

# Rapid Interactive Structural Analysis

Daniel Weber, Tim Grasser, Johannes Mueller-Roemer, André Stork (Fraunhofer IGD)

## 1 Zusammenfassung

Auf dem Weg zu einem hocheffizienten Produktentwicklungsprozess, der aus Design, Simulation, Analyse und Iterationen besteht, liegen noch einige ungenutzte Potenziale. Bei der Analyse des Prozesses wird häufig die Integration verschiedener Software-Werkzeuge entlang der Prozesskette als eine der Schwachstellen identifiziert. Hierbei sind Interoperabilität und die Standardisierung von Austauschformaten insbesondere für die Kombination von Software verschiedener Hersteller von zentraler Bedeutung. Jedoch hat die Dauer der Simulation ebenfalls einen maßgeblichen Einfluss auf die Effizienz, um schlussendlich Produkte schneller auf den Markt zu bringen oder eine höhere Qualität zu erzielen. Wenn die Ergebnisse von Simulationen praktisch direkt nach ihrem Start zur Verfügung stehen würden, so könnten einzelne Iterationsschleifen drastisch verkürzt und damit eine Vielzahl von Design-Variationen exploriert werden. Auch die rechnergestützte Formoptimierung, bei der Hunderte von Simulationsrechnungen automatisiert durchgeführt werden, würde von solch kurzen Simulationszeiten stark profitieren. Im Projekt *Rapid Interactive Structural Analysis* wurde eine schnelle, interaktive Simulationslösung mit direkter Visualisierung auf Basis finiter Elemente entwickelt. Durch Nutzung der zur Verfügung stehenden, immensen Rechenpower von Graphikkarten (GPUs) können Simulationen, wie beispielsweise strukturmekanische Analysen, signifikant beschleunigt werden. Der Lösungsansatz basiert auf massivparallelen Algorithmen und beschleunigt dadurch linear-elastische Struktursimulationen um einen Faktor von bis zu 80. Mit dieser schnellen Simulationstechnologie werden neuartige Anwendungen möglich, wie beispielsweise die direkte Identifikation von Korrelationen zwischen geometrischen Änderungen und Spannungsverteilung oder die signifikante Beschleunigung von Form- oder Topologieoptimierungen. Die Genauigkeit und Geschwindigkeit des vorgestellten Ansatzes zu herkömmlichen Simulationen auf Basis der Finite-Elemente-Methode (FEM) wird verglichen.

## 2 GPU-beschleunigte Simulation

Die Prozessoren von Graphikkarten (GPU - graphics processing units) haben eine beachtliche Entwicklung hinter sich. Eine immer größer werdende Anzahl von Kernen ermöglicht immer mehr Gleitkommaoperationen pro Sekunde (FLOPs - floating point operations per second). Neueste Generationen von GPUs, wie beispielsweise die A100 von Nvidia [1] erreichen durch die Nutzung von fast 7000 CUDA-Cores 9,7 TeraFLOPs für doppelte bzw. 19,5 TeraFLOPs für einfache Genauigkeit.

Die Nutzung von Graphikkarten nicht zur Bildgenerierung, sondern als „Number Cruncher“, ist bekannt unter dem Namen *General-purpose computing on graphics processing units* (GPGPU). In diesem Feld gab es zu Beginn neben zahlreicher Forschungsarbeiten auch erste kommerzielle Simulationssoftware, die GPU-beschleunigte Algorithmen einsetzte. In Georgescu et al. [2] wurden diese analysiert, und es wurden vergleichsweise geringe Beschleunigungen berichtet, da höchstwahrscheinlich nur einzelne, numerisch intensive Berechnungen durch ihr GPU-beschleunigtes Pendant ersetzt wurden. Das hat zur Folge, dass große Datenmengen zwischen CPU- und GPU-Hauptspeicher transferiert werden müssen und den Geschwindigkeitsvorteil wieder mindern. Mittlerweile gibt es für viele kommerziell verfügbare Simulationssoftware im Bereich der Strukturmechanik die Möglichkeit, die GPU zur Beschleunigung einzusetzen [3]. Allerdings wird in den meisten Anwendungen das Simulationsmodell aus dem Preprocessor über das Dateisystem an den eigentlichen Löser übergeben. Ähnlich ist es bei der Übergabe der Simulationsergebnisse an den Postprocessor: Eine direkte Visualisierung oder Auswertung der Simulationsergebnisse, insbesondere, wenn die Berechnungen auf der GPU stattfinden, wird nicht eingesetzt – hier liegt viel ungenutztes Potenzial zur Beschleunigung.

