

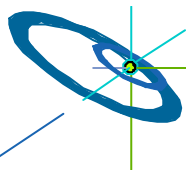
Parallelisierung eines Finite-Volumen-Verfahrens

A. Stock ¹

Universität Stuttgart

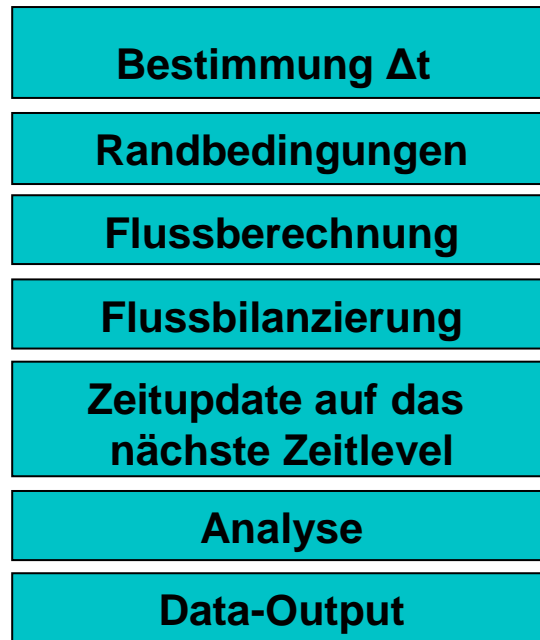
¹Institut für Aerodynamik und Gasdynamik (IAG)

www.iag.uni-stuttgart.de

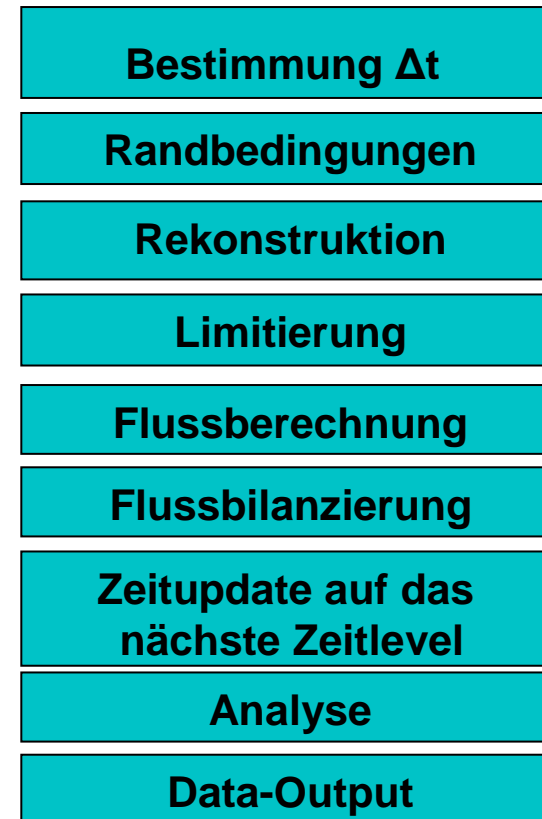


Vergleich der Verfahrensabläufe FV-Verfahren

1. Ordnung



2. Ordnung



Was kann an einem FV-Verfahren parallelisiert werden?

Auf Grund der Lokalität einer Zelle in einem FV-Verfahren werden alle Vorgänge i.d.R. als Schleifen über die Elemente programmiert.

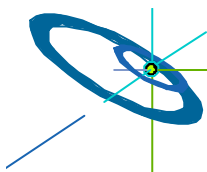
Typische Schleifen:

- Fluss-Berechnung
- Zeitupdate
- Randbedingungen
- Zeitschrittbestimmung
- Daten-Analyse

