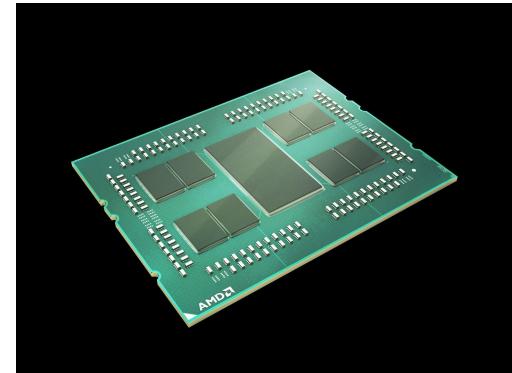
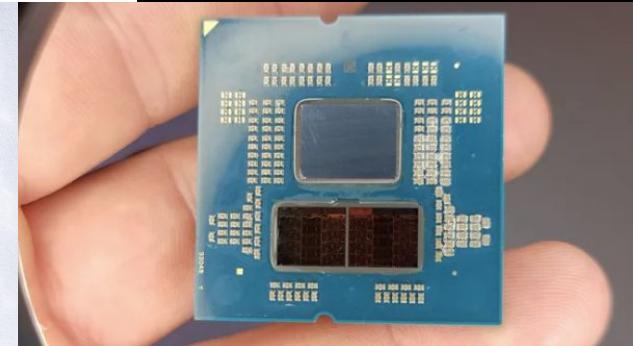
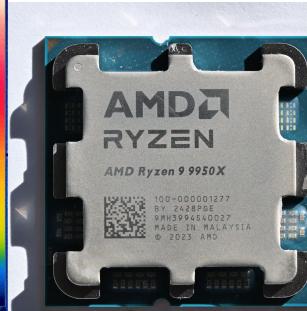
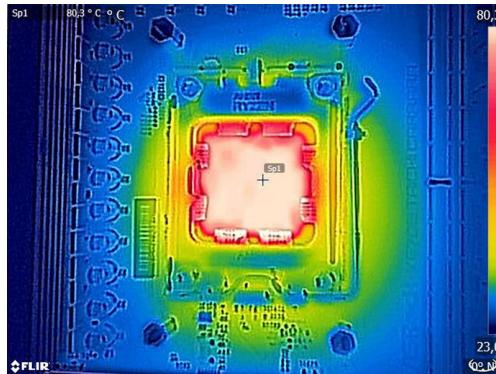


Wärmeleitungsgleichung

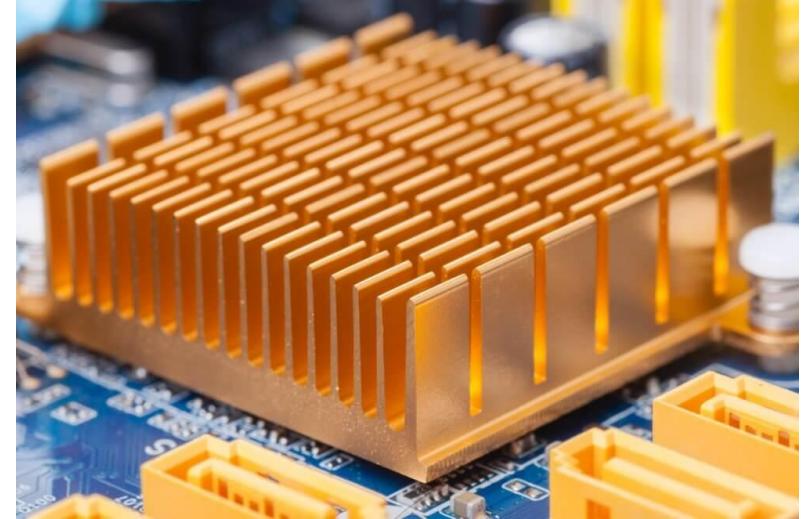
CPU Aufbau

- PCB mit einem oder mehreren Chips (Chiplets)
- Wärmeverteilung über “Heatspreaders”
- Wärme entsteht durch elektrischen Widerstand
- Kühlung ist notwendig für optimale Effizienz und Rechenleistung



