

Numerische Mathematik 3

Tutorium 1D FEM - Teil 2

In diesem Teil des Tutoriums beschäftigen wir uns mit zwei wichtigen Themen für die 1D FEM: Der Assemblierung und Einbindung von verschiedenen Randbedingungen.

1. Assemblierung

Was ist die Assemblierung und wozu brauchen wir sie? Das erfährt ihr im [Video III](#).

Lesematerial - Assemblierung

Nun geht es darum den Assemblierungsalgorithmus Schritt für Schritt für unser Beispiel aufzubauen.

[Lese Kapitel 3.4 Seite 93 bis Mitte der Seite 103](#) .

Notiz: Der Assemblierungsalgorithmus 3.1 auf S. 103 lässt sich für beliebige Dimensionen verwenden, nicht nur 1D. Das, was der Algorithmus benötigt, ist die Zuordnung der globalen auf lokalen Knotennummern.

Auf den letzten Seiten haben wir gelernt, wie assembliert wird. Nun können wir es in die Tat umsetzen. Der nächste Schritt ist die Assemblierung der Matrix und der rechten Seite einzuprogrammieren.

Öffne dafür das Skript 'assemblierung.m'.

2. Randbedingungen

Lesematerial - Einbau von Randbedingungen

Als nächstes schauen wir uns an, wie die Randbedingungen eingebaut werden.

[Lese Seite 103 bis Seite 105 bis zum Beispiel](#).

Dir ist klar, wie man die Randbedingungen einbaut? Dann ist es Zeit für eine kurze Selbstüberprüfung:

Selbsttest - Randbedingungen

1. Wie baut man Dirichlet Randbedingungen in das 1D Gleichungssystem ein?
2. Wie baut man Robin Randbedingungen in das 1D Gleichungssystem ein?
3. Wie baut man Neumann Randbedingungen in das 1D Gleichungssystem ein?

Antwortmöglichkeiten:

- a) Durch Veränderung der ersten/letzten Spalte/Zeile der Matrix.
- b) Durch Veränderung am ersten/letzten Diagonaleintrag der Matrix.
- c) Durch Veränderung des ersten/letzten Eintrages der rechten Seite.

Programmieraufgabe:

Nun kennen wir auch den letzten Baustein um den kompletten Algorithmus zusammenzusetzen. Diesen Algorithmus werden wir an Hand des einfachen Beispiels aus dem letzten Tutorium testen:

$$\begin{aligned} -u''(x) &= -2 && \text{fr } x \in I = (0, 1), \\ u(0) &= 0, \\ -u'(1) &= -2 = 1 \cdot (1 - 3) \end{aligned}$$

mit der Lösung $u(x) = x^2$.

Öffne dafür das Skript 'main.m'.

Numerische Mathematik 3

Videotext - Teil 2

Video III Text: Dies ist das dritte Video zum zweiten Teil des Tutoriums 1D FEM.

Um das gesamte Gleichungssystem aufzustellen, müssen wir die ganzen Einträge der Matrix und der rechten Seite ausrechnen. Schauen wir uns dafür einmal einen Matrixeintrag an. Seien i, j fix, dann ist

$$\begin{aligned} K_h[i, j] &= \int_a^b (\varphi_j' \varphi_i' + c \varphi_j \varphi_i)(x) dx + \alpha_b \varphi_j(b) \varphi_i(b) \\ &= \sum_l \int_{I_l} (\varphi_j' \varphi_i' + c \varphi_j \varphi_i)(x) dx + \alpha_b \varphi_j(b) \varphi_i(b). \end{aligned}$$

Wie berechnet man nun so ein Integral möglichst effizient? Wenn wir dieses Integral für jeden Eintrag in der Matrix berechnen, müssen wir dabei immer über n Intervalle integrieren. Und das für $(n+1) \cdot (n+1)$ Einträge. Das ist nicht effizient!

Wie können wir effizienter sein? Nun, dafür schauen wir uns erst einmal die Basisfunktionen an. φ_j ist zwischen x_{j-1} , x_j und x_{j+1} definiert und bewegt sich zwischen 1 und 0. Dahingegen ist φ_{j+1} definiert auf (x_j, x_{j+2}) und φ_{j+2} ist auf (x_{j+1}, x_{j+3}) definiert. Wenn wir nun die Intervalle $I_{j-1}, \dots, I_{j+2}$ einzeichnen, sehen wir auch deutlicher worauf φ_j und φ_{j+2} definiert sind. Wir wissen dass $I_j \cap I_{j+1} = \emptyset$. Damit folgt $\int_I \varphi_j \varphi_{j+2} dx = 0$. Denn entweder ist φ_j oder φ_{j+2} gleich Null. Ihr 'support' schneidet sich nicht. Das gleiche gilt für alle anderen φ_{j+3} , φ_{j+4} und so weiter. Genauso für $\varphi_{j-2}, \varphi_{j-3}$ usw.

Damit folgt für unsere Matrix

$$K_h[i, j] = \begin{cases} \int_{I_j \cap I_i} (\varphi_j' \varphi_i' + c \varphi_j \varphi_i)(x) dx + \alpha_b \varphi_j(b) \varphi_i(b) & \text{für } i \neq j, \\ \int_{I_{j-1} \cup I_j} \left((\varphi_j')^2 + c(\varphi_j)^2 \right)(x) dx + \alpha_b (\varphi_j(b))^2 & \text{für } i = j. \end{cases}$$

Auf jedem Intervall gibt es zwei φ_j 's die nicht gleich 0 sind. Alle anderen sind Null. Also reicht es wenn wir über jedes Intervall gehen und nur den lokalen Beitrag der beiden φ_j berechnen und zu den entsprechenden Einträgen in der Matrix zählen. Jeder Matrixeintrag hat nämlich maximal zwei Teile von zwei Intervallen.

Diesen Prozess nennt man Assemblierung.

Im Lesematerial werden die ganzen lokalen Einträge für unser Beispiel hergeleitet und berechnet. Außerdem erfährt ihr dort mehr über den Assemblierungsalgorithmus, d.h. wie ihr das, was wir gerade angesprochen haben, programmieren könnt. Damit wünsche ich euch: Viel Erfolg!

Numerische Mathematik 3

Antworten - Teil 2

Selbsttest - Randbedingungen

1. a & c
2. b & c
3. c