Einfachste Idee: Aufspaltung der Schleifen in n_procs Teile

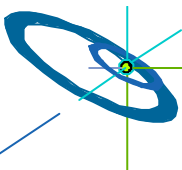
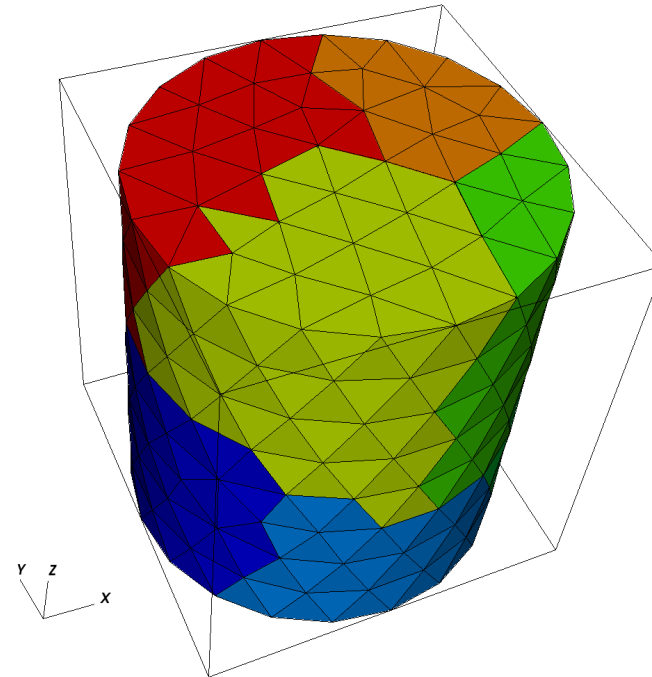
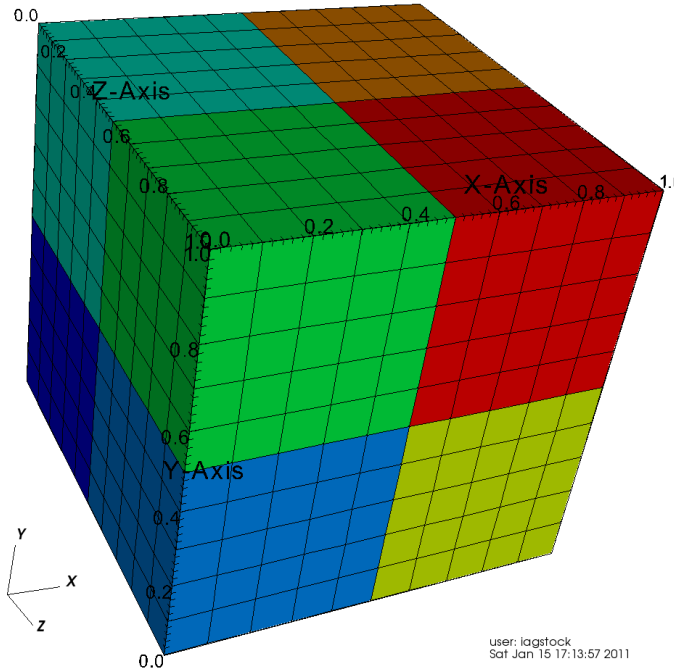
→ Unter Berücksichtigung von Nachbarschaftsbeziehungen der Elemente führt dies zur:

Gebietszerlegung (Domain Decomposition)

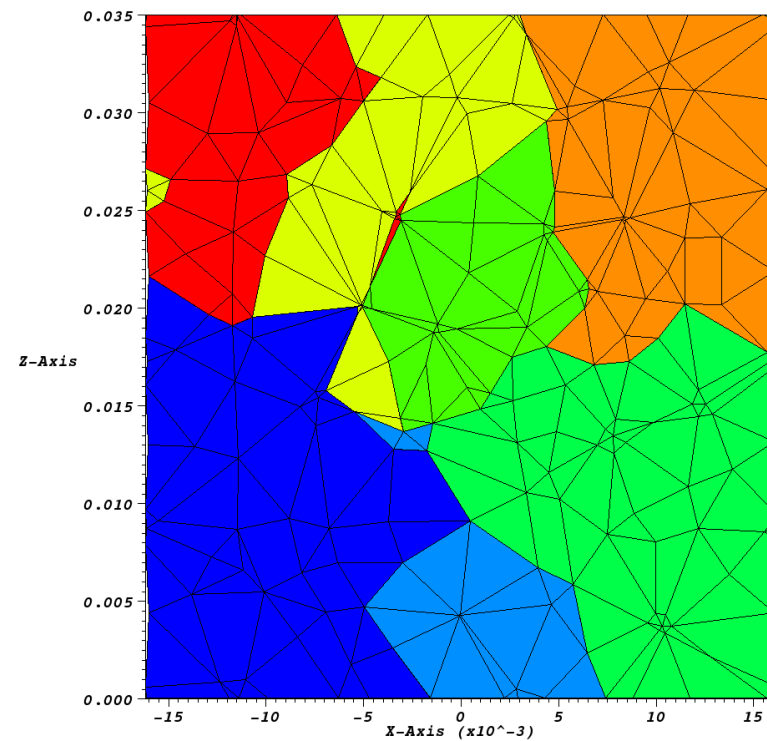
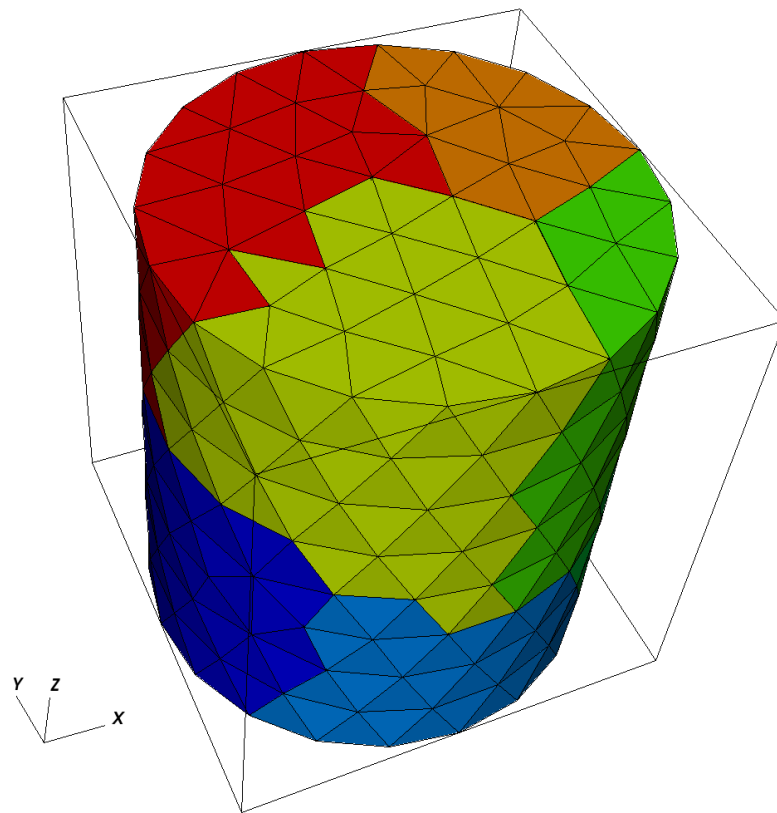


