

Praktische Übung zur Vorlesung

Einführung in Theorie und Numerik Partieller Differentialgleichungen

WS 2022/23 — Blatt 4

Abgabe: 21.12.2022, via Email an den Tutor.

Aufgabe 1 (Interpolation) (2 Punkte)

Lesen und verstehen Sie die Methode `interpolate()`. Verwenden Sie `plot_sin_function`, um die Interpolation verschiedener Funktionen auf Finite-Elemente-Räume mit unterschiedlichen Auflösungen und Polynomgraden zu untersuchen.

Aufgabe 2 (Jacobi-Matrix) (3 Punkte)

Implementieren Sie die Methode `jacobian()` von `Mesh`. Diese Methode soll die Jacobi-Matrix der Abbildung vom Referenzelement zum physikalischen Element $X \in K_{\text{ref}} \mapsto x \in K$ berechnen.

Hinweis: Sie können Ihre Implementierung mit `py.test test/test_09_jacobian.py` testen.

Aufgabe 3 (Integration) (5 Punkte)

Durch Aufteilen des Integrals über Ω von einer FE Funktion u als Summe über Elemente der Triangulierung \mathcal{T}_h

$$\int_{\Omega} u = \sum_{K \in \mathcal{T}_h} \int_K u,$$

implementieren Sie die Methode `integrate()`, um das Integral von u zu berechnen. Sie können Ihre Implementierung mit `py.test test/test_10_integrate_function.py` testen.

Ihre Methode sollte:

- eine passende `QuadratureRule` definieren;
- auf den Quadraturpunkte `tabulate()` verwenden;
- über jedes Element iterieren;
- `jacobian()` für jedes Element aufrufen und den Betrag ihrer Determinante berechnen (`numpy.absolute` und `numpy.linalg.det()` könnten hier nützlich sein).

Hinweis: Die Funktion `errornorm()` ist sehr ähnlich und kann eine nützliche Schablone für Ihre Arbeit sein.