STUDIO II

Denna datorlabb är inspirerad av den som Fardin Saedpanah skrev i 2019.
An English version is attached at the end of the document.

1. Problembeskrivning

Denna uppgift går ut på att studera finita elementlösningar för *värmeledningsproblemet*

$$\text{(1)} \qquad \begin{cases} u_t(x,t) - \alpha u_{xx}(x,t) = f(x,t), & 0 < x < 1, \quad 0 < t < T, \\ u_x(0,t) = u(1,t) = 0, & 0 < t < T, \\ u(x,0) = u_0(x), & 0 < x < 1. \end{cases}$$

Här är $u(x,t)$ en okänd temperaturfördelning och $f(x,t)$ och $u_0(x)$ är given källterm, respektive given begynnelsetemperaturfördelning. α är värmeledningskoefficienten, som vi i det följande låter vara 1. (Jämför ekvationen med ekvation (15-17) i Welty et al, Fundamentals of Momentum, Heat and Mass transfer (6th ed.)).

Notera Neumann-randvillkoret vid $x = 0$, $u_x(0,t) = 0$, och det homogena Dirichlet-randvillkoret vid $x = 1$, $u(1,t) = 0$.

2. Variationsformulering

Vårt första steg för att härleda en numerisk metod för att beräkna en approximativ lösning till (1) (med $\alpha = 1$) består av att bestämma en variationsformulering. Vi väljer ett testfunktionsrum $V = \{w\colon [0,1] \to \mathbb{R} \,:\, \|w\|^2_{L^2(0,1)} + \|w'\|^2_{L^2(0,1)} < \infty \text{ och } w(1) = 0\}$, där randvillkoren motsvarar ekvationens randvillkor, dvs inget villkor på testfunktionerna i $x = 0$ där vi har ett Neumann-randvillkor. Vi multiplicerar (1) med en testfunktion $v \in V$ och integrerar över $x \in (0,1)$,

$$\text{(2)} \qquad \int_0^1 u_t(x,t)v(x)\,\mathrm{d}x - \int_0^1 u_{xx}(x,t)v(x)\,\mathrm{d}x = \int_0^1 f(x,t)v(x)\,\mathrm{d}x.$$

Vi använder partialintegration i den andra integralen i (2) för att få

$$\text{(3)} \qquad \int_0^1 u_t(x,t)v(x)\,\mathrm{d}x - [u_x(x,t)v(x)]_{x=0}^{x=1} + \int_0^1 u_x(x,t)v_x(x)\,\mathrm{d}x = \int_0^1 f(x,t)v(x)\,\mathrm{d}x.$$

Eftersom $v(1) = 0$ och $u_x(0,t) = 0$, får vi

$$\text{(4)} \qquad \int_0^1 u_t(x,t)v(x)\,\mathrm{d}x + \int_0^1 u_x(x,t)v_x(x)\,\mathrm{d}x = \int_0^1 f(x,t)v(x)\,\mathrm{d}x.$$

Vår variationsformulering för (1) blir då följande: För varje fixt $t \in (0,T]$, finn $u(\cdot,t) \in V$ så att (4) uppfylls för varje $v \in V$.

Notera att vi har gjort en variationsformulering enbart i rumsvariabeln x. Tiden t betraktas här som en fix parameter.

3. Rumsdiskretisering

Som i Studio I använder vi variationsformuleringen för att göra en rumsdiskretisering. Vi betraktar en partition $\mathcal{T}_h : jh, j = 0, 1, \ldots, m+1$ av intervallet $0 \le x \le 1$ i $m+1$ delintervall av samma längd $h = \frac{1}{m+1}$ och väljer V_h som det underrum till V som består av kontinuerliga funktioner som är styckvis linjära på partitionen $\mathcal{T}_h$. Vi får finita elementformuleringen: För varje fixt $t \in (0,T]$, finn $u_h(\cdot,t) \in V_h$ så att

$$\int_0^1 u_{h,t}(x,t)\chi(x)\,\mathrm{d}x + \int_0^1 u_{h,x}(x,t)\chi'(x)\,\mathrm{d}x = \int_0^1 f(x,t)\chi(x)\,\mathrm{d}x, \qquad \forall \chi \in V_h. \tag{5}$$

Låt $\{\varphi_j\}_{j=0}^m$ vara standardbasen för V_h (det vill säga de vanliga hattfunktionerna; φ_{m+1} inkluderas ej här eftersom $u(1,t) = u_h(1,t) = 0$). Vi kan då skriva u_h som

$$u_h(x,t) = \xi_0(t)\varphi_0(x) + \xi_2(t)\varphi_2(x) + \ldots + \xi_m(t)\varphi_m(x) = \sum_{j=0}^m \xi_j(t)\varphi_j(x). \tag{6}$$

Observera att koefficienterna $\xi_j(t)$ är tidsberoende funktioner, men inte rumsberoende.

