergenzkoeffizienten für die nichtlineare Iteration.

Offensichtlich gelingt es bei Verschärfung der Konvergenzforderungen für das Newton-Verfahren, insgesamt eine genauere numerische Lösung zu erhalten, was zu insgesamt größeren Zeitschritten und damit weniger Gesamtaufwand führt. Es fällt auf, dass bei einem Faktor 0,02 und 0,06 die Geschwindigkeit steigt, bei 0,04 und 0,01 jedoch sinkt.

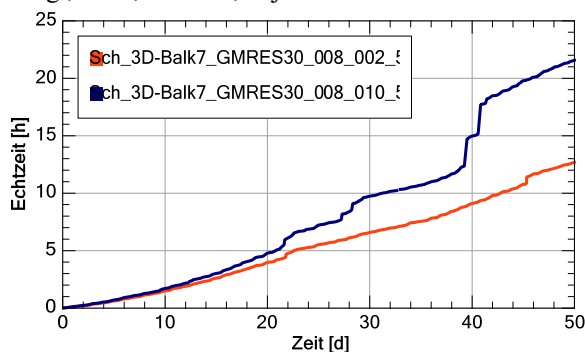


Abbildung 9: Langzeittrend der Varianten mit NonLinConvCoeff=0,1 und 0,02

Dieses Verhalten ist zum einen problemabhängig und gleichzeitig nicht direkt von GMRES auf BiCGStab übertragbar. Auch scheint die Wirkung dieses Parameters bei relativen Toleranzen $> 10^5$ ausgeprägter zu sein. Der Parameter muss also individuell angepasst werden. Der Aufwand lohnt sich jedoch, wie in Abb. 9 im Langzeitvergleich der Basisvariante mit Standardparametern GMRES(30-0,1-0,08) und der optimierten Variante GMRES(30-0,02-0,08) zu sehen ist. Hochgerechnet auf eine Jahressimulation entspricht dies mehr als 30% Simulationszeiteinsparung.

OPTIMIERUNGSMETHODIK

Zum gegenwärtigen Zeitpunkt lässt sich unserer Erfahrung nach keine allgemeingültige Vorschrift ableiten, welche Kombination aus numerischen Parametern für beliebige Geometrien, Materialdatensätze und Randbedingungen jeweils am günstigsten ist. Eine dynamische Anpassung des Krylov-Unterraums während der Simulation, basierend auf den Messwerten (Konvergenzfehlern) könnte allerdings recht einfach in die Software integriert werden, um so zumindest den wichtigsten Parameter automatisiert zu kalibrieren.

Bei langwierigen Simulationen schlagen wir daher folgende praktische Vorgehensweise zur Optimierung der Parameter vor, wobei ab einer Problemgröße von > 4000 Unbekannten jeweils GMRES(20), bzw. ab 8000 Unbekannten GMRES(30) als Startvariante für den LGS-Löser verwendet wird:

- Ausführung zweier Gittervarianten (*grob* und *normal*) parallel auf zwei getrennten Rechnern. Dabei wird der Unterschied von kritischen Sensorgrößen beobachtet. Weicht das gröbere Gitter zu stark ab, bricht man die Simulation ab und lässt auf dem gleichen Rechner eine noch detaillierte Variante (*fein*) laufen (währenddessen läuft die Variante *normal* weiter). Das Prozedere wird wiederholt, bis ausreichende Übereinstimmung erzielt wird (danach kann die Simulation mit dem detaillierteren Gitter abgebrochen werden).
- Jeder Simulationslauf wird hinsichtlich der linearen Konvergenzfehler überwacht (siehe Abb. 3, Zeile 12, DELPHIN-Log-Datei: LES_iterative_stats.tsv, Spalte NLCFails). Treten sehr viele Konvergenzfehler auf, d.h. das Verhältnis $NLCFails/Steps > 10\%$, wird die Simulation abgebrochen und der Krylov-Unterraum um 5-10 vergrößert. Um nicht die Ergebnisse der bisherigen Simulation zu verschenken, kann die Berechnung in DELPHIN einfach *unterbrochen* und mit verändertem Parameter *fortgesetzt* werden.

- Treten nur noch verhältnismäßig wenig lineare Konvergenzfehler auf, so kann man versuchen, den Konvergenzkoeffizienten für die nichtlineare Iteration zu reduzieren. Dafür sollten dann, wie in Abb. 8 gezeigt, einige Varianten über einen relevanten Zeitraum gerechnet werden, um den Einfluss zu analysieren. Auch hier kann durch Unterbrechen und Fortsetzen der Simulation Zeit gespart werden.

Ungeachtet dessen bleibt eine Gittersensitivitätsstudie und damit mögliche Reduktion der Systemgröße die wichtigste Stellschraube für die numerische Performance.

DANKSAGUNG

Die 3D-Simulationen basieren auf dem Forschungsprojekt OEKO-ID. Dieses wurde vom Österreichischen "Klima- und Energiefonds" der FFG innerhalb der Programmlinie „Neue Energien 2020“ gefördert und von der Steiermärkischen Landesregierung sowie vom Bundesministerium für Unterricht, Kunst und Kultur kofinanziert.

LITERATUR

- Davis, T., & Natarajan, E. 2010. Algorithm 907: KLU, A Direct Sparse Solver for Circuit Simulation Problems. ACM Trans. Math. Softw., 37(01).
- DIN, 2007. DIN EN 15026:2007-07 - Hygrothermal performance of building components and building elements - Assessment of moisture transfer by numerical simulation. Beuth Verlag.
- Hindmarsh, A.C., Brown, P.N., Grant, K.E., Lee, S.L., Serban, R., Shumaker, D.E., & Woodward, C.S. 2005. SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. ACM TOMS, 31(3), 363–396.
- Meister A. 2008. Numerik linearer Gleichungssysteme. 3. Auflage edn. Springer Vieweg.
- Nicolai A. 2007. Modeling and Numerical Simulation of Salt Transport and Phase Transitions in Unsaturated Porous Building Materials. Ph.D. thesis, Syracuse University.
- Nicolai A. 2020. DELPHIN 6 Webseite. <https://bauklimatik-dresden.de/delphin>.
- Press W. 2002. Numerical recipes in C++ : the art of scientific computing. Cambridge, UK New York: Cambridge University Press.
- Saad Y. 2003. Iterative Methods for Sparse Linear Systems, Second edition. 2. ed., [nachdr.] edn. Philadelphia, PA: SIAM, Society for Industrial and Applied Mathematics.