

# Schlussbericht

---

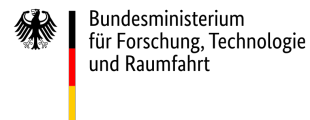
## Verbundprojekt

**StroemungsRaum: Neuartige Exascale-Architekturen mit heterogenen Hardwarekomponenten für Strömungssimulationen**

<b>Zuwendungsempfänger</b>	FAU
<b>Förderkennzeichen</b>	16ME0707
<b>Laufzeit</b>	10/2022 – 12/2025
<b>Teilprojektleitung</b>	G. Wellein, FAU

---

Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.



## **Inhaltsverzeichnis**

<b>1</b>	<b>Kurze Darstellung des Vorhabens</b>	<b>3</b>
1.1	Aufgabenstellung . . . . .	3
1.2	Voraussetzungen zur Durchführung und wissenschaftlicher Stand . .	3
1.3	Planung und Ablauf . . . . .	4
1.4	Zusammenarbeit mit anderen Stellen . . . . .	5
<b>2</b>	<b>Eingehende Darstellung der erzielten Ergebnisse, mit Gegenüberstellung der vorgegebenen Ziele</b>	<b>5</b>
<b>3</b>	<b>Notwendigkeit und Angemessenheit der geleisteten Arbeit</b>	<b>10</b>
<b>4</b>	<b>Wichtigste Positionen des zahlenmäßigen Nachweises</b>	<b>11</b>
4.1	Wirtschaftliche Anschlussfähigkeit . . . . .	11
4.2	Wissenschaftliche-technische Anschlussfähigkeit . . . . .	12
<b>5</b>	<b>Neue projektrelevante FE-Ergebnisse von dritter Stelle</b>	<b>12</b>
<b>6</b>	<b>Erfolgte und geplante Veröffentlichungen</b>	<b>13</b>
<b>7</b>	<b>Sonstige Literatur</b>	<b>13</b>

## 1 Kurze Darstellung des Vorhabens

### 1.1 Aufgabenstellung

Im Projekt “Neuartige Exascale-Architekturen mit heterogenen Hardwarekomponenten für Strömungssimulationen – StroemungsRaum” bestand die zentrale Aufgabenstellung in der Erweiterung der Skalierbarkeit der Anwenderlösung *StrömungsRaum*®. Hierzu sollte der auf der Open-Source-Software FEATFLOW basierende Simulationskern so weiterentwickelt werden, dass hochskalierbare CFD-Simulationen auf zukünftigen Exascale-Architekturen mit heterogenen Hardwarekomponenten effizient durchgeführt werden können.

Im Fokus stand die Entwicklung und Integration neuer algorithmischer Ansätze zur Verbesserung von Skalierbarkeit und Effizienz. Dazu zählten insbesondere Mehrgitterverfahren für Beschleunigerhardware, hochskalierbare nichtlineare Gebietszerlegungsverfahren für CPU- und GPU-Systeme sowie die Anbindung algebraischer Multilevelverfahren zur Behandlung komplexer Geometrien. Ergänzend sollten zeitparallele Verfahren erschlossen werden, um zusätzliche Parallelisierungspotenziale zu nutzen und die Rechenzeiten zu reduzieren.

Ein weiterer Schwerpunkt der Aufgabenstellung lag im konsequenten Co-Design von Methodenentwicklung und Performance Engineering. Ziel war es, sowohl die Performance auf Knotenebene als auch die globalen Kommunikationsstrukturen zu optimieren, um eine effiziente und energieeffiziente Nutzung zukünftiger Exascale-Hardware zu gewährleisten.

Darüber hinaus sollte sichergestellt werden, dass die entwickelten Verfahren unmittelbar in die bestehende Software-as-a-Service-Umgebung integriert und damit für industrielle Anwendungen nutzbar gemacht werden können. Flankierend war der Aufbau von Kompetenzen im Bereich Exascale-Methoden an den beteiligten Forschungseinrichtungen sowie die Stärkung der Zusammenarbeit vorgesehen.

### 1.2 Voraussetzungen zur Durchführung und wissenschaftlicher Stand

Die in diesem Projekt zugrundeliegende Strömungssimulationssoftware FEATFLOW ist eine Open Source Software, die sich primär an modernen mathematischen, numerischen und algorithmischen Techniken für die effiziente Simulation von partiellen Differentialgleichungen orientiert und damit offene Schnittstellen für methodische Techniken bietet, die den aktuellen state-of-art bzgl. numerischer Simulation, Höchstleistungsrechnen und Performance Engineering darstellen. Im Folgenden erläutern

wir die wichtigsten methodischen und algorithmischen Komponenten, die in diesem Teilprojekt erweitert und realisiert werden sollen, und stellen dabei auch den zugehörigen Stand der Wissenschaft und Technik dar.