Mit Ansys Discovery live [4] gibt es einen Ansatz, der sowohl GPU-beschleunigte Simulation nutzt als auch das mit direkter Visualisierung verbindet. Darüber hinaus ist es möglich, geometrische Änderungen durchzuführen. Diese werden direkt in die Simulation übertragen und die Effekte sofort visualisiert. Jedoch ist das eingesetzte numerische Verfahren und die Qualität der Simulationsnetze nicht transparent.

Das Projekt *Rapid Interactive Structural Analysis* setzt ebenfalls auf eine GPU-beschleunigte Simulation in Verbindung mit einer direkten Visualisierung der Simulationsergebnisse. Für die Eingabe werden hier jedoch Tetraedernetze eingesetzt und zur Strukturanalyse die Methode der Finite Elemente genutzt. In Kombination mit einem hochqualitativen Netzgenerator können so Ergebnisse vergleichbar zu den sehr

verbreiteten FEM-Softwarepaketen erzielt werden, allerdings mit signifikanter Beschleunigung. Zur Erschließung des Beschleunigungspotenzials wurden spezialisierte Datenstrukturen und Algorithmen entwickelt.

Zur Minimierung von Datentransfers und damit zur vollen Nutzung des Potenzials von Graphikkarten werden verschiedene Strategien verfolgt. Die Simulationsergebnisse, die nach der Berechnung im GPU-Speicher vorliegen, werden direkt für die Visualisierung genutzt. Beispielsweise kann die von-Mises Vergleichsspannung pro Element oder pro Knoten ebenfalls GPU-beschleunigt berechnet und für eine Falschfarbvisualisierung mit Hilfe einer Farbrampe direkt die Einfärbung der Pixel bestimmt werden. Ein weiterer Fokus ist es, möglichst viele Berechnungen auf der GPU durchzuführen (siehe Abbildung 1). Im Gegensatz zu anderen Ansätzen, wird hier das Tetraedernetz zu Beginn in den Graphikspeicher übertragen und die Berechnung der Elementsteifigkeiten sowie der Zusammenbau der globalen Steifigkeitsmatrix GPU-beschleunigt durchgeführt. Diese Schritte ebenfalls auf der GPU zu berechnen, führt zu weiterer Beschleunigung, zu minimalem Datentransfer und damit verringerten Berechnungszeiten.



Abbildung 1: Übersicht der Berechnungen, die auf der CPU und GPU ausgeführt werden.

Zur vollständigen Abdeckung der Pipeline wurde ein weiteres Tool entwickelt, in dem man CAD-Daten importieren und Randbedingungen auf CAD-Flächen spezifizieren kann (siehe Abbildung 2). Durch Anbindung eines externen Netzgenerators können hoch-qualitative Finite-Elemente-Netze generiert und genutzt werden.

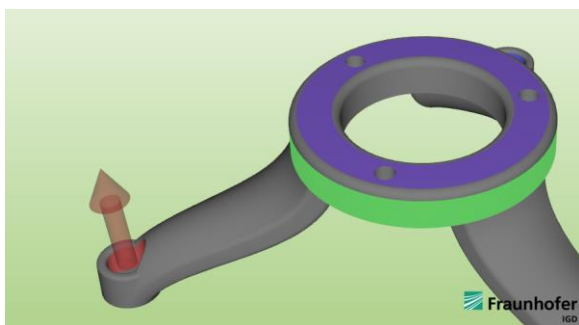


Abbildung 2: Screenshot eines Tools für den Import von CAD-Daten, der Spezifikation von Randbedingungen mit Anbindung eines Netzgenerators. Das resultierende Netz kann direkt an den Simulator übergeben werden.