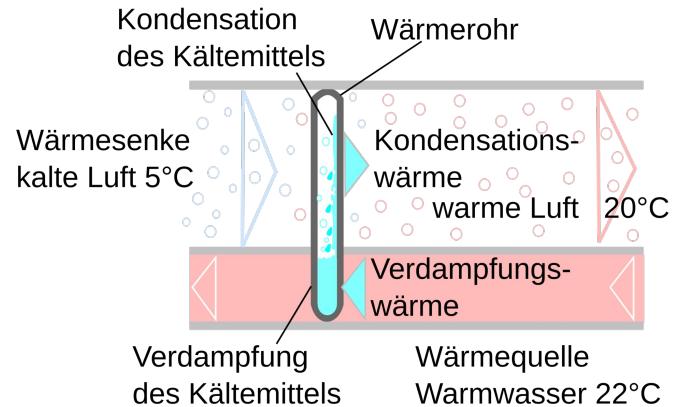
Kühllösungen

- Am häufigsten wird Luftkühlung eingesetzt
- Eine große Oberfläche wird benötigt um Wärme an die Umgebungsluft abzugeben und auszustrahlen
- Besteht meistens vor allem aus Aluminium ($\sim 200 \text{ W/mK}$) oder Kupfer ($\sim 400 \text{ W/mK}$)



Verbesserung der Kühlleistung

- Nutzung von Lüftern zur abkühlung der Umgebungsluft
- Auflagefläche aus Materialien mit höherer Wärmeleitfähigkeit
- “Heatpipes” (bis zu 100 kW/mK) zur besseren Wärmeverteilung auf die Kühllamellen



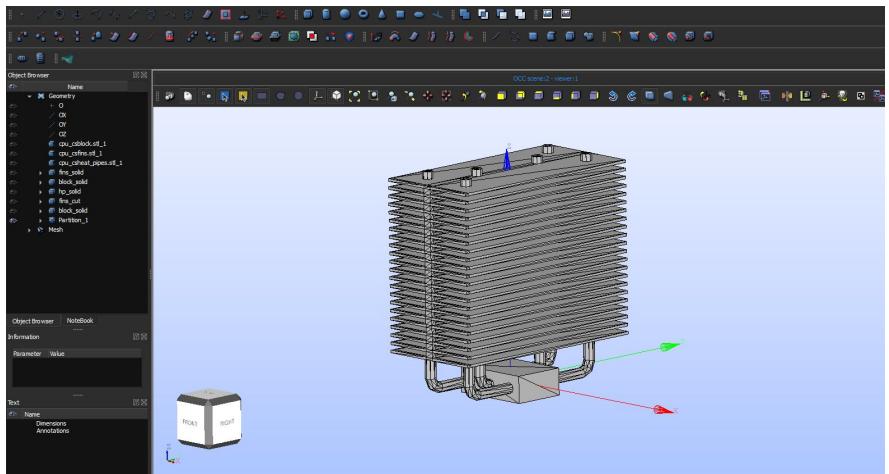
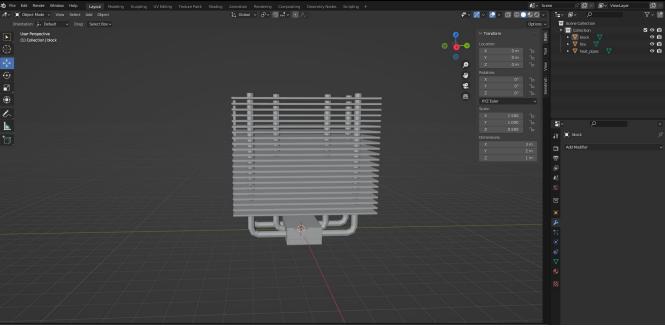
Optimaler Aufbau von Luftkühlungen

- Besteht aus einem Kühlblock (als Auflagefläche), mehreren Heatpipes (zur Wärmeverteilung), Kühllamellen (mit möglichst großer Oberfläche) und einem oder mehreren Lüftern
- Häufige Arten von Luftkühlungen sind Top-Blower und Tower-Kühler



Mesh Erstellung

- Erstellung der Geometrie mittels Blender
- Export als STL-Datei und import als “Shell” in Salome
- Erstellen von Solids aus Shell
- Boolean Cut Operator für Heatpipes
- Partition erstellen -> Physical Groups erstellen -> Meshing



