

FINITA ELEMENTMETODEN I 1D – DEL1 ELLIPTISKA RANDVÄRDESPROBLEM

1

I denna datorlaboration kommer vi att studera finita elementmetoden till stationära värmeledningsekvationen i en dimension

$$\begin{cases} -(a(x)u')' = f(x), & x \in (0, L), \\ u(0) = q_0, \quad u(L) = q_L. \end{cases} \quad (1)$$

1. REDUKTION TILL ETT LINJÄRT EKVATIONSSYSTEM

I detta kapitel repeterar vi hur man kan bestämma en approximativ lösning till en stationär PDE med hjälp av FEM.

Låt oss ta ett intervall $I = (0, L)$ och införa ett jämnt fördelat beräkningsnät som består av N noder $0 = x_1 < x_2 < \dots < x_N = L$, med steget $h = x_{i+1} - x_i$. Vi kommer att bestämma en tillnärmad styckvis affin lösning u_h som tillhör ett ändligt dimensionellt delrum V_h i $V = \{u : \int_0^L (u^2 + (u')^2) dx < \infty\}$:

$$V_h = \{u_h \in C[0, 1] : u_h \text{ är affine för } x \in [x_i, x_{i+1}], \quad 1 \leq i \leq N\}. \quad (2)$$

Testfunktioner ska uppfylla homogena Dirichlets randvillkor och tillhöra delrummet

$$V_{0h} = \{v_h \in C[0, 1] : v_h(0) = v_h(L) = 0 \text{ och } v_h \text{ är affine för } x \in [x_i, x_{i+1}], \quad 1 \leq i \leq N\}. \quad (3)$$

Variationsformuleringen till approximationen lyder:

Bestäm $u_h \in V_h$ så att $u_h(0) = q_0, u_h(L) = q_L$ sådan att

$$\int_0^L a(x)u'_h(x)v'_h(x) dx = \int_0^L f(x)v_h(x) dx, \quad \text{för alla } v_h \in V_{0h}. \quad (4)$$

I V_h väljer vi styckvis affina basfunktioner genom att sätta $\phi_i(x_i) = 1$ och $\phi_i(x_j) = 0$ för $j \neq i$ som illustrerat i Figur 1. Explicita uttryck för basfunktionerna är

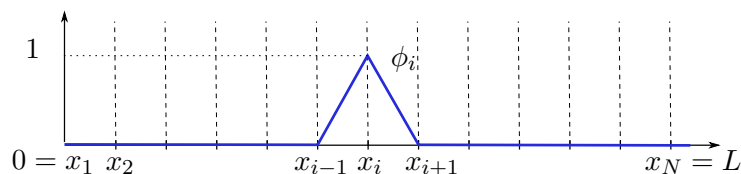


FIGURE 1. Basfunktion $\phi_i(x)$

¹Irina Pettersson, MV CTH/GU

$$\phi_1(x) = \begin{cases} \frac{x_2 - x}{h}, & x_1 \leq x \leq x_2; \\ 0 & \text{för övrigt} \end{cases} \quad (5)$$

$$\phi_j(x) = \begin{cases} \frac{x - x_{j-1}}{h}, & x_{j-1} \leq x \leq x_j \\ \frac{x_{j+1} - x}{h}, & x_j \leq x \leq x_{j+1}, \\ 0 & \text{för övrigt} \end{cases} \quad 2 \leq j \leq N-1; \quad (6)$$

$$\phi_N(x) = \begin{cases} \frac{x - x_{N-1}}{h} & x_{N-1} \leq x \leq x_N. \\ 0 & \text{för övrigt} \end{cases} \quad (7)$$

Vi skriver lösningen till (4) i basen $\{\phi_j\}_{j=1}^N$:

$$u_h(x) = \sum_{j=1}^N u_h(x_j) \phi_j(x).$$

Låt oss ta testfunktionen $v_h = \phi_i$, $2 \leq i \leq (N-1)$ och sätta in i (4):

$$\sum_{j=1}^N u_h(x_j) \int_0^L a(x) \phi_j' \phi_i' dx = \int_0^L f \phi_i dx, \quad 2 \leq i \leq (N-1). \quad (8)$$