### 1.3 Planung und Ablauf

Das gesamte Projekt gliedert sich in 4 zentrale Arbeitspakete.

- **AP1 - Methodenentwicklung:** Algorithmische Weiter- und Neueentwicklung von Methoden zur schnelleren und robusteren Lösung von CFD-Problemen, maßgeschneidert für den späteren Einsatz in FEAT und StrömungsRaum und auf modernen Hardware-Architekturen.
- **AP2 - Parallele Implementierung:** Parallele Implementierung der in AP1 entwickelten Verfahren.
- **AP3 - Skalierung und Performance Engineering:** Definition von Benchmarkproblemen (ProxyApps), Skalierbarkeitstests auf modernen Supercomputern und anschließendes sowie begleitendes Performance Benchmarking.
- **AP4 - Integration in FEATFLOW und Validierung in StrömungsRaum:** Integration der parallelen Implementierungen in StrömungsRaum durch schaffung von skalierbaren Software-Interfaces zu FEAT bzw. FEATFLOW.

Die wissenschaftlichen und technischen Beiträge der Projektpartner griffen komplementär ineinander. Die TU Dortmund (TU Do) entwickelte und erweiterte zentrale numerische Kernverfahren, insbesondere im Bereich geometrischer Mehrgittermethoden, zeitsimultaner Newton–Krylov–Mehrgitter-Verfahren sowie Prehandling-basierter Pressure-Poisson-Löser. Die Universität zu Köln (UzK) und die TU Freiberg (TUF) brachten mit FROSch robuste überlappende Schwarz-Gebietszerlegungsverfahren sowie algebraische Mehr-Level-Verfahren ein, die für große und geometrisch anspruchsvolle Fluidprobleme neue Skalierungsoptionen eröffneten. Das Forschungszentrum Jülich trieb (FZJ) die zeitparallelen Integrationsverfahren und deren Portierung auf großskalige HPC-Systeme voran, während FAU Erlangen (FAU) die systematische Performance-Analyse, Proxy-App-basierte Modellierung und hardware-orientierte Optimierung der kritischen Routinen verantwortete. IANUS übernahm die Übersetzung dieser Fortschritte in ein industriell verwertbares Gesamtsystem durch Integration, Testautomatisierung, Benchmarking, Deployment und Bereitstellung in StrömungsRaum®.

Der projektspezifische Mehrwert lag gerade nicht in der bloßen Parallelität dieser Arbeiten, sondern in ihrer konvergenten Zusammenführung. Die im Verbund entwi-

ckelten Verfahren wurden so in eine Plattform eingebettet, in der neue Löser und Laufzeitoptimierungen unter realen Randbedingungen gegen Referenzfälle, Prozessdaten, Ergebnisdarstellungen und Nutzerinteraktionen geprüft werden konnten. Damit entstand ein Entwicklungspfad, in dem jeder Partnerbeitrag nicht nur wissenschaftlich bewertbar, sondern auch hinsichtlich seines operativen Nutzens für Stabilität, Laufzeit, Genauigkeit und Skalierbarkeit des Gesamtsystems messbar wurde.

#### 1.4 Zusammenarbeit mit anderen Stellen

Über die enge Kooperation innerhalb des Konsortiums hinaus gab es im Rahmen dieses Teilprojektes regen Austausch mit den FROSch-Entwicklern im In- und Ausland; siehe <https://shylu-frosch.github.io/team/>, sowie mit der internationalen Sparse-Solver-Community bestehend aus Trilinos-, PETSc- und Hypre-Entwicklern. Insbesondere die derzeitige Integration der RACE-Bibliothek [1] für Cache-Blocking-Techniken für iterative Löser [2] in FROSch erfolgte in Zusammenarbeit mit Alexander Heinlein (Technische Universität Delft, Niederlande).