Annahmen

Diffusionstensor als nur gebietsabhängiger isotroper Skalar

$$\underline{D}(t, x, c) \rightarrow D_V, \quad V \subset \Omega$$

Materialparameter als rein gebietsabhängige, isotrope Parameter

$$m(t, x, c) \rightarrow m_V, \quad V \subset \Omega$$

CPU erwärmt gleichmäßig gesamtes Volumen

Nur konvektive Kühlung mit konstanten Koeffizienten, keine Strahlung

Sonst adiabatischer Rand

Konvektion

Über Robin-Randbedingungen

$$ac + b\partial_n c = g(x), x \in \Gamma$$

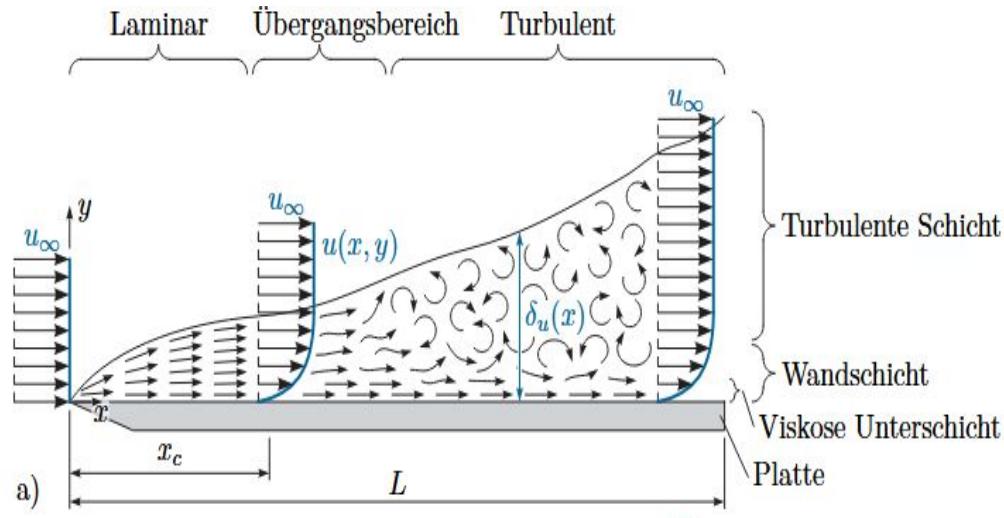
mit Fourier'schem Gesetz ergibt sich:

$$\partial_t q_n = -\alpha(x)(c_\infty - c|_\Gamma)$$

für laminare, stationäre Strömung.

Es gilt:

$$D\nabla c = \frac{\lambda}{\rho c_\rho} \nabla c = -\frac{1}{\rho c_\rho} \partial_t q$$



Bildquelle :

https://www.acin.tuwien.ac.at/fileadmin/cds/lehre/mblg/Modellbildung_Kapitel_3.pdf
zuletzt eingesehen am 19.01.2026

Weak Form

$$\partial_t c = \nabla(D\nabla c)$$

$$\int_{\Omega} \partial_t c v dx = \int_{\Omega} \nabla(D\nabla c) v dx$$

$$\int_{\Omega} \partial_t c v dx = \int_{\Gamma} D\nabla c \cdot n v dS - \int_{\Omega} D\nabla c \nabla v dx = - \int_{\Gamma} \frac{1}{\rho c_\rho} \partial_t q_n v dS - \int_{\Omega} D\nabla c \nabla v dx$$

Mit konvektivem Rand und Quellterm

$$\int_{\Omega} \partial_t c v dx = \int_{\Gamma} \frac{\alpha}{\rho c_\rho} (c_\infty - c) v dS - \int_{\Omega} D\nabla c \nabla v dx + \int_{\Omega} f v dx$$

Zeitintegration

Via Crank-Nicolson Verfahren.