Insättning av (6) i (5) och med testfunktioner $\chi = \varphi_i$, för $i = 0, 1, \ldots, m$ ger

$$\int_0^1 \Big(\sum_{j=0}^m \dot{\xi}_j(t)\varphi_j(x)\Big)\varphi_i(x)\,\mathrm{d}x + \int_0^1 \Big(\sum_{j=0}^m \xi_j(t)\varphi_j'(x)\Big)\varphi_i'(x)\,\mathrm{d}x = \int_0^1 f(x,t)\varphi_i(x)\,\mathrm{d}x$$

för $i = 0, 1, \ldots, m$.

En omskrivning ger

$$\sum_{j=0}^m \dot{\xi}_j(t)\Big(\int_0^1 \varphi_j(x)\varphi_i(x)\,\mathrm{d}x\Big) + \sum_{j=0}^m \xi_j(t)\Big(\int_0^1 \varphi_j'(x)\varphi_i'(x)\,\mathrm{d}x\Big) = \int_0^1 f(x,t)\varphi_i(x)\,\mathrm{d}x$$

vilket är ett system av $m+1$ ordinära differentialekvationer (för $i = 0, \ldots, m$) för $m+1$ obekanta funktioner $\{\xi_j(t)\}_{j=0}^m$. På matrisform kan detta skrivas

$$M\dot{\boldsymbol{\xi}}(t) + S\boldsymbol{\xi}(t) = \mathbf{F}(t). \tag{7}$$

Här är $\boldsymbol{\xi}(t)$ en vektor som innehåller nodvärdena $\xi_j(t)$ för $U(x,t)$.

Matriselementen för massmatrisen M och styvhetsmatrisen S är givna av

$$m_{ij} = \int_0^1 \varphi_i(x)\varphi_j(x)\,\mathrm{d}x, \quad s_{ij} = \int_0^1 \varphi_i'(x)\varphi_j'(x)\,\mathrm{d}x, \qquad i,j = 0,1,\ldots,m, \tag{8}$$

som vi redan har beräknat (kapitel 5.3 i boken) och elementen för lastvektorn $\mathbf{F}(t)$ är given av

$$F_i(t) = \int_0^1 f(x,t)\varphi_i(x)\,\mathrm{d}x, \qquad i = 0, 1, \ldots, m. \tag{9}$$

4. Tidsdiskretisering

För att tillämpa finita elementmetoden ovan, återstår det att lösa det *semidiskreta* (diskretiserat i x) problemet $M\dot{\boldsymbol{\xi}}(t) + S\boldsymbol{\xi}(t) = \mathbf{F}(t)$. Vi introducerar en partition $0 = t_0 < t_1 < t_2 < \ldots < t_n = T$ av $[0, T]$ till n delintervall av samma längd $k = T/n$ (så att $t_\ell = \ell k,\ \ell = 0, \ldots, n$) och approximerar tidsderivatan $\dot{\boldsymbol{\xi}}(t)$ med en differenskvot enligt

$$M\frac{\boldsymbol{\xi}^{(\ell)} - \boldsymbol{\xi}^{(\ell-1)}}{k} + S\boldsymbol{\xi}^{(\ell)} = \mathbf{F}(t_\ell), \qquad \ell = 1, \ldots, n \tag{10}$$

där $\boldsymbol{\xi}^{(\ell)} \approx \boldsymbol{\xi}(t_\ell)$ för $\ell = 0, \ldots, n$. Omarrangering av termer ger följande iterativa schema

$$(M + kS)\boldsymbol{\xi}^{(\ell)} = M\boldsymbol{\xi}^{(\ell-1)} + k\mathbf{F}(t_\ell), \tag{11}$$

som också är känd som *Eulers bakåtmetod*[1]. Det finns olika val för data $\boldsymbol{\xi}^{(0)}$, men det enklaste valet är den linjära interpolanten av $u_0(x)$, det vill säga att låta $\xi_j^{(0)} = u_0(x_j)$ för $j = 0, \ldots, m$. Vanligtvis kan inte elementen i lastvektorn beräknas analytiskt utan behöver approximeras genom ett val av kvadraturformel. Man kan använda olika kvadraturregler för att beräkna elementen $F_i(t_\ell)$. Med andra ord startar man med initialdata och en kvadraturformel för $F_i(t_\ell)$, och sedan beräknar man successivt approximationer $\boldsymbol{\xi}^{(1)}$, $\boldsymbol{\xi}^{(2)}$, …, $\boldsymbol{\xi}^{(n)}$ genom att använda schemat (11).