## 2 Eingehende Darstellung der erzielten Ergebnisse, mit Gegenüberstellung der vorgegebenen Ziele

Das Vorhaben *StroemungsRaum* verfolgte das Ziel, die industrielle Simulation-as-a-Service-Plattform StrömungsRaum<sup>®</sup> durch die systematische Integration exascale-relevanter Methoden, skalierbarer Löseransätze und hardware-naher Performance-Optimierungen substanziell weiterzuentwickeln. Ausgangspunkt war die im industriellen Einsatz etablierte Kopplung der Cloud-Plattform mit dem Simulationskern FEATFLOW. Die wissenschaftlich-technische Herausforderung bestand darin, die in den Teilvorhaben der Verbundpartner entwickelten Verfahren nicht isoliert zu demonstrieren, sondern in eine durchgängig nutzbare, belastbare und für Nicht-HPC-Expertinnen und -Experten zugängliche Anwenderlösung zu überführen. Im Zentrum stand damit nicht allein die Entwicklung neuer Algorithmen, sondern deren reproduzierbare, validierte und produktionsnahe Bereitstellung innerhalb eines Live-Systems.

Die Projektlogik war dabei explizit vertikal angelegt: Methodenentwicklung, parallele Implementierung, Performance Engineering, Systemintegration, Referenzfall-Definition, Benchmarking und Webservice-Bereitstellung wurden nicht als getrennte Ketten, sondern als eng gekoppelte Entwicklungsstufen behandelt. Gerade diese Durchgängigkeit war entscheidend, um wissenschaftliche Fortschritte in einen für Endnutzer unmittelbar wahrnehmbaren Mehrwert zu übersetzen. Der Nutzen des

Verbunds entstand somit wesentlich daraus, dass die Beiträge der Partner nicht auf dem Niveau einzelner Prototypen verblieben, sondern über FEATFLOW in StrömungsRaum<sup>®</sup> integriert, über CI/CD abgesichert und im laufenden System test- und bewertbar gemacht wurden.

Das hier präsentierte Teilprojekt hat zu den Arbeitspaketen und Projektzielen wie folgt beigetragen:

Die FAU war in dem Projekt von zentraler Bedeutung, indem sie die Hardwareeffizienz, Skalierbarkeit und Laufzeitleistung der verschiedenen Softwarepakete in StrömungsRaum<sup>®</sup> systematisch verbesserte. Der Schwerpunkt lag auf dem Arbeitspaket 3, das von der FAU geleitet wurde. Hier wurden zunächst die Leistungsmerkmale wichtiger Softwarekomponenten mithilfe analytischer Modelle analysiert und beschrieben, um Engpässe auf CPU- und GPU-Systemen gezielt zu identifizieren. Auf dieser Grundlage entwickelte die FAU Optimierungsstrategien, implementierte diese als Prototypen und integrierte sie anschließend in den Produktionscode. Darüber hinaus wurden Kommunikationsmuster, Grobgitterverfahren und Ressourcenzuweisungen für zeit- und raumparallele Ansätze untersucht, um die Software auf zukünftige Exascale-Architekturen vorzubereiten.

Es ist bekannt, dass Multigrid-Glättungsverfahren in der Regel den größten Teil der Laufzeit bei Multigrid-Lösungsverfahren beanspruchen. Dies wurde für FEATFLOW während der anfänglichen Laufzeitanalyse bestätigt. Im Zusammenhang mit **AP 3.1** war ein Leistungsprofil der in FEAT(FLOW) verwendeten Glättungsalgorithmen ein logischer nächster Schritt. Aus der Untersuchung der Skalierungsläufe mit MPI in Abbildung 1 geht hervor, dass die in FEATFLOW verwendeten SOR-Glättungsalgorithmen („Successive Over Relaxation“) den Großteil der Zeit bis zu etwa 650 Ränge beanspruchen. Ab diesem Punkt wird die globale Synchronisation zum primären Engpass. Dieser Trend setzt sich bei steigender Anzahl an Rängen fort und bleibt über verschiedene Testfälle und Größen hinweg bestehen. Ein ähnliches Verhalten wurde bei den in FEAT verwendeten Vanka-Glättungsalgorithmen beobachtet.