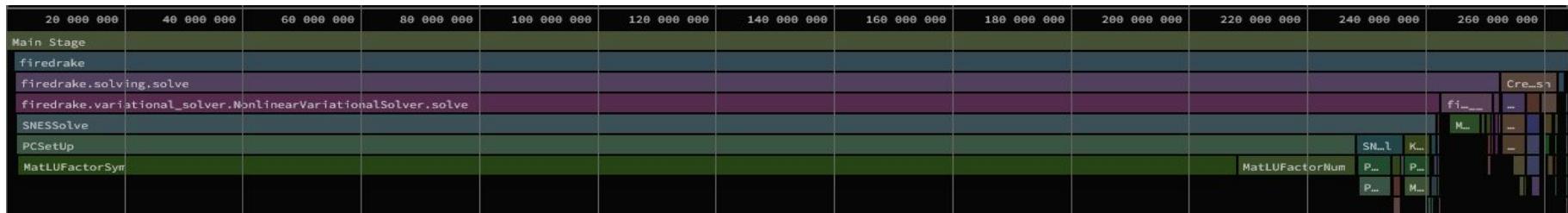
$$u^{n+1} = u^n + \frac{h}{2}(F^n + F^{n+1}) \quad F^n = \nabla(D^n \nabla c^n) + f^n$$

Nach Einsetzen wird die Weak Form gebildet

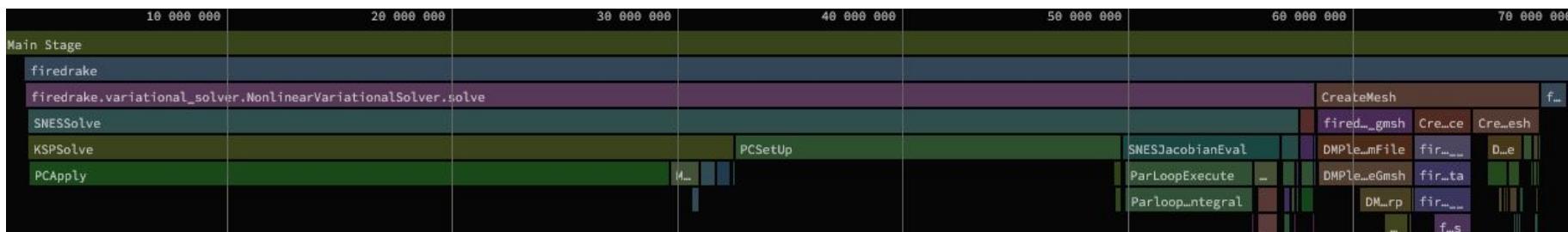
$$\begin{aligned} & \int_{\Omega} c^{n+1} v dx + \frac{h}{2} \int_{\Omega} D \nabla c^{n+1} \nabla v dx + \frac{h}{2} \int_{\Gamma} \frac{\alpha}{\rho c_\rho} c^{n+1} v dS \\ &= \int_{\Omega} c^n v dx - \frac{h}{2} \int_{\Omega} D \nabla c^n \nabla v dx - \frac{h}{2} \int_{\Gamma} \frac{\alpha}{\rho c_\rho} c^n v dS + h \int_{\Gamma} \frac{\alpha}{\rho c_\rho} c_\infty v dS + \frac{h}{2} \int_{\Omega} (f^n + f^{n+1}) v dx \end{aligned}$$

Firedrake-Optimierung

solve() Loop:



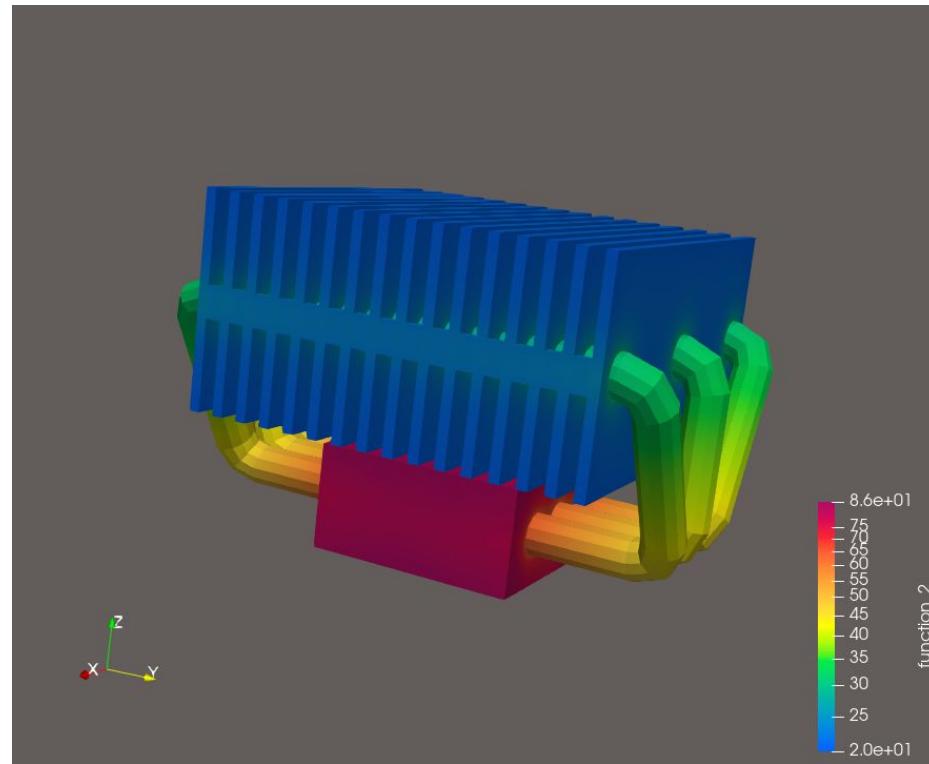
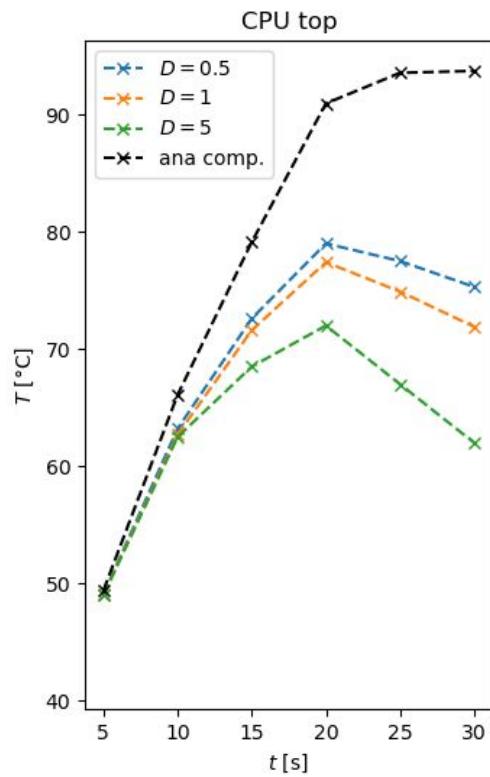
LinearVariationalProblem() + LinearVariationalSolver() + solverparams(CG+hypre):