### 3 Auswertung

Um die Laufzeit und die Qualität des entwickelten GPU-beschleunigten Ansatzes einzuordnen, wurden verschiedene Szenarien mit zwei kommerziellen Simulationstools („Software A“ und „Software B“) berechnet und verglichen. In beide kommerzielle Simulationstools wurden jeweils CAD-Geometrien importiert, vernetzt, Randbedingungen spezifiziert und eine lineare Strukturanalyse durchgeführt. Zusätzlich wurde das Simulationsnetz aus den generierten Dateien extrahiert, um für den GPU-beschleunigten Ansatz dieselben Netze zu verwenden. Die Tests wurden auf einer Workstation mit Intel i7 6700K@4.0GHz Quad-Core Prozessor und einer Nvidia GP100 Quadro GPU durchgeführt. Die Berechnungszeiten wurden aus den Log-Dateien von „Software A“ und „Software B“ übernommen, wobei hier die Datei-Ein- und -Ausgabe in der Zeitmessung mit enthalten ist. Für den GPU-beschleunigten Ansatz wurde die Zeit vom Start des Programms bis zur Visualisierung der Simulationsdaten gemessen.

In Abbildung 3 sind die Ergebnisse des Vergleichs mit „Software A“ zusammengefasst. Neben Visualisierung der simulierten Geometrien sind die Charakteristika der Simulation und die Laufzeit angegeben. In „Software A“ war keine GPU-Beschleunigung verfügbar, daher wurde die schnellste parallele CPU-Version (mit 8 Threads) verwendet. Mit dem entwickelten GPU-parallelen Ansatz wird eine Beschleunigung der Simulation um einen Faktor von bis zu 80 erzielt. Die Abweichungen der Maximalwerte des

Betrags der Verschiebung und der von-Mises-Spannung zwischen „Software A“ und der GPU-parallelen Entwicklungen sind jeweils unter einem Prozent.

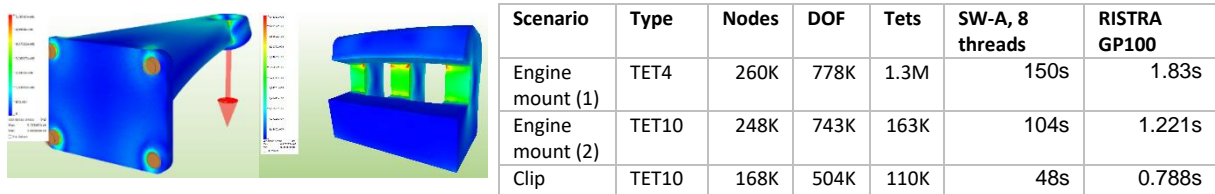


Abbildung 3: Links: Simulationsszenarien mit Visualisierung der von-Mises Vergleichsspannung für den Vergleich mit „Software A“. Tabelle rechts: Charakteristiken der Szenarien mit Elementtyp (lineare (TET4) und quadratische (TET10) Ansatzfunktionen), Anzahl der Knoten und Elemente sowie die Berechnungszeiten von „Software A“ und des neuen Ansatzes in Sekunden.