Durch Entwicklungsaufwand in die Umstrukturierung von MPI\_Wait-Aufrufen sowie das Hinzufügen einer Option für die MPI-Statusprüfung war das Entfernen dieses globalen Synchronisationsengpasses möglich. In Bezug auf **AP 3.3** wurde diese synchronisationsfreie Ausführungsmethode in den Produktionscode integriert. Bei etwa 650 Rängen spart das Deaktivieren der globalen Synchronisation in den Smoothen

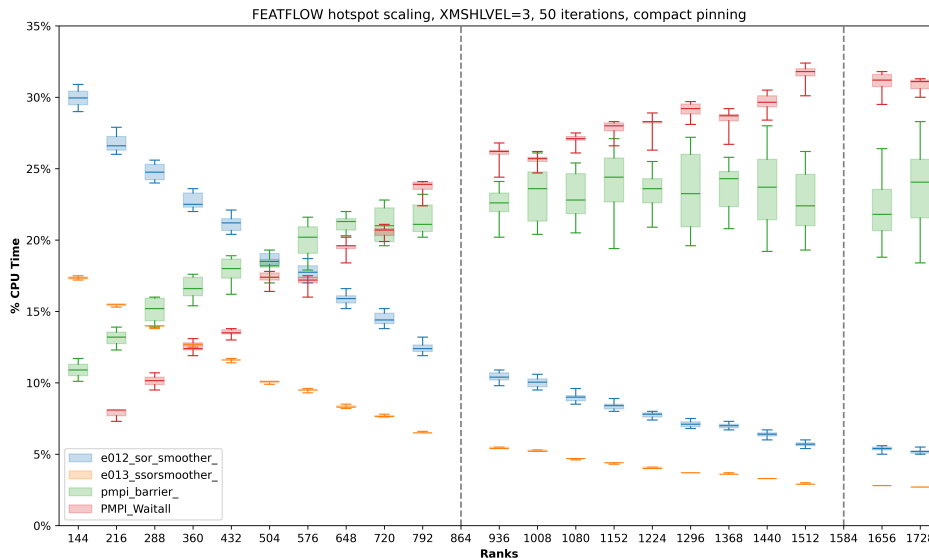


Abb. 1: Hotspots in der FEATFLOW-Laufzeit unter Fritz. Die SOR-Multigrid-Glättungsverfahren sind zunächst der größte Verbraucher an Rechenzyklen, doch nach etwa 650 Rängen werden Routinen im Zusammenhang mit der globalen Synchronisation zum dominierenden Engpass.

etwa 5% der Gesamtlaufzeit ein. Es ist zu erwarten, dass die Performancevorteile bei größeren Prozessorzahlen mit  $O(\log(n))$  weiter ansteigen.

Innerhalb von FEAT wird im Backend für den Block-CRS-Kernel zur Multiplikation von dünnbesetzten Matrizen und dichtbesetzten Vektoren (SpMV) die cuSPARSE-Bibliothek von NVIDIA aufgerufen. Der Navier-Stokes-Löser von FEAT erzeugt aufgrund der Art und Weise, wie die Gleichungen miteinander gekoppelt sind, häufig eine Block-CRS-Systemmatrix mit rechteckigen Blöcken. CuSPARSE dagegen unterstützt nur quadratische Blöcke. Wenn cuSPARSE innerhalb einer Block-CRS-Matrix auf einen rechteckigen Block trifft, füllt es diesen mit Nullen auf, um ihn in einen quadratischen Block umzuwandeln, was letztlich zusätzlichen Datenverkehr verursacht. Im Zusammenhang mit **AP 3.2** haben wir uns vorgenommen, diese leistungskritischen Kernel manuell zu optimieren. Gemeinsam mit TUDo wurden handgeschriebene Block-CRS-SpMV-Kernel in CUDA programmiert, um diese Einschränkung zu überwinden. Abbildung 2 zeigt eine Auswahl dieser Kernel im Vergleich zu cuSPARSE für eine bestimmte 3x1-Block-CRS-Matrix aus FEAT. Zusätzlich experimentierten wir mit dem SELL-C- $\sigma$ -Matrixformat anhand mehrerer Matrizen mit quadratischem Block-Sparsity-Muster aus realistischen Anwendungen, um dessen Eignung als neues