Notera att testfunktionen v_h måste uppfylla homogena Dirichlet-villkor vid ändpunkterna, varför vi inte kan ta ϕ_1 och ϕ_N som en testfunktion i (4). På grund av detta fick vi $(N-2)$ ekvationer i (8) och inte N som i Neumans eller Robins problem. Vi kan komplettera (8) med två randvillkor: $u_h(x_1) = q_0$, $u_h(x_N) = q_L$. Eftersom basfunktionen ϕ_i har stöd i $[x_{i-1}, x_{i+1}]$, så är

$$\int_0^L a(x) \phi_j' \phi_i' dx = 0 \quad \text{då} \quad j \neq i-1; i; i+1.$$

Beteckna

$$(K_h)_{i,j} = \int_0^L a(x) \phi_j' \phi_i' dx, \quad i = 2, \dots, (N-1); \quad j = 1, \dots, N;$$

$$(b_h)_i = \int_0^L f \phi_i dx, \quad i = 2, \dots, (N-1).$$

Vi skriver om den svaga formuleringen (8) på matrisform:

$$\begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ (K_h)_{2,1} & (K_h)_{2,2} & (K_h)_{2,3} & \dots & 0 & 0 \\ 0 & (K_h)_{3,2} & (K_h)_{3,3} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & (K_h)_{(N-1),(N-1)} & (K_h)_{(N-1),N} \\ 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix} \begin{pmatrix} u_h(x_1) \\ u_h(x_2) \\ u_h(x_3) \\ \dots \\ u_h(x_{N-1}) \\ u_h(x_N) \end{pmatrix} = \begin{pmatrix} q_0 \\ (b_h)_2 \\ (b_h)_3 \\ \dots \\ (b_h)_{N-1} \\ q_L \end{pmatrix}.$$

För att skriva det sista ekvationssystemet med hjälp av matriser, låt oss införa följande notationer:

$$\begin{aligned}
 (\text{Nodvärden}) \quad & U_h = (u_h(x_i))_{i=1}^N \\
 (\text{Styvhetsmatrisen}) \quad & \begin{cases} (K_h)_{1,1} = (K_h)_{N,N} = 1, \\ (K_h)_{1,j} = 0, \quad j = 2, \dots, N, \\ (K_h)_{i,j} = \int_0^L a(x) \phi'_j \phi'_i dx, \quad i = 2, \dots, (N-1); \quad j = 1, \dots, N \\ (K_h)_{N,j} = 0, \quad j = 1, \dots, N-1, \end{cases} \\
 (\text{Lastvektorn}) \quad & \begin{cases} (b_h)_1 = q_0, \quad (b_h)_N = q_L, \\ (b_h)_i = \int_0^L f \phi_i dx, \quad i = 2, \dots, (N-1). \end{cases}
 \end{aligned}$$

Då kan vi skriva den sista matrisekvationen på kort form

$$\boxed{K_h U_h = b_h.} \tag{9}$$

Systemet (9) kan lösas med hjälp av klassiska metoder i linjär algebra och i MATLAB genom att sätta $U_h = K_h \setminus b_h$. Lägg märke till att vektorn b_h ska vara en kolumnvektor. Lösningen U_h till det sista ekvationssystemet används sedan för att räkna ut den approximativa lösningen

$$\boxed{u_h(x) = \sum_{i=1}^N u_h(x_i) \phi_i(x).} \tag{10}$$

För att uppsumma: du måste lösa ekvationssystemet (9) och sedan räkna ut den approximativa lösningen (10). Formlerna som du kan använda för att räkna ut integralerna finner du i nästa kapitel.

2. KVADRATURFORMLER

Om koefficienten $a(x)$ är komplicerade, så kan vi inte räkna ut integralerna i styvhetsmatrisen K_h explicit. Då ska man använda sig av kvadraturformler för numerisk integration. För våra problem räcker det bra att använda

Trapecformeln: $\frac{1}{x_{i+1} - x_i} \int_{x_i}^{x_{i+1}} f(x) dx \approx \frac{1}{2} (f(x_{i+1}) + f(x_i)).$

För numerisk integration kan du använda material från tidigare kurser. Titta t ex på datorlaborationer som finns på

<http://www.math.chalmers.se/Math/Grundutb/CTH/Matlab/Samhallsbyggnad/ht19/>.

