Wissenschaftliches Rechnen

Aufgabenblatt 4 (Praxis)

wr@cg.tu-berlin.de

WiSe 2023/2024

Allgemeine Hinweise:

- Die Aufgaben sind von jeder/m Studierenden *einzeln* zu bearbeiten und abzugeben (Plagiate werden entsprechend der Studienordnung geahndet).
- Verwenden Sie die vorgegebene Code-Basis. **Sie dürfen keine weiteren Module importieren**. Die zu implementierenden Funktionen befinden sich in der Datei **main.py**. Ihr Code ist an den mit `# TODO: ...` gekennzeichneten Stellen einzufügen. Die `NumPy`-Funktionen, welche Sie zur Lösung einer Aufgabe nicht verwenden dürfen, sind unter `Forbidden` in der Docstring Beschreibung der entsprechenden Funktion aufgelistet.
- Wir stellen einige rudimentäre Unit-Tests zur Verfügung, welche Sie verwenden sollen, um die Funktionalität ihres Codes zu testen. Sie sollten diese Tests während der Implementierung Ihrer Lösung vervollständigen (Funktionalität beschrieben in Python `unittest`). Sie können die Tests mit dem Aufruf `python3 tests.py -v [Tests.test_<function>]` ausführen.
- Bitte reichen Sie die Datei `main.py` mit Ihren Lösungen bis **Montag, den 08.01.2022, um 08:00 Uhr** auf `https://autolab.cg.tu-berlin.de` mit ihren Zugangsdaten ein. Ein mehrfacher Upload bis zum Abgabeende ist möglich. Die letzte Version wird bewertet.

Hinweis: Für alle Aufgaben sind die Interpolationsmatrizen so aufzustellen, dass der höchste Polynomgrad die linke Spalte der Matrix bzw. des Matrixblocks formt und der niedrigste die letzte Spalte. Für ein lineares Interpolationsproblem bedeutet dies zum Beispiel, dass die Vandermonde-Matrix folgende Form hat:

$$V = \begin{pmatrix} x_1 & 1 \\ x_2 & 1 \end{pmatrix} \tag{1}$$

Aufgabe 1: Interpolation der Runge-Funktion (2.5 Punkte)

Die Runge-Funktion ist definiert durch:

$$f(x) = \frac{1}{1+x^2}.$$

In dieser Aufgabe betrachten wir die Interpolation dieser Funktion mit verschiedenen Verfahren.

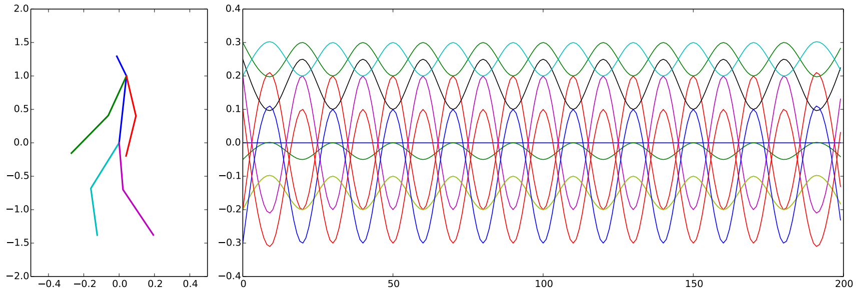
Aufgabe 1.1: Lagrangesche Interpolationsformel (1 Punkte)

Implementieren Sie die Funktion `lagrange_interpolation()`, welche das globale Interpolationspolynom mit Hilfe der Lagrangschen Interpolationsformel bestimmt. Neben dem Interpolationspolynom soll die Funktion zusätzlich die Lagrange Basisfunktionen $\ell_j(x)$ in einer Liste zurückgeben, wobei das k-te Listenelement $\ell_k(x)$ die Basisfunktion für den k-ten x-Wert (aufsteigend sortiert) enthält.

Aufgabe 1.2: Hermite-Interpolation (1.5 Punkte)

Implementieren Sie die Funktion `hermite_cubic_interpolation()`, welche die kubischen Hermite-Interpolationspolynome für gegebene Funktionswerte und Ableitungen an den Stützstellen bestimmt. Die Polynome der einzelnen Abschnitte sollen in einer Liste zurückgegeben werden. Das Polynom des k-ten Abschnitts zwischen den Stützstellen k und k+1 soll an k-ter Position in der Liste stehen.

Aufgabe 2: Keyframe Animation (2.5 Punkte)



Aufgabe 2.1: Kubische Spline Interpolation mit natürlichen Randbedingungen (1.5 Punkte)

Implementieren Sie für die Keyframe Animation die Funktion `natural_cubic_interpolation()`, welche einen kubischen Spline mit natürlichen Randbedingungen berechnet. Die Polynome der einzelnen Abschnitte sollen in einer Liste zurückgegeben werden. Das Polynom des k-ten Abschnitts zwischen den Stützstellen k und k+1 soll an k-ter Position in der Liste stehen.

Aufgabe 2.2: Kubische Spline Interpolation mit periodischen Randbedingungen (1 Punkte)

Implementieren Sie für die Keyframe Animation die Funktion `periodic_cubic_interpolation()`, welche einen kubischen Spline mit periodischen Randbedingungen berechnet. Die Polynome der einzelnen Abschnitte sollen in einer Liste zurückgegeben werden. Das Polynom des k-ten Abschnitts zwischen den Stützstellen k und k+1 soll an k-ter Position in der Liste stehen.