

Lösung der stationären Wärmeleitungsgleichung mittels Physics-Informed Neural Networks (PINNs) im Vergleich zur Finite-Elemente-Methode.

★★★★☆	Programmierkenntnisse (Python)	Beginn:	jederzeit
★★★★★	Theorie (Neuronale Netzwerke)	Dauer:	6 Monate
★★★★☆☆	Entwicklung	Sprache:	Englisch/Deutsch
★★★★☆	Forschung		

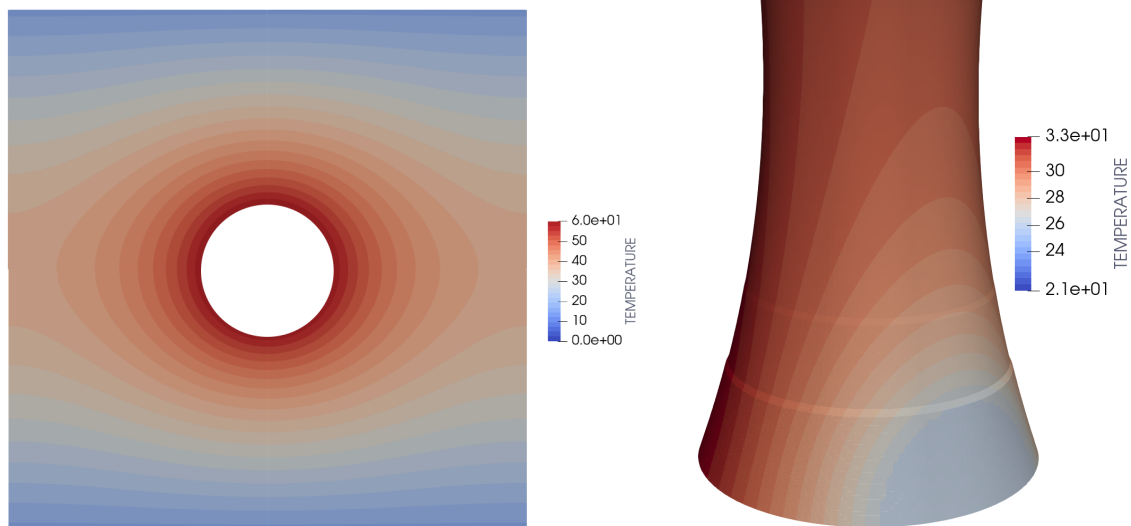


Bild 1: Stationäre Wärmeströmung, Temperaturfeld in einem beheizten Kontinuum (links) und in einem rückseitig sonnenbestrahlten Kühlturm.

Motivation

Die numerische Simulation von Wärmeübertragungsprozessen in technischen Bauteilen basiert klassischerweise auf der Lösung partieller Differentialgleichungen (PDE) mittels etablierter Verfahren wie der Finite-Elemente- oder Finite-Differenzen-Methode. Diese Ansätze liefern hochpräzise Ergebnisse, sind jedoch insbesondere bei komplexen Geometrien, stark nichtlinearen Materialeigenschaften oder zeitabhängigen Randbedingungen mit erheblichem Rechenaufwand verbunden.

Sogenannte physikinformierte neuronale Netze (*Physics-Informed Neural Networks, PINNs*) haben sich in der Vergangenheit als vielversprechende Alternative etabliert. PINNs integrieren physikalisches Vorwissen direkt in den Trainingsprozess, indem die zugrunde liegenden Differentialgleichungen als Nebenbedingungen in die Verlustfunktion eingebettet werden. Dadurch ist es möglich, kontinuierliche Lösungsfelder zu approximieren, ohne auf klassische Gitterdiskretisierungen angewiesen zu sein. Insbesondere für parametervarierte oder echtzeitnahe Vorhersagen bieten PINNs ein großes Potenzial.

Ziel dieses Projekts ist die Implementierung eines einfachen physikinformierten neuronalen Netzes zur Vorhersage der Temperaturverteilung in einem Bauteil. Hierzu soll ein neuronales Netz (z. B. unter Verwendung von Python und PyTorch) entwickelt werden, das die Temperatur als Funktion von Raum und Zeit approximiert. Die Verlustfunktion des Netzes soll neben der Abweichung von gegebenen Anfangs- und Randbedingungen insbesondere einen Term enthalten, der die Verletzung der zugrunde liegenden Wärmeleitungsgleichung bestraft. Die physikalischen Zwangsbedingungen werden dabei über automatische Differentiation direkt aus dem Netzansatz berechnet.

Die Implementierung ist anhand eines einfachen Modellproblems (stationäre Wärmeleitung in einer zwei- oder dreidimensionalen Geometrie) zu verifizieren. Abschließend sollen die Ergebnisse mit einer Referenzlösung aus einer klassischen numerischen Methode verglichen und hinsichtlich Genauigkeit, Stabilität und Rechenaufwand diskutiert werden.

Folgende Arbeitspakete sollen im Rahmen des Projekts bearbeitet werden:

- Einarbeitung in (i) die Grundlagen der Wärmeleitungsgleichung und (ii) in neuronale Netze und Physics-Informed Neural Networks (PINNs). Sie verschaffen sich außerdem einen Überblick über relevante Frameworks (Python, PyTorch) zur KI-Modellierung.
- Modellierung und Analyse von zwei oder drei Referenzmodellen mit Hilfe eines existierenden JAVA-Frameworks.
- Konzept und Implementierung eines Neuronalen Netzwerks mit PyTorch, Formulierung der PDE-Residuen, Nutzung automatischer Differentiation zur Berechnung der Ableitungen und Implementierung der kombinierten Verlustfunktion (PDE, Rand- und Anfangsbedingungen).
- Training und numerische Experimente: Auswahl und Parametrisierung des Optimierers, Trainingsdurchläufe und Konvergenzstudien, Untersuchung des Einflusses von Netzgröße und Trainingspunkten.
- Auswertung der PINN Ergebnisse und Vergleich mit den PDE-basierten Referenzmodellen
- Erstellung und Ausarbeitung einer wissenschaftlichen Abschlussarbeit mit \LaTeX .

Wir bieten

- Regelmäßige Besprechungen und kontinuierliche wissenschaftliche Betreuung während der gesamten Projektlaufzeit.
- Einführung in aktuelle Entwicklungen und Forschungsthemen im wissenschaftlichen Rechnen im Ingenieurwesen.
- Transparente und klar definierte Bewertungskriterien.
- Praxisnahe Software-Entwicklung im Ingenieurwesen auf Grundlage wissenschaftlicher Ergebnisse.

Kontakt

Numerische Mechanik (Prof. Dr.-Ing. habil. Martin Ruess)

e-mail: martin.ruess@hs-duesseldorf.de