5. Uppgifter

a) Implementera schemat (11) för numerisk lösning av värmeledningsekvationen. För att validera koden, dvs kontrollera att ni gjort rätt, kan ni jämföra med lösningarna som plottats i figur F.1 och F.7 i appendix F i Welty et al, Fundamentals of Momentum, Heat and Mass transfer (6th ed.). Se även ekvation (17-7) på sida 259 och avsnitt 17.2 i Welty et al.

Notera att Neumann-villkoret i (1) motsvarar att vi löser problemet i en halv platta och speglar lösningen till den andra halvan. Dirichlet-villkoret motsvarar att $h \to \infty$ med Weltys beteckningar. Om vi låter $u_0(x) = 1 - x$ och $f(x,t) = 0$ då löser vi samma problem som på sida 259 i Welty för $\alpha = 1$, $x_1 = 1$, $T_\infty = 0$, $T_0 = 1$ och $h \to \infty$. För att jämföra med figurerna gäller alltså att $m = 0$, $Y = U$, $X = t$ och $n = x$.

[1] Man kan också använda andra Finita differensmetoder för tidsdiskretiseringen, se kapitel 6.2 i boken.

Använd er kod för att reproducera figurerna F.1 (för långa tidsintervall) och F.7 (för korta tidsintervall) för $m = 0$. Notera den logaritmiska skalan på y-axeln (använd Matlab-kommandot `semilogy` för att plotta). Prova med några olika rums- och tidssteg och se om det har effekt på lösningen.

Tips 1: Tänk igenom vad som behöver göras innan ni börjar koda!

Tips 2: På Canvas kurssidan finns en mall för koden som ni kan utgå från och figur F.1 och F.7.

b) Välj nu källterm och begynnelsedata enligt följande

$$f(x,t) = \frac{10}{\sigma^2}\exp\left(-t - \frac{(x-\bar{x})^2}{\sigma^2}\right),$$
$$u_0(x) = \cos\left(\frac{\pi x}{2}\right).$$

Välj olika $\bar{x} \in (0,1)$, till exempel $\bar{x} = 1/4, 1/2$ och $\bar{x} = 3/4$, och ta $\sigma = 0.02$, $T = 2$ och lämpligt val av tidssteg. Visualisera resultatet (till exempel en `plot` för varje tidssteg, eller med hjälp av `surf`) och bedöm hur rimligt det verkar, i termer av värmeledning, i förhållande till begynnelsedata, källterm och randvillkor.

*Tips: Man kan använda inbyggd kvadraturregel i Matlab (**`quad`** eller **`trapz`**) för beräkning av lastvektorn, eller skapa en egen med de metoder vi lärt oss i kursen.*

1. Description of the problem

The aim of this task is to study the FE approximation of the *heat equation*

$$\begin{cases} u_t(x,t) - \alpha u_{xx}(x,t) = f(x,t), & 0 < x < 1, \quad 0 < t < T, \\ u_x(0,t) = u(1,t) = 0, & 0 < t < T, \\ u(x,0) = u_0(x), & 0 < x < 1. \end{cases} \tag{12}$$

Here, the unknown $u(x,t)$ is the temperature distribution, $f(x,t)$ and $u_0(x)$ are given source term, resp. initial values. For ease of presentation, we set $\alpha = 1$. Compare the above PDE with equation (15-17) in Welty et al, Fundamentals of Momentum, Heat and Mass transfer (6th ed.).

Observe that one has homogeneous Neumann BC for $x = 0$, that is $u_x(0,t) = 0$, and homogeneous Dirichlet BC for $x = 1$, that is $u(1,t) = 0$.

2. Variational formulation

In order to get a numerical approximation of solutions to the PDE (12) (with $\alpha = 1$), we start with deriving a variational formulation. We choose the space of test functions $V = \{w\colon [0,1] \to \mathbb{R} : \|w\|^2_{L^2(0,1)} + \|w'\|^2_{L^2(0,1)} < \infty \text{ and } w(1) = 0\}$. This choice is taken in order to match the BC of the PDE (no condition on the test function at $x = 0$ since one has Neumann BC). We then multiply the PDE (12) with a test function $v \in V$ and integrage (with respect to x) over the domain $(0,1)$. This gives us the following

$$\int_0^1 u_t(x,t)v(x)\,\mathrm{d}x - \int_0^1 u_{xx}(x,t)v(x)\,\mathrm{d}x = \int_0^1 f(x,t)v(x)\,\mathrm{d}x. \tag{13}$$

