Projekt 1: kubische Spline Interpolation

Simon Stingelin (stiw@zhaw.ch)

13. März 2023

1 Lerninhalt

- Anwenden der kubischen Spline Interpolation in der Ebene.
- Effizientes Lösen von linearen Gleichungssystemen, welche durch Tridiagonalmatrizen gegeben sind.
- Illustrieren und Dokumentieren der Resultate.

2 Ausgangslage

Im vorliegenden Projekt soll die Bahn eines Roboters berechnet werden. Als Beispiel für einen Roboter sei hier ein Staubsauger erwähnt (vgl. Abbildung).

Im File Punkte_Interpolation.xlsx sind die Koordinaten von Punkten in der x-y-Ebene in Abhängigkeit der Zeit im ascii-Format gespeichert. Diese Punkte bilden die Grundlage für die Berechnung der Bahnkurve zur Bewegung eines Roboters. Die Kurve soll durch Interpolation der Punkte ermittelt werden.

Abbildung 1: Staubsauger

3 Kubische Spline Interpolation

Die kubische Spline Interpolation für äquidistante Stützstellen erfordert letzlich das Lösen eines linearen Gleichungssystems (vgl. Skript Numerik, Kapitel 5.1.1) gegeben durch

$$\begin{pmatrix} 4 & 1 & & & 0 \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ 0 & & & 1 & 4 \end{pmatrix} \begin{pmatrix} m_1 \\ \\ \vdots \\ \\ m_{n-1} \end{pmatrix} = \frac{6}{\Delta t^2} \begin{pmatrix} f_0 - 2f_1 + f_2 \\ \\ \vdots \\ \\ f_{n-2} - 2f_{n-1} + f_n \end{pmatrix}, \tag{1}$$

wobei die Spline Interpolierende durch die stückweisen Polynome

$$\begin{aligned} S|_{I_j}(t) =& \frac{(t_{j+1}-t)^3 m_j + (t-t_j)^3 m_{j+1}}{6\Delta t} + \frac{(t_{j+1}-t) f_j + (t-t_j) f_{j+1}}{\Delta t} \\ &- \frac{1}{6}\Delta t((t_{j+1}-t)m_j + (t-t_j)m_{j+1}) \quad j = 0, \ldots n-1. \end{aligned} \tag{2}$$

gegeben ist. Natürliche Randbedingungen implizieren für die beiden Momente m_0, m_n

$$m_0 = m_n = 0.$$

Das System (1) ist durch eine Tridiagonalmatrix gegeben und lässt sich entsprechend sehr effizient mit dem Algorithmus 2.6 im Skript lösen.

4 Aufgaben

1. Studieren Sie im Selbststudium das Kapitel 5.1.1. Leiten Sie das obige System für die Momente her.

2. Laden (`coordinates.txt`) Sie die Punktkoordinaten und die zugehörigen Zeitkoordinaten in ein python/matlab-Skript. Stellen Sie die Punkte in einem Graph dar.

3. Berechnen Sie für die x-Koordinate die Spline. Benutzen Sie dazu den Algorithmus 2.6 ohne unnötige Speichplatzverschwendung.

4. Berechnen Sie die `spline`-Interpolationskurve in der x-y-Ebene und geben Sie diese Kurve mit den gegebenen Stützpunkten graphisch aus. Visualisieren Sie die Bewegung eines punktförmigen Roboters auf dieser Kurve in Zeitschritten $\Delta t = 0.1$s.

5. Berechnen Sie die momentane Geschwindigkeit und Beschleunigung des Roboters und visualisieren Sie diese.

6. **Knacknuss:** Wählen Sie eine Parametrierung der Spline

$$S(\tau) = S|_{I_j}(t(\tau))$$

so, dass die Geschwindigkeit

$$\|\dot{S}(\tau)\| = \left\| \frac{d}{d\tau} S|_{I_j}(t(\tau)) \right\| \equiv 0.2\text{m/s}$$

konstant ist. Gehen Sie davon aus, dass die Koordinaten in cm gegeben sind.

5 Abgabe

Dokumentieren Sie Ihre Arbeit in Form eines Kurzberichts und kommentierten python/matlab Skripts. Die Bewertung erfolgt im Rahmen der auf moodle kommunizierten Bewertungsschema.