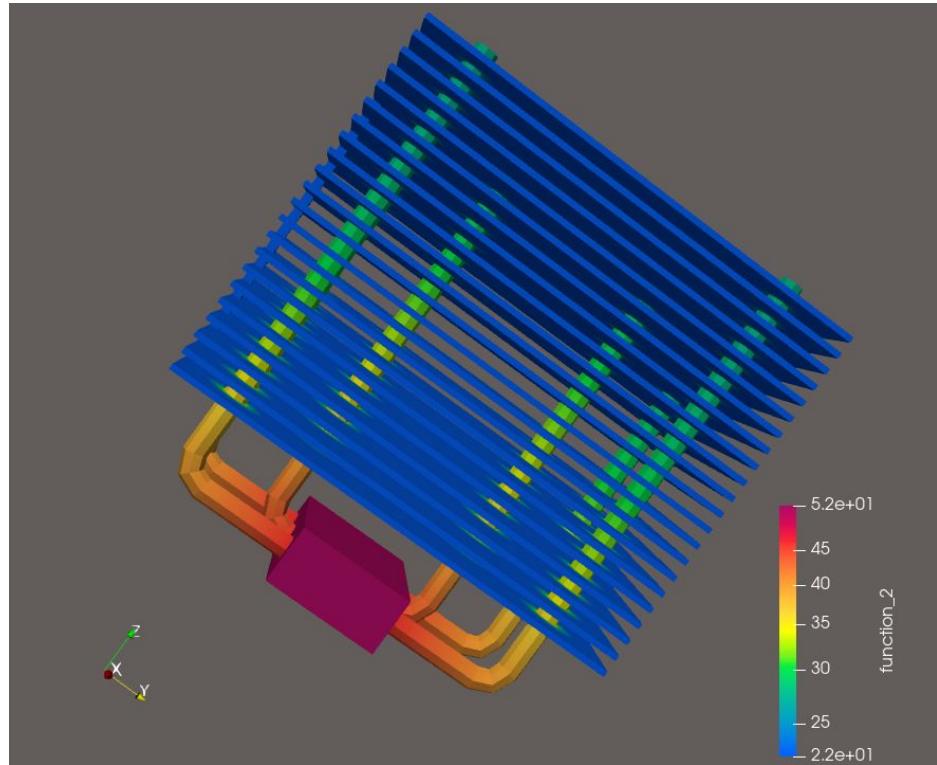
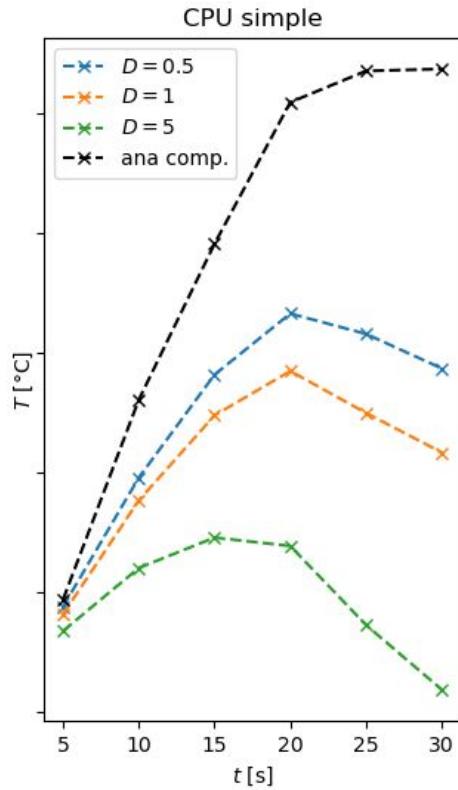
Erstellung: `python3 $1.py -log_view:$1.txt:ascii_flamegraph`