Performing a partial integration in the second integral in (13) gives us

$$\int_0^1 u_t(x,t)v(x)\,\mathrm{d}x - [u_x(x,t)v(x)]_{x=0}^{x=1} + \int_0^1 u_x(x,t)v_x(x)\,\mathrm{d}x = \int_0^1 f(x,t)v(x)\,\mathrm{d}x. \tag{14}$$

Since $v(1) = 0$ and $u_x(0,t) = 0$, one gets

$$\int_0^1 u_t(x,t)v(x)\,\mathrm{d}x + \int_0^1 u_x(x,t)v_x(x)\,\mathrm{d}x = \int_0^1 f(x,t)v(x)\,\mathrm{d}x. \tag{15}$$

This then gives us the following variational formulation for the PDE (12): For all fixed $t \in (0,T]$, find $u(\cdot,t) \in V$ such that equation (15) is fulfilled for all $v \in V$.

Observe that the above variational formulation is only in the spatial variable x. The time variable t is considered as a fixed parameter.

3. Spatial discretisation

Like in the first computer lab, we shall use the above variational formulation to get a discretisation in space. We consider the partition $\mathcal{T}_h : jh$ for $j = 0, 1, \ldots, m+1$ of the interval $0 \leq x \leq 1$ in $m+1$ subintervals of the same length $h = \frac{1}{m+1}$. We then choose the space V_h to be a subspace of V consisting of continuous functions which are piecewise linear on the partition $\mathcal{T}_h$. The FE formulation then reads: For all fixed $t \in (0, T]$, find $u_h(\cdot, t) \in V_h$ such that

$$\int_0^1 u_{h,t}(x,t)\chi(x)\,\mathrm{d}x + \int_0^1 u_{h,x}(x,t)\chi'(x)\,\mathrm{d}x = \int_0^1 f(x,t)\chi(x)\,\mathrm{d}x, \qquad \forall \chi \in V_h. \tag{16}$$

Let $\{\varphi_j\}_{j=0}^m$ be the standard basis of V_h consisting of the usual hat functions (observe that φ_{m+1} is not included since $u(1,t) = u_h(1,t) = 0$). One can then write the numerical approximation u_h as

$$u_h(x,t) = \xi_0(t)\varphi_0(x) + \xi_2(t)\varphi_2(x) + \ldots + \xi_m(t)\varphi_m(x) = \sum_{j=0}^m \xi_j(t)\varphi_j(x). \tag{17}$$

Note that the coefficients $\xi_j(t)$ depend on the time variable but not on the spatial one.

Inserting equation (17) in equation (16) and using the test functions $\chi = \varphi_i,\ i = 0, 1, \ldots, m$, one then obtains

$$\int_0^1 \Big(\sum_{j=0}^m \dot{\xi}_j(t)\varphi_j(x)\Big)\varphi_i(x)\,\mathrm{d}x + \int_0^1 \Big(\sum_{j=0}^m \xi_j(t)\varphi_j'(x)\Big)\varphi_i'(x)\,\mathrm{d}x = \int_0^1 f(x,t)\varphi_i(x)\,\mathrm{d}x$$

for $i = 0, 1, \ldots, m$.

A rearrangement gives

$$\sum_{j=0}^m \dot{\xi}_j(t)\Big(\int_0^1 \varphi_j(x)\varphi_i(x)\,\mathrm{d}x\Big) + \sum_{j=0}^m \xi_j(t)\Big(\int_0^1 \varphi_j'(x)\varphi_i'(x)\,\mathrm{d}x\Big) = \int_0^1 f(x,t)\varphi_i(x)\,\mathrm{d}x$$

which is a system of $m+1$ ODEs (for $i = 0, \ldots, m$) with $m+1$ unknown functions $\{\xi_j(t)\}_{j=0}^m$. In matrix notation, one gets

$$M\dot{\boldsymbol{\xi}}(t) + S\boldsymbol{\xi}(t) = \mathbf{F}(t). \tag{18}$$

Here, $\boldsymbol{\xi}(t)$ is the vector containing the nodal values $\xi_j(t)$ of the spatial approximation $u_h(x,t)$.