² Basfunktionernas (svaga) derivator är styckvis konstanta:

$$\phi'_1(x) = \begin{cases} -h^{-1} & \text{om } x_1 \leq x < x_2 \\ 0 & \text{för övrigt} \end{cases} \quad (11)$$

$$\phi'_i(x) = \begin{cases} h^{-1} & \text{om } x_{i-1} \leq x < x_i \\ -h^{-1} & \text{om } x_i \leq x < x_{i+1} \\ 0 & \text{för övrigt} \end{cases} \quad i = 2, 3, \dots, N-1. \quad (12)$$

$$\phi'_N(x) = \begin{cases} h^{-1} & \text{om } x_{N-1} \leq x < x_N \\ 0 & \text{för övrigt} \end{cases} \quad (13)$$

Om vi använder trapesformeln för att beräkna integraler, så gäller

$$\begin{aligned} (K_h)_{i,i} &= \int_0^L a(x) \phi'_i \phi'_i dx = \int_{x_{i-1}}^{x_i} a(x) (h^{-1})^2 dx + \int_{x_i}^{x_{i+1}} a(x) (-h^{-1})^2 dx \\ &\approx \frac{a(x_{i-1}) + 2a(x_i) + a(x_{i+1}))}{2h}, \quad i = 2, \dots, (N-1); \end{aligned}$$

$$\begin{aligned} (K_h)_{i,i+1} &= (K_h)_{i+1,i} = \int_0^L a(x) \phi'_i \phi'_{i+1} dx = - \int_{x_i}^{x_{i+1}} a(x) (h^{-1})^2 dx \\ &\approx - \frac{a(x_i) + a(x_{i+1}))}{2h}, \quad i = 1, \dots, (N-1); \end{aligned}$$

$$(K_h)_{1,1} = (K_h)_{N,N} = 1,$$

$$(K_h)_{1,j} = 0, \quad j = 2, \dots, N,$$

$$(K_h)_{N,j} = 0, \quad j = 1, \dots, N-1.$$

På samma sätt kan man räkna ut lastvektorn

$$\begin{aligned} (b_h)_i &= \int_0^L f(x) \phi_i(x) dx = \int_{x_{i-1}}^{x_{i+1}} f(x) \phi_i(x) dx \approx h f(x_i), \quad i = 2, \dots, N-1; \\ (b_h)_1 &= q_0, \quad (b_h)_N = q_L. \end{aligned}$$

3. KONVERGENS

Det är viktigt att kontrollera att den numeriska algoritmen fungerar som den ska. En möjlighet är att använda så kallad “method of manufactured solutions”. Den går ut på att välja en funktion u (en exakt lösning), sätta den in i den ursprungliga ekvationen och räkna ut höger led f samt värden på randen $u(0), u(L)$. Sedan använder man FEM för att bestämma en approximativ lösning u_h och jämför den med u .

²Eftersom basfunktionerna är styckvis affina, så är de inte deriverbara i klassisk mening. Däremot kan man definiera en svag derivata med hjälp av delvis integrationsformeln:

Vi säger att f' är en svag derivata av f om

$$\int_0^L f' v dx = - \int_0^L f v' dx, \quad \forall v \in C^\infty(0, L), v(0) = v(L) = 0.$$

För tillräckligt glatta a och f gäller följande estimat:

$$\|u - u_h\| \leq Ch^2, \quad (14)$$

där normen är $\|g\| = \left(\int_0^L g(x)^2 dx\right)^{1/2}$. Om den numeriska metoden inte fungerar på ett korrekt sätt, så får man kanske h^r med $r \neq 2$. Ett enkelt sätt att se om vi verkligen får rätt potens h^2 är att göra följande

$$\frac{\|u - u_{2h}\|}{\|u - u_h\|} \sim \frac{(2h)^r}{h^r} \sim 2^r \quad \Rightarrow \quad r \sim \log_2 \frac{\|u - u_{2h}\|}{\|u - u_h\|}. \quad (15)$$

Om r som vi får i (15) närmar sig 2 då h minskar, så har vi en anledning för att tro att implementationen av metoden är korrekt.