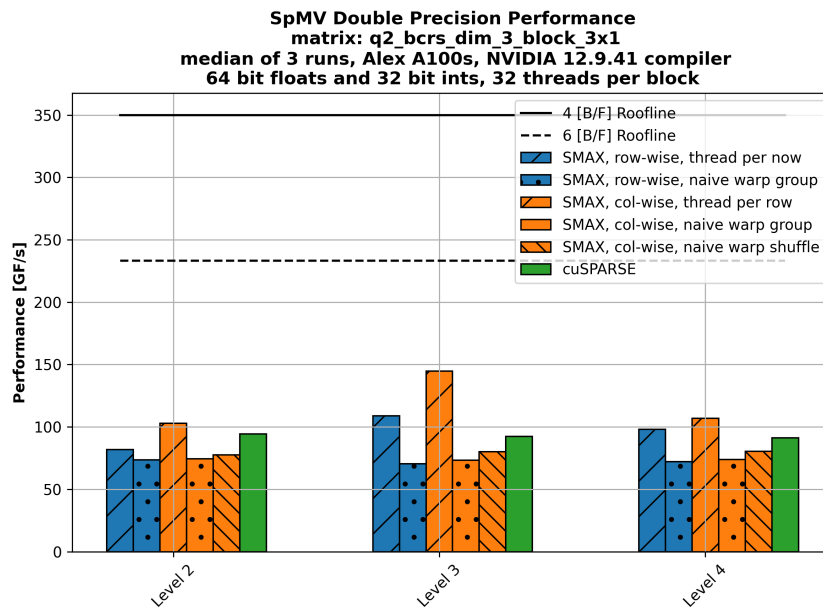
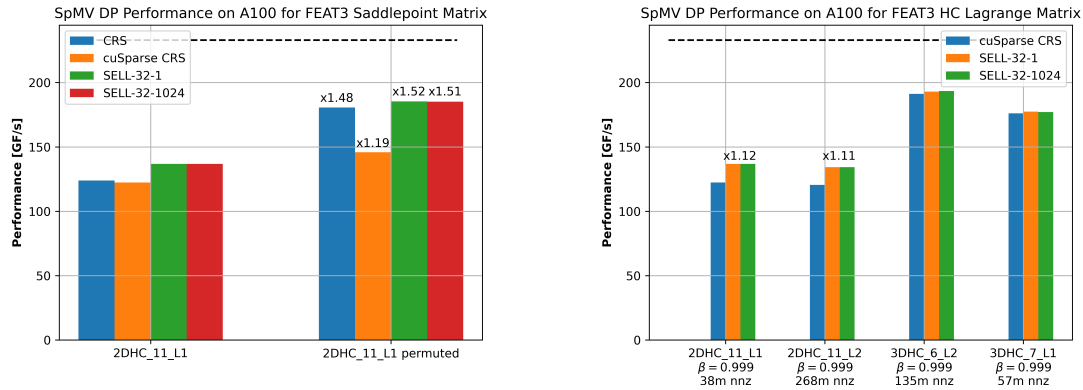


Abb. 2: Leistung verschiedener Block-CRS-SpMV-Kernel. Die horizontalen Linien stellen Schätzungen der Lichtgeschwindigkeitsleistung nach dem Roofline-Modell dar. SMAX bezieht sich auf einen Benchmark-Code für eigenständige Kernel.

Backend-Format für FEAT zu testen. Die Ergebnisse sind in Abbildung 3a und Abbildung 3b zu sehen. Matrixpermutationen führen in der Regel zu einer höheren SpMV-Leistung aufgrund einer besseren Lokalisierung der Datenzugriffe, und SELL-C- $\sigma$  SpMV übertrifft hier auf der A100-GPU cuSparse SpMV.

Im Zusammenhang mit **AP 3.4** wurden die Grobgitterkommunikationsmuster für die Domänendekomposition in FROSch analysiert. Es ist bekannt, dass die Kommunikation mit vielen Rängen zum Engpass in der dreistufigen Schwartz-Methode wird, da unsere Benchmark-Systeme jedoch nur moderate Rangzahlen erreichen konnten, war dieses Problem schwieriger zu erkennen. Allerdings lässt sich bei der Anwendung des „CoarseOperator“ eine potenzielle Nicht-Skalierbarkeit feststellen, wie in Abbildung 4 zu sehen ist. Hier ist zu erkennen, dass eine der drei Phasen der Anwendung des CoarseOperator langsamer skaliert als die beiden anderen Phasen.

Unsere Hypothese lautet, dass dieses Problem auf die doImport/doExport-Implementierungen in Trilinos zurückzuführen ist. Mit ITAC wurde ein potenziell nicht-skalierbares „one-to-many“-Muster von MPI\_Sends beobachtet, das die Netzwerkbandbreite nicht angemessen nutzen konnte. Es wird aktuell daran gearbeitet, diese Teile des FROSch-Codes durch MPI-Collectives zu ersetzen.