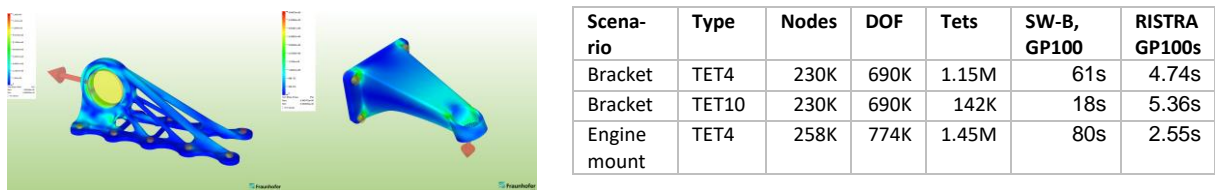


Abbildung 4: Links: Simulationsszenarien mit Visualisierung der von-Mises Vergleichsspannung für den Vergleich mit „Software B“ (Das Bracket Modell wurde uns von Airbus Operations GmbH zu Verfügung gestellt). Tabelle rechts: Charakteristiken der Szenarien mit Elementtyp (lineare (TET4) und quadratische (TET10) Ansatzfunktionen), Anzahl der Knoten und Elemente sowie die

Abbildung 4 zeigt in einer Übersicht den Vergleich mit „Software B“. Neben Visualisierung der Geometrien sind die Charakteristika der Simulation und die Laufzeit angegeben. In „Software B“ wurde die GPU-Beschleunigung aktiviert. Die Beschleunigungen durch den entwickelten GPU-parallelen Ansatz liegen zwischen einem Faktor von 3 – 31. Es zeigt sich, dass Simulationen mit quadratischen Ansatzfunktionen (TET10) in „Software B“ generell schneller sind als mit linearen Ansatzfunktionen (für alle Szenarien ist die Größe der Gleichungssysteme näherungsweise gleich). Ggf. wird hier ein besserer, spezialisierter Vorkonditionierer eingesetzt.

#### 4 Anwendungen

Durch die erzielten Beschleunigungen wird eine Vielzahl von neuen Anwendungen möglich. In Abbildung 5 ist ein Anwendungsbeispiel skizziert, in dem Auswirkungen von geometrischen Änderungen auf den Spannungsverlauf und andere Simulationsergebnisse in Bruchteilen von Sekunden beurteilt werden können. Durch die direkte Visualisierung können damit verschiedene What-If-Szenarien in kurzer Zeit analysiert werden.

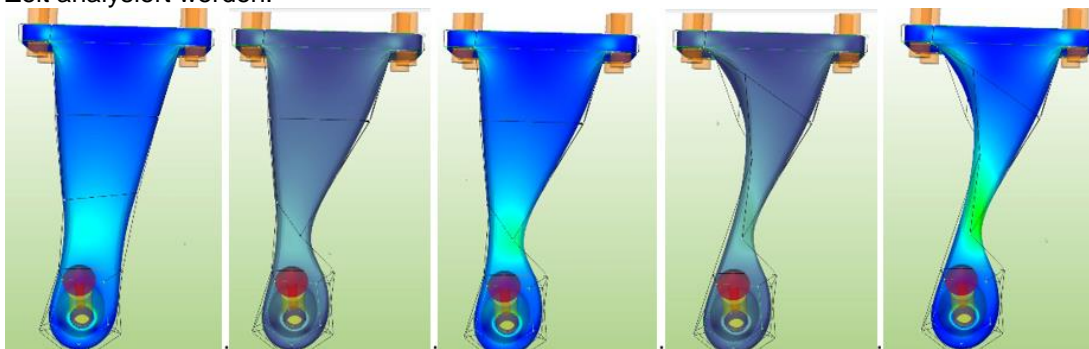


Abbildung 5: Geometriemodifikation mit dem GPU-beschleunigten Löser: Mit spezifizierten Randbedingungen wird die initiale Simulation durchgeführt (erstes Bild). Im zweiten und vierten Bild wird die Geometrie durch Morphing modifiziert – die dunklere Einfärbung weist darauf hin, dass die Simulationsergebnisse ungültig sind. Nach Abschluss der Modifikation (drittes und letztes Bild) sind die neuen Simulationsergebnisse in Bruchteilen von Sekunden verfügbar.