När koefficienten i ekvationen är diskontinuerlig, som det är i vårt fall, så gäller inte (15) och man kan inte förvänta bättre än $O(h)$, men det gäller att $\|u - u_h\| \leq Ch$.

Man kan illustrera konvergensresultaten genom att halvera steget h och fylla i följande Tabell 1. Ett annat vanligt sätt att studera konvergensthastigheten numeriskt är att plotta $\log h$ mot $\log \|u - u_h\|$ och läsa av grafens lutning.

TABLE 1. Fel och konvergensraten

h	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
fel $\ u - u_h\ $					
$r \sim \log_2 \frac{\ u - u_{2h}\ }{\ u - u_h\ }$					

4. UPPGIFTER

Uppgift 1

Vi vill studera värmeledningsekvationen i en stav gjort av ett kompositmaterial som kommer att bestå av två olika material med olika värmeledningsegenskaper. Låt oss införa en värmeledningskoefficient. Tag den styckvis konstanta funktionen $a(y)$

$$a(x) = \begin{cases} 1, & 0 \leq x < 0.5, \\ 2, & 0.5 \leq x < 1. \end{cases} \quad (16)$$

Betrakta följande Dirichlet randvärdesproblem i en heterogen endimensionell stav:

$$\begin{cases} -(a(x)u'(x))' = 4x, & x \in (0, 1), \\ u(0) = 0, \quad u_\varepsilon(1) = 1. \end{cases} \quad (17)$$

- (1) Bestäm den exakta lösningen $u(x)$ till (17) genom upprepad integration och plotta den i Matlab. Hint: funktionen $u(x)$ måste vara kontinuerlig i $x = 0.5$. Detta ger dig en extra (i tillägg till randvillkor) ekvation för att bestämma okända konstanter.

- (2) Skriv en MATLAB-funktion `[K] = StiffnessMatrix(x)` som räknar ut styvhetsmatrisen K_h (använd trapesformeln för numerisk integration). Funktionen input är noder i beräkningsnätet x . Funktionen output ska vara en gles matris. För värmeledningskoefficienten kan du använda koden i slutet i denna fil.
- (3) Skriv en MATLAB-funktion `[b] = LoadVector(x,q0,qL)` som räknar ut lastvektorn $\mathbf{b}$ beroende på antalet noder N och Dirichlet-villkoren q_0, q_L .
- (4) Använd `[K] = StiffnessMatrix(x)` och `[b] = LoadVector(x, q0, qL)` för att lösa numeriskt (17) samt plotta lösningen $u_h(x)$.
- (5) Jämföra den exakta och numeriska lösningar. Nämligen, räkna ut u_h för flera h och fylla i Tabell 1. Hint: Du kan t ex använda funktionen `trapz` (med ett mindre steg h) för att räkna ut felet.

```
% Take a finer mesh with step=h/8
x1 = 0:h/8:1;
% Interpolate the approximation Uh to get the values
% in all new points
Uh1 = interp1(x,Uh,x1);
% Compute the exact solution u(x) at these points
Uexact = u(x1);
% Compute the L2-error between the exact solution and
% the approximation
ErrorL2 = sqrt(trapz(x1,(Uexact-Uh1).^2))
```

Uppgift 2

Nästa steg är att studera värmeledningen i en stav med snabbt oscillerande värmeledningskoefficient. Tag igen den styckvis konstanta funktionen $a(y)$

$$a(y) = \begin{cases} 1, & 0 \leq y < 0.5, \\ 2, & 0.5 \leq y < 1. \end{cases} \quad (18)$$

Vi utvidgar den periodiskt till hela $\mathbb{R}$ genom att sätta $a(y+1) = a(y)$, $0 \leq y < 1$. T ex $a(3.5) = a(3.5 - 1) = a(2.5 - 1) = a(1.5 - 1) = a(0.5) = 1$. Funktionen $a(y)$ blir då periodisk med perioden 1. Låt $\varepsilon > 0$ vara en (liten) parameter. Funktionen $a^\varepsilon(x) = a\left(\frac{x}{\varepsilon}\right)$ är då en ε -periodiskt oscillerande funktion som beskriver värmeledningsegenskaper hos staven.

