

Numerische Mathematik 3

Videotext - Teil 1

Video 0 Text: Hallo! Mein Name ist Julia Hauser und ich habe dieses Tutorium zur 1D FEM erstellt. Das Tutorium ist als Einführung zur Implementation der finiten Element-Methode gedacht. Es ist unter der Creative Commons Lizenz CC BY lizenziert.

Nun zum Tutorium. Worum geht es dabei genau? Im ersten Teil werden wir uns mit einer kurzen Wiederholung zu linearer Interpolation und der Vorgehensweise von finiten Element Methoden auseinandersetzen. Im zweiten Teil beschäftigen wir uns mit einer effizienteren Implementierung der finiten Element-Methode und verschiedenen Randbedingungen. Im dritten Teil betrachten wir Konzepte mit denen man die 1D FEM auch auf 2D und 3D übertragen kann.

Alle Teile beinhalten Einführungsvideos und Lesematerial für das Selbststudium. Ihr könnt euch durch kleine Selbsttests und Programmierübungen selber überprüfen um ein Gefühl dafür zu bekommen, wie viel ihr davon verstanden habt. Die Auflösung zu den Tests findet ihr immer am Ende der jeweiligen Teile des Tutoriums.

Alle Lesehinweise sind aus dem Buch "Methode der finiten Elemente für Ingenieure : Eine Einführung in die numerischen Grundlagen und Computersimulation", von Michael Jung und Ulrich Langer aus dem Jahr 2013. Dieses Buch ist in unserer Bibliothek in Onlineform frei zugänglich. Unter dem Video findet ihr den Link zur Seite der Bibliothek, auf der ihr das Buch herunterladen könnt.

Die zur Verfügung gestellten Codeschnipsel sind in Matlab und Octave implementiert und entsprechend gekennzeichnet. Bitte entscheidet euch bereits jetzt ob ihr in Matlab oder Octave programmieren möchtet und verwendet dann den entsprechenden Code. Ebenso findet ihr in dem Tutorium auch immer die entsprechenden Lösungen, sodass ihr euch selbst danach kontrollieren könnt.

Ich wünsche euch viel Erfolg und frohes Lernen bei diesem Tutorium!

Video 1 Text: Dies ist das erste Video zum ersten Teil des Tutoriums 1D FEM.

Also, in diesem Tutorium geht es um die Finite-Element-Methode, F-E-M, in kurz FEM, und was dahinter steckt. Um die Methode zu verstehen, gehen wir alle Schritte anhand eines Beispiels durch. Dieses Beispiel besteht aus zwei Teilen. Der erste Teil ist die Differentialgleichung

$$\begin{aligned} -u''(x) + cu(x) &= f(x) \quad \text{für } x \in I = (a, b), \\ u(a) &= g_a, \\ -u'(b) &= \alpha_b(u(b) - u_b), \end{aligned}$$

wobei c, g_a, α_b, u_b Konstanten sind, also in \mathbb{R} . Das was ihr hier sieht, ist die Differentialgleichung. Zu beachten ist, dass am linken Rand, also bei a , eine Dirichlet-Randbedingung gegeben ist. Während wir am rechten Rand bei b eine Neumann Randbedingung haben.

Zudem gibt es noch eine zweite, sehr wichtige Gleichung: Die dazugehörigen Variationsformulierung. Sie sieht folgendermaßen aus:

Gesucht $u \in V_g := \{u \in H^1(a, b) : u(a) = g_a\}$, sodass

$$a(u, v) = \langle F, v \rangle \quad \forall v \in V_0 := \{v \in H^1(a, b) : v(a) = 0\} \quad (0.1)$$

und

$$a(u, v) := \int_a^b (u'v' + cuv) dx + \alpha_b u(b)v(b),$$

$$\langle F, v \rangle := \int_a^b f v dx + \alpha_b u_b v(b).$$

Wir nehmen nun an, dass die Variationsformulierung eindeutig lösbar ist.

Als dritten Teil des Beispiels haben wir die gesuchte Lösung u .

Nun, wie wir von der Differentialgleichung auf die Variationsformulierung kommen und wie man die eindeutige Lösbarkeit zeigt, das werdet ihr in der Vorlesung lernen. Hier gehen wir davon aus, dass alles lösbar ist und wir uns dazu keine Gedanken machen müssen.

Der zweite Schritt, wie man von einer Variationsformulierung zu einer Lösung kommt. Genauer gesagt der approximierten Lösung. Dieser Schritt benötigt die FEM. Was also hinter diesem Pfeil steckt und wie man eine Lösung berechnet, das lernt ihr in diesem Tutorium.