Gebietszerlegung / Domain Decomposition

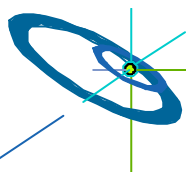
- nur in 2D/3D-relevant
- auf strukturierten Gittern einfach (geometrisch machbar)
- auf unstrukturierten Gittern sehr komplex → Graphen-Theorie (METIS, PARMetis)



Domain Decomposition für unstrukturierte Gitter



Parallelisierung eines Finite-Volumen-Verfahrens [CFD-17]



FV Parallelisierung
[CFD-17] Slide 5 of 10

Roller, Haas, Munz, Dumbser
Universität Stuttgart

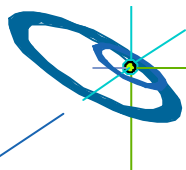
IAF

H L R I S

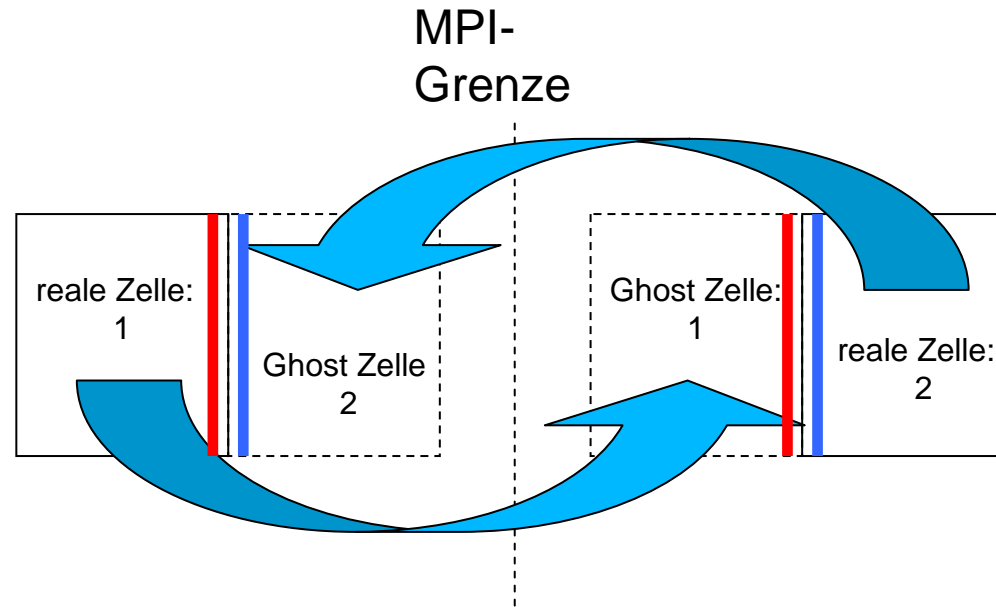


MPI-Parallelisierung eines FV-Verfahren

- skaliert fast beliebig
- komplexe Domain Decomposition
- Kommunikation:
 - An den Domain-Grenzen müssen Zustände auf den Kanten kommuniziert werden.
 - Verwendung von Ghost Zellen
 - An folgenden Stellen im Code muss kommuniziert werden:
 - **vor Fluss-Berechnung**
 - **vor Rekonstruktion**
 - **vor Data-Output**
- I/O, Data-Output, seriell oder parallel