Betrakta följande Dirichlet randvärdesproblem i en heterogen endimensionell stav:

$$\begin{cases} -(a_\varepsilon(x)u'_\varepsilon(x))' = 4x, & x \in (0, 1), \\ u_\varepsilon(0) = 0, \quad u_\varepsilon(1) = 1. \end{cases} \quad (19)$$

Eftersom värmeledningskoefficienten $a_\varepsilon(x)$ beror på ε , så beror också lösningen $u_\varepsilon(x)$ på ε , dvs att om vi ändrar ε så ändras både a_ε och lösningen u_ε .

- (1) Plotta $a_\varepsilon(x)$ som en funktion av $x \in [0, 1]$ för $\varepsilon = 2^{-k}$, $k = 0; 1; 2; 3$. Det måste synas tydligt att $a_\varepsilon(x)$ är ε -periodisk.

Hint: eftersom $a(y)$ är 1-periodisk, så gäller $a(\frac{x}{\varepsilon} - \lfloor \frac{x}{\varepsilon} \rfloor)$, där $\lfloor \frac{x}{\varepsilon} \rfloor$ är det största talet mindre än eller lika med x/ε .

För att räkna ut $a_\varepsilon(x)$ kan du sätta `xve = x./ve - floor(x./ve)` (`ve` står för “varepsilon”) för att “skicka” varje x/ε till intervallet $[0, 1]$ och sedan använda funktionen `a(x)` i slutet i denna dokument för att plotta koefficienten:

`plot(x,a(xve),'r-','LineWidth',2).`

- (2) Använd FEM-koden i Uppgift 1 för att lösa (19) numeriskt och plotta lösningen $u_h^\varepsilon(x)$. Istället för $a(x)$ ska du sätta in `a(xve)` i din funktion `StiffnessMatrix(x)`. För att kunna ändra ε , är det enklast att lägga till ε som en input-parameter: `StiffnessMatrix(x, ve)`.
- (3) Då ε blir mindre och mindre, kommer det heterogena materialet som består av två material med värmeledningskoefficienterna $a_1 = 1$ och $a_2 = 2$ blir närmare och närmare (i någon mening) ett homogent material med effektiva värmeledningskoefficienten

$$a^{\text{eff}} = 2(a_1^{-1} + a_2^{-1})^{-1} = 2(1^{-1} + 2^{-1})^{-1} = \frac{4}{3}.$$

Denna formel bevisar vi inte (det kan man göra genom att använda homogeniseringsteori). Man kan tolka den som den effektiva konduktansen för två motstånd kopplade i serie.

Låt oss se att det stämmer numeriskt. För $h = 2^{-8}$ och för $\varepsilon = 2^{-2}, 2^{-3}, 2^{-4}$ plotta $u_h^\varepsilon(x)$ och den explicita lösningen $u_e(x)$ till Dirichlets problemet

$$\begin{cases} -\frac{4}{3}u_e''(x) = 4x, & x \in (0, 1), \\ u_e(0) = 0, \quad u_e(1) = 1. \end{cases} \quad (20)$$

Ser det ut som att u_e approximerar u_h^ε bra då ε blir mindre och mindre?

Man kan definiera värmeledningskoefficienten $a(x)$ i Uppgift 1 på följande sätt:

```
% Coefficient a(x) in Problem 1
% a= 1 if 0<=x<0.5,
% a= 2 if 0.5<=x<1
% Input: vector x
% Output: a(x)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function coeff=a(x)
% Vector k contains all indices such that x<0.5
k0 = find(x < 0.5);
k1 = setdiff(1:length(x),k0);
% Define the piecewise constant function a(x)
coeff = zeros(1,length(x));
coeff(k0) = ones(1,length(k0));
coeff(k1) = 2*ones(1,length(k1));
end
```