- (a) Tests mit der Sattelpunktmatrix von FEAT. Die rechten Balken beziehen sich auf dieselbe Matrix, wobei zu Versuchszwecken eine die Bandbreite reduzierende Breadth-First-Search-Permutation angewendet wurde.
- (b) Tests mit verschiedenen Formen der Lagrange-Matrix von FEAT. Ein  $\beta$ -Wert nahe 1 bedeutet, dass beim SELL-C- $\sigma$ -Format nur sehr wenige Nullen zum Auffüllen verwendet werden.

Abb. 3: SpMV-Leistung bei verschiedenen Formaten für spärliche Matrizen. Die gestrichelte horizontale Linie stellt die „6-B/F-Obergrenze“ aus dem Roofline-Modell dar.

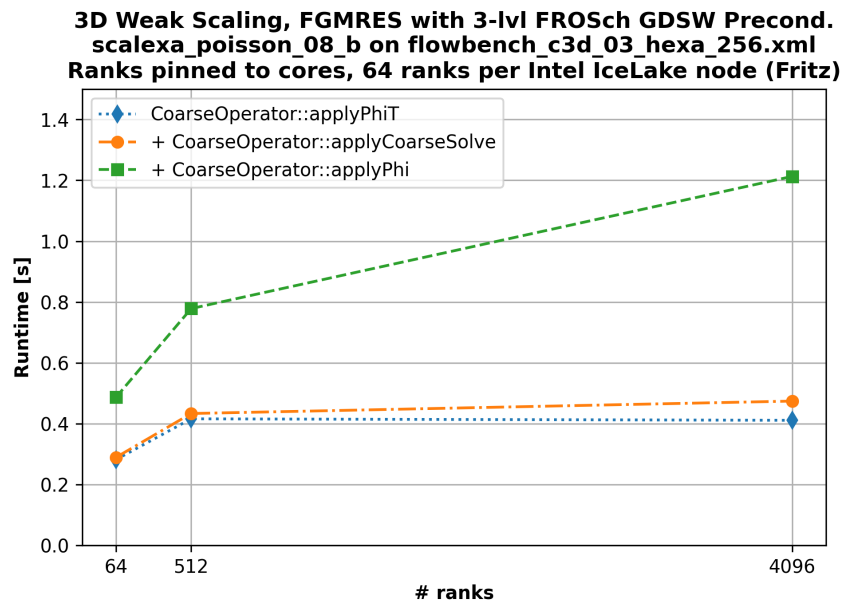


Abb. 4: Skalierungslauf für einen realistischen FROSch-Benchmark. Es ist zu erkennen, dass „applyPhi“ im Vergleich zu den anderen Phasen des „CoarseOperator“ eine gewisse Nicht-Skalierbarkeit aufweist.

Im Rahmen von **AP 3.6** wurden Tensor-Kerne als potenzielle Beschleuniger für die von der TUDo implementierten zeitparallelen Strategien untersucht. Diese Ansätze basieren im Wesentlichen auf der Multiplikation von dünnbesetzten Matrizen mit dichtbesetzten Matrizen (SpMMV). Obwohl diese Operation grundsätzlich für die Nutzung von Tensor-kernen geeignet ist, deutet die Leistungsmodellierung des SpMMV-Kernels [3] darauf hin, dass die Maschinenbalance (das Verhältnis der Hauptspeicherbandbreite zur Spitzen-FLOPS-Rate) aktueller GPU-Architekturen keine signifikante Verbesserung des Durchsatzes erwarten lässt.