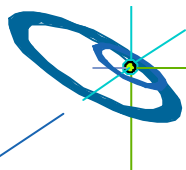


Kommunikation an MPI-Domain-Grenzen (Ghost Zellen)



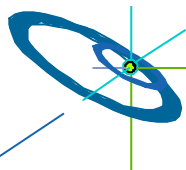
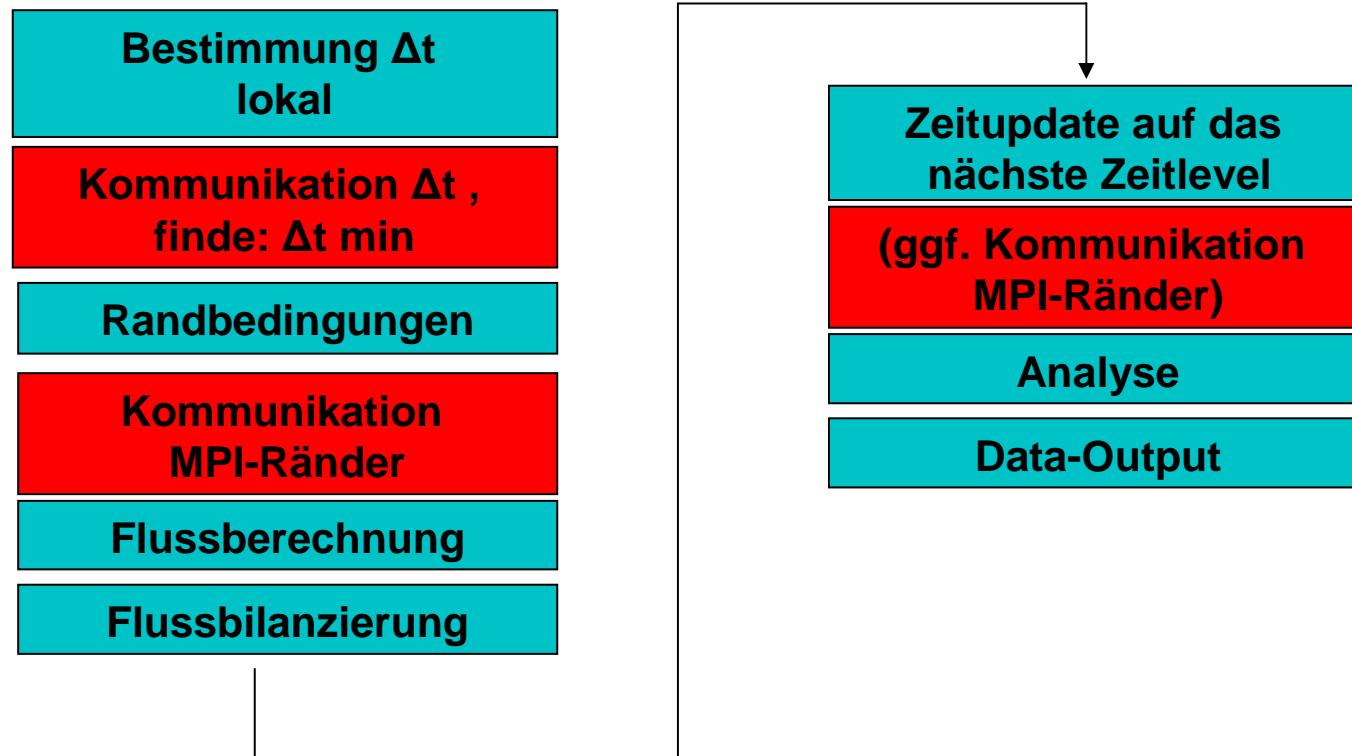
Was wird kommuniziert?