The elements of the mass matrix M and stiffness matrix S are given by

$$m_{ij} = \int_0^1 \varphi_i(x)\varphi_j(x)\,\mathrm{d}x, \quad s_{ij} = \int_0^1 \varphi_i'(x)\varphi_j'(x)\,\mathrm{d}x, \qquad i,j = 0, 1, \ldots, m, \tag{19}$$

see for instance the lecture or Chapter 5.3 in the book). The elements of the load vector $\mathbf{F}(t)$ are given by

$$F_i(t) = \int_0^1 f(x,t)\varphi_i(x)\,\mathrm{d}x, \qquad i = 0, 1, \ldots, m. \tag{20}$$

4. Discretisation in time

In order to use the above FEM, one has to solve the *semi-discrete* problem (that is discrete in x) $M\dot{\boldsymbol{\xi}}(t) + S\boldsymbol{\xi}(t) = \mathbf{F}(t)$. To do so, we introduce a partition $0 = t_0 < t_1 < t_2 < \ldots < t_n = T$ of the time interval $[0, T]$ in n subintervals of same length $k = T/n$ (hence $t_\ell = \ell k,\ \ell = 0, \ldots, n$). We then approximate the time derivative $\dot{\boldsymbol{\xi}}(t)$ with the difference quotient

$$M\frac{\boldsymbol{\xi}^{(\ell)} - \boldsymbol{\xi}^{(\ell-1)}}{k} + S\boldsymbol{\xi}^{(\ell)} = \mathbf{F}(t_\ell), \qquad \ell = 1, \ldots, n \tag{21}$$

where $\boldsymbol{\xi}^{(\ell)} \approx \boldsymbol{\xi}(t_\ell)$ for $\ell = 0, \ldots, n$. A reorganisation of the above gives the iterative scheme

$$(M + kS)\boldsymbol{\xi}^{(\ell)} = M\boldsymbol{\xi}^{(\ell-1)} + k\mathbf{F}(t_\ell), \tag{22}$$

which is called the *backward Euler scheme*[2]. There are several possibilities to choose the initial value $\boldsymbol{\xi}^{(0)}$, but the easiest one is to consider a linear interpolation of $u_0(x)$, that is $\xi_j^{(0)} = u_0(x_j),\ j = 0, \ldots, m$. One can use various quadrature formulas for approximating the integrals in $F_i(t_\ell)$. In other words, one start with some initial values and some quadrature formula, and then compute successively the approximation $\boldsymbol{\xi}^{(1)}$, $\boldsymbol{\xi}^{(2)}$, …, $\boldsymbol{\xi}^{(n)}$ using the numerical scheme (22).

5. Tasks

a) Implement the numerical scheme (22) for an approximation of solutions to the above heat equation. In order to validate your codes, you can compare the results of your implementation with the plots in Figures F.1 and F.7 in the appendix F in Welty et al, Fundamentals of Momentum, Heat and Mass transfer (6th ed.). See also equation (17-7) on page 259 and section 17.2 in Welty et al.

   Observe that the Neumann BC in the PDE (12) corresponds to solving the problem on a half plate and mirror the solution to the other half. Dirichlet BC would correspond to $h \to \infty$ in Weltys notations. Setting $u_0(x) = 1 - x$ and $f(x,t) = 0$ in the above PDE, one then gets the problem on page 259 in Welty, where $\alpha = 1$, $x_1 = 1$, $T_\infty = 0$, $T_0 = 1$ and $h \to \infty$. For comparison with these figures, one then takes $m = 0$, $Y = U$, $X = t$ and $n = x$.

[2] other possibilities of FD approximation are presented in Chapter 6.2 in the book.

Use your code to reproduce the figures F.1 (longtime) and F.7 (short time) for $m = 0$. Observe the logarithmic scale on the y-axis (use the Matlab command `semilogy` to get such plots). Test several different discretisation parameters (h and k) and see how this affects the numerical solutions.

Tips 1: Think before you start to implement!

Tips 2: You can download a template of the code and the figure F.1 and F.7 on the Canvas homepage of the course.

b) Now choose the source term and the initial value as follows

$$f(x,t) = \frac{10}{\sigma^2}\exp\left(-t - \frac{(x-\bar{x})^2}{\sigma^2}\right),$$
$$u_0(x) = \cos\left(\frac{\pi x}{2}\right).$$

Choose different values of $\bar{x} \in (0,1)$, for instance $\bar{x} = 1/4, 1/2$ and $\bar{x} = 3/4$, and take $\sigma = 0.02$, $T = 2$ and an appropriate choice for the timestep k. Visualise your results (using a `plot` at each timestep, or using `surf`) and assess how reasonable your results are, in terms of the heat conduction, the inital values, the source term, and the BC.

*Tips: You can use a quadrature formula implemented in Matlab (**`quad`** or **`trapz`**) to compute the load vector, or even better, you can implement your own Matlab function to do this, see the lecture.*