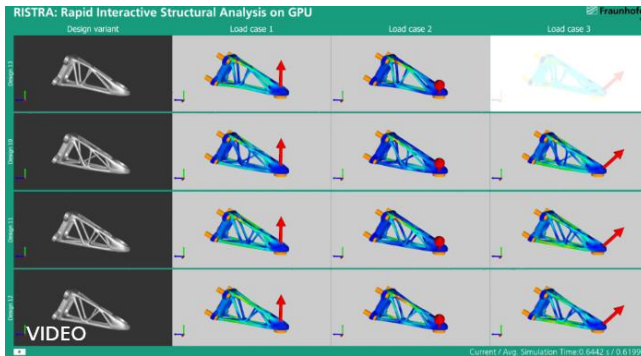


Abbildung 6: Optimierungsszenario mit dem GPU-beschleunigten Löser. Linke Spalte: Variationen einer Motorhaltergeometrie mit unterschiedlichen Knotenpunkten für die Streben. Restliche Spalten: Drei verschiedene Simulationsszenarien werden pro Geometrie ausgewertet. Die Simulationsergebnisse eines jeden Szenarios sind in Bruchteilen von Sekunden verfügbar. Optimierungen können damit signifikant beschleunigt werden.

Abbildung 6 zeigt eine beispielhafte Optimierungsanwendung. Da in solchen Szenarien hunderte bis tausende einzelne Simulationen durchgeführt werden müssen, ergibt sich hier ein immenses Einsparpotenzial.

## 5 Zusammenfassung und Ausblick

Wie in dieser Arbeit vorgestellt, existieren in GPU-beschleunigten Simulationsansätzen in kommerzieller Software noch große Potenziale zur weiteren Reduktion der Berechnungszeit. Essentiell hierbei ist, möglichst einen Großteil von Berechnungen auf der GPU durchzuführen, um die immense Anzahl an CUDA-Cores zu nutzen und ineffiziente Datentransfers zu vermeiden. Beschleunigungen von ein- bis fast zwei Größenordnungen sind so möglich. Interaktive Simulationsszenarien und beschleunigte Optimierungen – manuell oder automatisiert – werden so möglich.

Auch für andere Simulationsarten können GPU-beschleunigte Ansätze einen großen Mehrwert bieten. Für Modalanalysen [5] und Topologieoptimierungen [6] deuten erste Tests auf ebenfalls große Beschleunigungen hin. Des Weiteren ist die Integration mit am Markt verfügbarer Software ein interessanter Aspekt. Nutzer der Software müssen sich nicht in ein neues Softwarepaket einarbeiten, kommen aber dennoch in den Genuss schnellerer Simulationen. Denkbar ist es, mit einer direkten Integration maximale Geschwindigkeitsvorteile zu erzielen oder aber auch bestehende Preprocessor-Formate zu importieren und schneller zu berechnen. Mit der Firma Meshparts und der gleichnamigen Software, in der der GPU-beschleunigte Ansatz für Submodeling eingesetzt werden soll, existiert bereits eine Kooperation.

## 6 References

- [1] Nvidia, NVIDIA A100 Tensor Core GPU, <https://www.nvidia.com/en-us/data-center/a100/> [online, accessed on 2020-08-05].
- [2] Georgescu, S., Chow, P., & Okuda, H. (2013). GPU acceleration for FEM-based structural analysis. *Archives of Computational Methods in Engineering*, 20(2), 111-121.
- [3] GPU-accelerated applications, <https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/tesla-product-literature/gpu-applications-catalog.pdf> [online, accessed on 2020-08-13]
- [4] Ansys Discovery Live, <https://www.ansys.com/de-de/products/3d-design/ansys-discovery> [online, accessed on 2020-08-12].
- [5] David R. Mackay and Kincho H. Law. 1992. An implementation of a generalized Lanczos procedure for structural dynamic analysis on distributed memory computers. Technical Report. Stanford University, Stanford, CA, USA
- [6] Bendsoe, Martin Philip, and Ole Sigmund. *Topology optimization: theory, methods, and applications*. Springer Science & Business Media, 2013.