- 1. Ordnung: nur Wert an der Zellkannte
- 2. Ordnung: auch Wert im Bary-Zentrum (für Rekonstruktion)



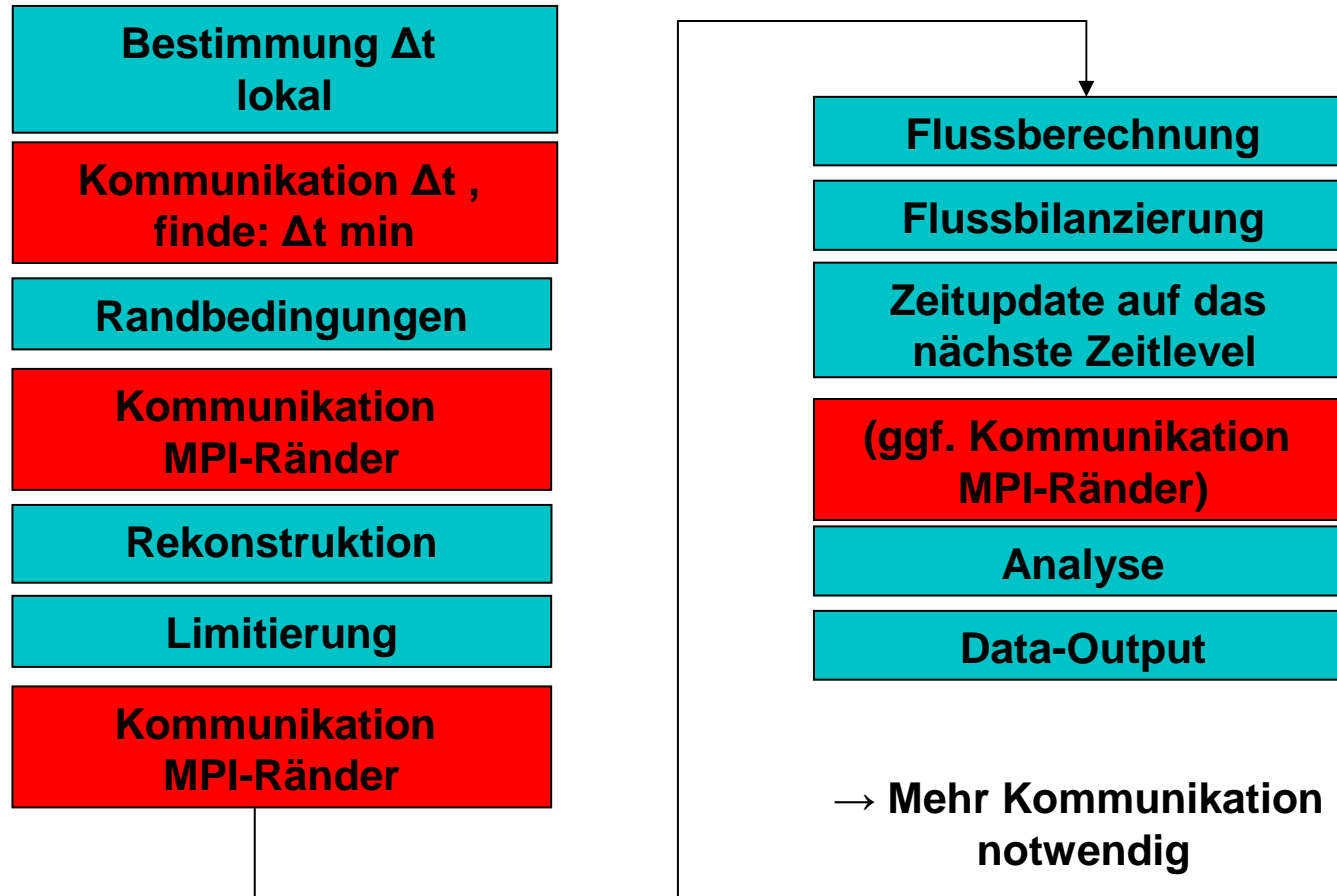
Vergleich der Verfahrensabläufe FV-Verfahren

1. Ordnung



Vergleich der Verfahrensabläufe FV-Verfahren

2. Ordnung



Parallelisierung eines Finite-Volumen-Verfahrens [CFD-17]



Weitere Parallelisierungs-Methoden für FV-Verfahren

Open-MP:

- zum Einstieg eine gute Option um schnell Erfolge zu erzielen
- ohne Gebietszerlegung möglich
- Berücksichtigung der Nachbarschaftsbeziehungen trotzdem notwendig um Race-Conditions zu verhindern
- weitaus geringere Komplexität als MPI

GPU (CUDA, OpenCL):

- extrem komplexe Parallelisierung
- FV-Verfahren schlecht geeignet, da viel Kommunikation notwendig ist
- Anordnung der Daten im Speicher entspricht nicht der geometrischen Anordnung (Problem, da GPU dies besonders ausnutzt)
- bei strukturiertem FV-Verfahren vielleicht noch möglich (jedoch weniger sinnvoll)