Video II Text: Dies ist das zweite Video zum ersten Teil des Tutoriums 1D FEM.

Im vorherigen Teil zur linearen Interpolation haben wir gelernt, wie man eine Funktion approximiert. Nehmen wir diese hübsche Funktion beispielsweise her. Sie ist zunächst auf unserem Intervall $I = (a, b)$ definiert. Nun wählen wir uns Stützpunkte x_0 bis x_5 und legen die Intervalle I_0, \dots, I_4 fest. Dabei ist $n = 5$ die Anzahl der Intervalle. Innerhalb dieser Intervalle haben wir die Funktion mittels einer linearen Interpolation approximieren indem wir die jeweiligen Punktauswertungen $u(x_j)$ interpolieren. Wenn diese Intervalle kleiner werden, so wird die Funktion auch besser approximiert.

Solche Funktionen, die stückweise linear und global stetig sind, gehören zu $S_h^1(\mathcal{T})$. Dieser Funktionenraum wird folgendermaßen aufgeschrieben:

$$S_h^1(\mathcal{T}) := \{u \in C(a, b) : u|_{I_j} \in \mathbb{P}_1(I_j) \forall I_j \in \mathcal{T}\},$$

Dabei ist $\mathcal{T} = \{I_j : |I_j| = h_j \text{ und } I_i \cap I_j = \emptyset \forall i \neq j\}$ mit der Eigenschaft dass $\sum_j h_j = |I|$ bzw. $\cup_j I_j = (a, b) = I$. Dieses \mathcal{T} heißt Triangulierung, oder auch Zerlegung genannt. Das heißt also, dass $S_h^1(\mathcal{T})$ aus allen stetigen Funktionen besteht, welche lokal auf jedem Teilintervall lineare Polynome sind. Dieser Funktionenraum $S_h^1(\mathcal{T})$ wird verwendet um $H^1(a, b)$ zu approximieren. Also wir verwenden nun einen endlichen Teilraum um den unendlich dimensionalen Ansatz- und Testraum zu approximieren. Der Ansatz- und Testraum war ja $H^1(a, b)$. Also nehmen wir statt H^1 -Funktionen nun S_h^1 -Funktionen.

Dafür nehmen wir an, dass sich u folgendermaßen darstellen lässt: $u_h(x) = \sum_{j=0}^n u_j \varphi_j(x)$

Zusätzlich wählen wir $v(x) = \phi_i(x)$, also die Basisfunktionen unseres endlichen Teilraumes, und setzen alles in die Variationsformulierung ein. Ges. $u \in V_g$ sodass $a(u, v) = \langle F, v \rangle$ für alle $v \in V_0$.

Lasst uns nun anschauen, wie sich das auf unsere Gleichung auswirkt. Setzen wir in die variationelle Formulierung ein, dann bekommen wir $a(\sum_{j=0}^n u_j \varphi_j, \varphi_i) = \langle F, \varphi_i \rangle$. Ziehen wir die Summe heraus, bekommen wir $\sum_{j=0}^n u_j a(\varphi_j, \varphi_i) = \langle F, \varphi_i \rangle$

Nun fassen wir die Terme zusammen. Wenn wir genau hinschauen, sehen wir, dass wir ein lineares Gleichungssystem vor uns haben. u_j entspricht dem j-ten Eintrag des Vektors \underline{u}_h , $a(\varphi_j, \varphi_i)$ ist der (i,j)-te Eintrag der Systemmatrix K_h und auf der rechten Seite steht der i-te Eintrag des Vektors \underline{f}_h . Also steht nun dort: $K_h \underline{u}_h = \underline{f}_h$. Diese Gleichung können wir lösen.

Somit haben wir uns ein Gleichungssystem hergeleitet, das wir lösen können um eine approximierte Lösung von der Gleichung zu erhalten. Wie gut die Approximation ist, hängt von der Variationsformulierung, der dadurch entstandenen Matrix und dem eigentlichen Interpolationsfehler ab. Dieser

Vorgang wird Diskretisierung genannt. Die Variationsformulierung, in die wir eingesetzt haben, wird diskrete Variationsformulierung genannt.

Die genaue Berechnung der Matrixeinträge findet ihr im Lesematerial. Sobald ihr das gelesen habt, könnt ihr euch an der Programmierung versuchen. In diesem Teil findet ihr ein Programmierbeispiel, indem ihr Codezeilen ausbessern könnt und euch damit selbst überprüfen, ob ihr den Ablauf der FEM verstanden habt. Damit wünsche ich euch: Viel Erfolg!