Visualisierung:
<https://www.speedscope.app/>

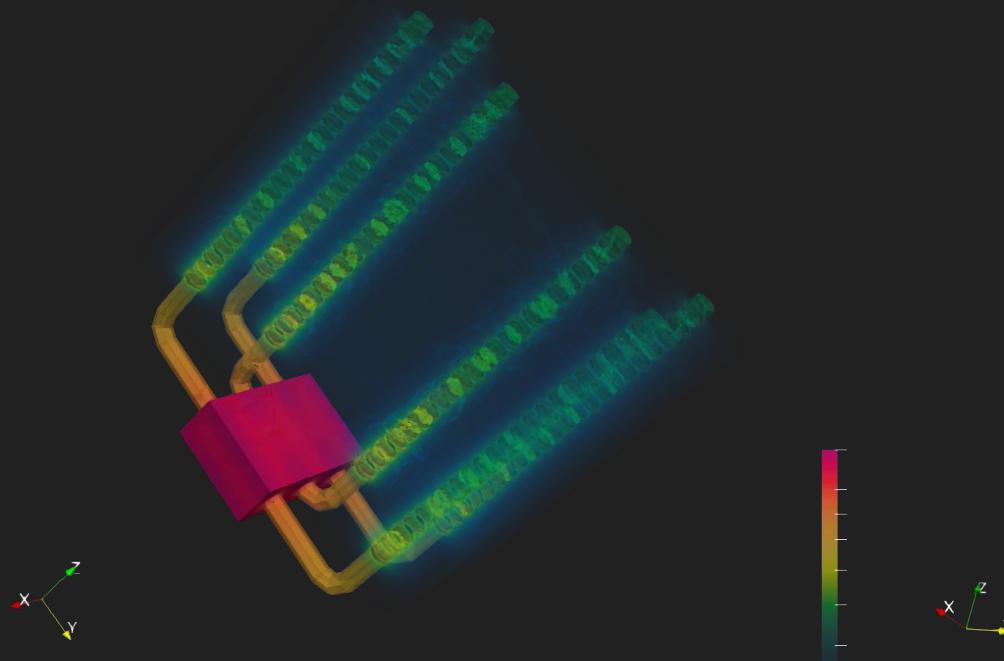
CPU Cooler Top



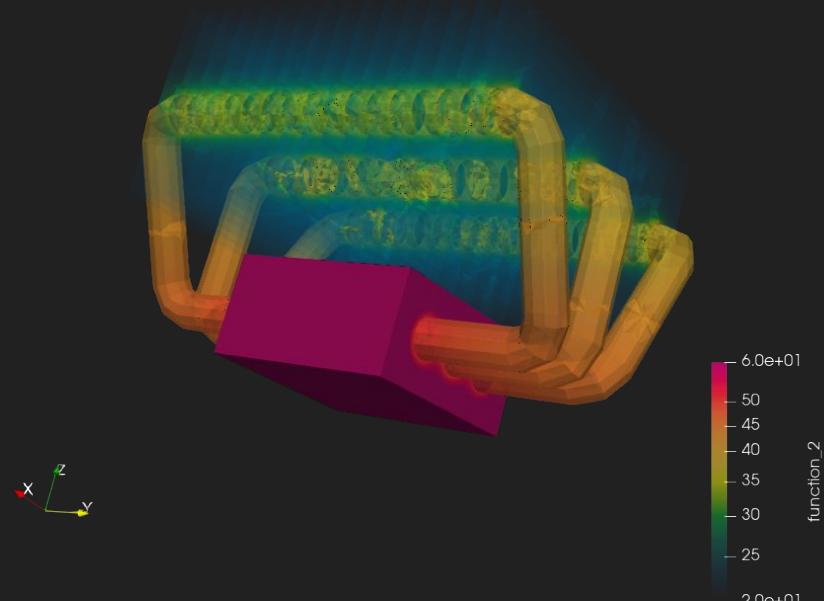
CPU Cooler Simple



Simple Cooler

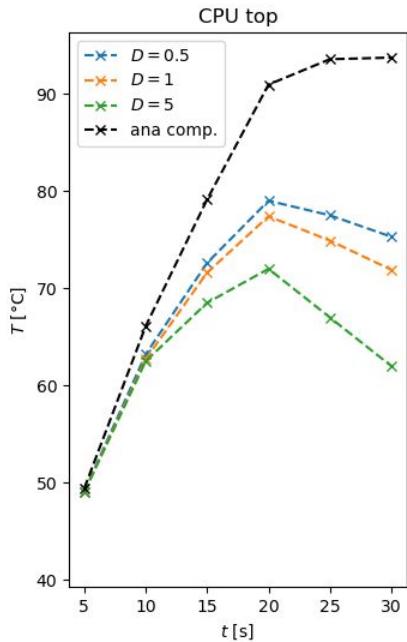


Top Cooler

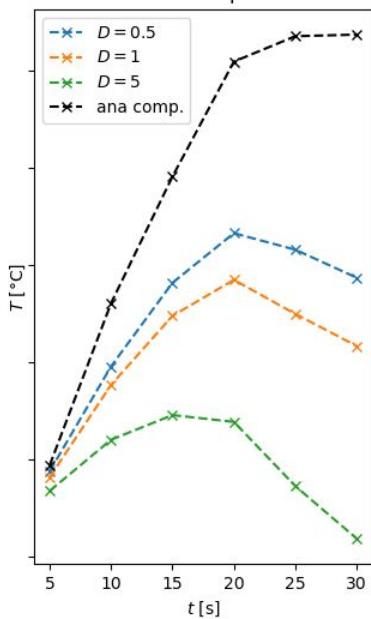


Vergleich

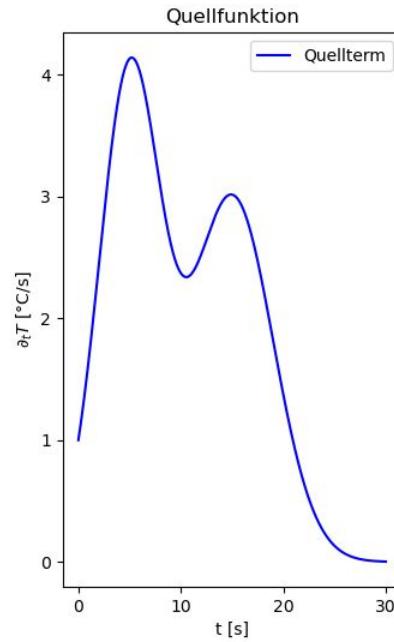
Temperaturmaxima in dem Volumen



CPU simple



Prozessfunktion



Integrierte Temperatur