### 3 Notwendigkeit und Angemessenheit der geleisteten Arbeit

Das Projektkonsortium hat durch seine Zusammensetzung aus Universitäten, Forschungseinrichtungen mit HPC-Betrieb, Nutzenden von HPC-Software und einem KMU eine interdisziplinäre Bearbeitung des Forschungsthemas gewährleistet, die aufgrund der hohen Komplexität der Aufgabenstellung und dem sowohl wissenschaftlichen also auch wirtschaftlichen Risiko nur mit Fördermitteln finanzierbar war. Der erhebliche Mehraufwand war nur mit Hilfe zusätzlicher, qualifizierter Mitarbeiter:innen zu bewältigen. In der für das Forschungszentrum Jülich maßgeblichen Programm-orientierten Förderung (PoF) ist dieser Aufwand nicht vorgesehen gewesen, so dass hier eine gesonderte Förderung durch das BMFTR notwendig wurde. Auch der Einstieg eines KMU wie IANUS in dieses Feld war ohne Förderung unmöglich. Insbesondere aus Sicht des Simulationsdienstleistungsgeschäftes ist die ausländische Konkurrenz in Form von generalistischen Softwarelösungen sehr stark. Mit *StrömungsRaum*® wurde zwar ein grundsätzlich ganz anderer Ansatz an den Markt gebracht, da er auf individuellen, stabilen, hochverfügbaren und schnellen Simulationslösungen basiert. Dieser kann aber nur erfolgreich sein, wenn man Vorteile insbesondere durch geringere Stückkosten der Simulationen generieren kann. Europäische Initiativen versuchen zwar, KMU mit Accelerator Programmen wie EIC zu stärken und auch die HPC Infrastruktur stärker in den Vordergrund zu stellen und Industrie mit Datacentern stärker zu vernetzen (z.B. EuroCC). Um deutsche Digitalprodukte in diesem Bereich weiter voranzutreiben, ist aber eine nationale Drittmittelbeteiligung unerlässlich, da diese Instrumente eher nicht auf der wissenschaftlichen Ebene greifen, sondern an Scaleup und Marktdurchdringung nahezu fertiger Lösungen ansetzen.

Die Zusammensetzung des Projektkonsortiums mit seinem klaren Schwerpunkt in Deutschland, mit der Thematik und der konkreten Ausrichtung auf exascale-fähige Strömungssimulation mit industrieller Relevanz ist einzigartig. Eine anderweitige

Förderung etwa durch das BMWÉ war nicht denkbar, da erst in diesem Projekt die entscheidenden Schritte hin zu einer Nutzbarmachung der Technologie für industriell relevante Probleme erreicht wurden. Umgekehrt war aufgrund des Umfangs und der Ausrichtung die Einwerbung von DFG-Mitteln in dieser Form nicht realistisch. Allein durch Nutzung bilateraler Förderprogramme (bspw. innerhalb der Helmholtz-Gemeinschaft) war die Erreichung des Gesamtziels nicht realisierbar. Aufgrund der essentiellen deutschen Kompetenzen, der Zielsetzung und der hier vorliegenden technisch-methodischen Tiefen kamen europäische Fördermittel (z.B. durch EuroHPC JU, Horizon Europe oder Centers of Excellence (CoE), aber auch durch etwaige bilaterale Calls bspw. mit Frankreich) ebenfalls nicht in Frage.

#### **4 Wichtigste Positionen des zahlenmäßigen Nachweises**

Dieser Abschnitt enthält Angaben zu den aufgewendeten Personal- und Reisekosten im Verbundvorhaben StroemungsRaum. An der FAU wurden für die Durchführung des Projekts Fördermittel in Höhe von insgesamt 249.876,00 € in Anspruch genommen, die in erster Linie zur Finanzierung zwei wissenschaftlicher Mitarbeiter (Christie Alappat und Dane Lacey). Es wurden insgesamt 36 Personenmonate verbraucht. Der Rest der Fördermittel wurde für Reisen zu und von den Kooperationspartnern sowie für die Teilnahme an Konferenzen und Workshops verwendet.

- Dienstreisen
  - Kick-Off-Meeting, TU Dortmund
  - Verbundprojekt-Statustreffen, TU Dortmund
  - SIAM SCE Konferenz mit Vortrag, Amsterdam, die Niederlande
  - BMBF-Projektstatustreffen, Universität zu Köln
  - Algorithmmy Konferenz mit Vortrag, Bratislava, Slowakei
  - Statustreffen (SCALEXA), TU Dresden
  - Konferenz Supercomputing 2024, Posterpräsentation und Tutorials, Atlanta, USA
  - DKRZ Workshop 2025 mit Vortrag und Diskussionsrunde, Leogang, Österreich
  - M2P Workshop 2025 mit Vortrag und Diskussionspanel, Valencia, Spanien
  - Konferenz ISC 2025 mit Vortrag und Diskussionsrunde, Hamburg
  - Konferenz Supercomputing 2025, Diskussionspanel, St. Louis, USA

##### **4.1 Wirtschaftliche Anschlussfähigkeit**

Die FAU hat keine wirtschaftliche Verwertung geplant.

## 4.2 Wissenschaftliche-technische Anschlussfähigkeit

Das wissenschaftlich-technische Ergebnis des Vorhabens besteht in der erfolgreichen Verbindung von Methodenfortschritten aus der numerischen Mathematik, hochparalleler Implementierung und Performance Engineering mit einer industriell verwertbaren Simulationsplattform. Im Projekt wurden neue bzw. erweiterte Löseransätze für raum- und zeitparallele Strömungssimulationen entwickelt, auf moderne HPC- und GPU-Architekturen übertragen, analysiert und in FEATFLOW sowie StrömungsRaum<sup>®</sup> integriert. Entscheidend ist, dass diese Ergebnisse nicht nur in Form separater Forschungsprototypen vorliegen, sondern in einer funktionsfähigen Toolchain zusammenwirken.

Besonders hervorzuheben ist die erfolgreiche Überführung von Verfahren aus den Bereichen Mehrgitter, Gebietszerlegung, Zeitparallelisierung und Prehandling in ein gemeinsames Gesamtsystem mit kontinuierlicher Integrations- und Testlogik. Auf dieser Grundlage konnten neue Leistungsklassen für industrierelevante CFD-Szenarien erschlossen werden. Die Ergebnisse zeigen damit, dass Exascale-orientierte Methodenentwicklung dann einen hohen Verwertungsgrad erreicht, wenn sie nicht nur auf Rechensystemen skaliert, sondern zugleich über standardisierte Integrationspfade in anwendungsnahe Plattformen übersetzt wird.

Die in diesem Projekt erzielten Ergebnisse wurden auf verschiedenen internationalen Fachtagungen (ECCOMAS, ENUMATH, SIAM, GAMM) vorgestellt und dort diskutiert. Die wesentlichen Neuerungen wurden in verschiedenen wissenschaftlichen Publikationen veröffentlicht, sind eingereicht oder werden vorbereitet.

Die im Bereich Datenstrukturen und Algorithmen gesammelten Erfahrungen fließen sowohl in unsere weithin anerkannten Lehrveranstaltungen und Workshops zum Thema Hochleistungsrechnen (z. B. „Node Level Performance Engineering“ und „Performance Engineering for Linear Solvers“), als auch in die innerhalb der Gruppe entwickelten Softwarebibliotheken (z. B. „Recursive Algebraic Coloring Engine“) ein.

## 5 Neue projektrelevante FE-Ergebnisse von dritter Stelle

Alle für das Projekt im Sinne der Ziele relevanten Ergebnisse wurde innerhalb des Konsortiums erzielt. Es sind keine direkt projektrelevanten Ergebnisse von dritter Seite bekannt.

## 6 Erfolgte und geplante Veröffentlichungen

### Erfolgte Veröffentlichungen

- Cache Blocking of Distributed-Memory Parallel Matrix Power Kernels, D. Lacey, C. Alappat, F. Lange, G. Hager, H. Fehske, and G. Wellein [4]

### Geplante Veröffentlichungen

- **(geplant)** Efficient Inexact Subdomain Solvers for Domain Decomposition Methods with RACE, D. Lacey, C. Alappat, A. Heinlein, F. Alves, and G. Wellein
- **(geplant)** Scalable Algorithms for Exascale Flow Simulation - The StroemungsRaum Approach, alle am Projekt StroemungsRaum beteiligten Wissenschaftler:innen

## 7 Sonstige Literatur

- [1] Christie Alappat, Achim Basermann, Alan R. Bishop, Holger Fehske, Georg Hager, Olaf Schenk, Jonas Thies, and Gerhard Wellein. A recursive algebraic coloring technique for hardware-efficient symmetric sparse matrix-vector multiplication. *ACM Trans. Parallel Comput.* **7** (2020), [link].
- [2] Christie Alappat, Jonas Thies, Georg Hager, Holger Fehske, and Gerhard Wellein. Algebraic temporal blocking for sparse iterative solvers on multi-core cpus. *The International Journal of High Performance Computing Applications* **39**, 230–250 (2025), [link]. <https://doi.org/10.1177/10943420241283828>.
- [3] Moritz Kreutzer. *Performance Engineering for Exascale-Enabled Sparse Linear Algebra Building Blocks*, vol. Reihe B: Medizin (FAU University Press, 2018), [link].
- [4] Dane Lacey, Christie Alappat, Florian Lange, Georg Hager, Holger Fehske, and Gerhard Wellein. Cache blocking of distributed-memory parallel matrix power kernels. *The International Journal of High Performance Computing Applications* **39**, 385–404 (2025), [link]. <https://doi.org/10.1177/10943420251319332>.