TMA225 Differential Equations and Scientific Computing for Kb2, part A

NOTES ON TIME DEPENDENT PROBLEMS

1. The Model Problem

We consider the following time dependent model problem,

$$
\begin{aligned}
\dot{u} - (au')' &= f, \qquad x_{\min} < x < x_{\max}, \quad 0 < t < T, \\
u(x_{\min}, t) &= 0, \qquad\qquad\qquad\qquad\quad 0 < t < T, \\
u(x_{\max}, t) &= 0, \qquad\qquad\qquad\qquad\quad 0 < t < T, \\
u(x, 0) &= u_0(x), \quad x_{\min} < x < x_{\max},
\end{aligned} \tag{1}
$$

where $u = u(x,t)$ is the unknown function that we wish to compute, with time derivative, $\frac{\partial u}{\partial t}$, denoted by $\dot{u}$, and x-derivative, $\frac{\partial u}{\partial x}$, denoted by u'. The functions $a = a(x,t)$ and $f = f(x,t)$ are *data* to the problem. We also need to specify *boundary data*: in (1) we have *homogeneous Dirichlet boundary conditions* at both end-points, $x = x_{\min}$, $x_{\max}$, for all times, $0 < t < T$, and *initial data*: $u_0(x)$, which specifies the solution, for $x_{\min} < x < x_{\max}$, at time $t = 0$.

2. The Numerical Method

We construct a numerical method by *first discretizing in space* (using finite elements) to obtain a finite dimensional system of linear, ordinary differential equations. We then *discretize in time* and solve the system of ODE numerically (using the backward Euler method).

2.1. Space Discretization.

2.1.1. *Variational Formulation.* Multiply the differential equation in (1) by a *test function* $v(x) \in H_0^1([x_{\min},\, x_{\max}]) := \left\{ v(x) : \int_{x_{\min}}^{x_{\max}} v'(x)^2\, dx < \infty,\ v(x_{\min}) = v(x_{\max}) = 0 \right\}$, and integrate over $[x_{\min},\, x_{\max}]$:

$$\int_{x_{\min}}^{x_{\max}} \dot{u} v\, dx \;-\; \int_{x_{\min}}^{x_{\max}} (au')'v\, dx \;=\; \int_{x_{\min}}^{x_{\max}} fv\, dx, \quad 0 < t < T.$$

We now integrate by parts:

$$\int_{x_{\min}}^{x_{\max}} \dot{u} v\, dx \;-\; \left[(au')v\right]_{x=x_{\min}}^{x=x_{\max}} \;+\; \int_{x_{\min}}^{x_{\max}} au'v'\, dx \;=\; \int_{x_{\min}}^{x_{\max}} fv\, dx, \quad 0 < t < T.$$

Since

$$v(x_{\min}) = v(x_{\max}) = 0,$$

we obtain

$$\int_{x_{\min}}^{x_{\max}} \dot{u} v\, dx \;+\; \int_{x_{\min}}^{x_{\max}} au'v'\, dx \;=\; \int_{x_{\min}}^{x_{\max}} fv\, dx, \quad 0 < t < T.$$

We now state the following *variational formulation* of (1):

Find $u(x,t)$ such that, for every fixed t: $u(x,t) \in H_0^1([x_{\min},\, x_{\max}])$, and

Date: September 16, 2002.

$$\int_{x_{\min}}^{x_{\max}} \dot{u} v \, dx \; + \; \int_{x_{\min}}^{x_{\max}} a u' v' \, dx \; = \; \int_{x_{\min}}^{x_{\max}} f v \, dx, \quad 0 < t < T, \quad \forall v \in H_0^1([x_{\min},\, x_{\max}]). \tag{2}$$

2.1.2. *Discretization in Space.* In order to discretize (2) in space, we introduce the vector space, $\overset{\circ}{V}_h$, of *continuous, piecewise linear* functions, $v(x)$, on a partition, $x_{\min} = x_0 < x_1 < \ldots < x_N < x_{N+1} = x_{\max}$, of $[x_{\min},\, x_{\max}]$, such that $v(x_{\min}) = v(x_{\max}) = 0$, and state the following *(space) discrete* counterpart of (2):

Find $U(x,t)$ such that, for every fixed t: $U(x,t) \in \overset{\circ}{V}_h$, and

$$\int_{x_0}^{x_{N+1}} \dot{U} v \, dx \; + \; \int_{x_0}^{x_{N+1}} a U' v' \, dx \; = \; \int_{x_0}^{x_{N+1}} f v \, dx, \quad 0 < t < T, \quad \forall v \in \overset{\circ}{V}_h \; . \tag{3}$$

2.1.3. *Ansatz.* We now seek a solution, $U(x,t)$, to (3), expressed (for every fixed t) in the basis of *hat functions* $\{\varphi_i\}_{i=1}^N \subset \overset{\circ}{V}_h$. (Note that φ_0 and φ_{N+1} do *not* belong to the basis, since all functions in $\overset{\circ}{V}_h$ are zero at the end-points.) In other words, we make the *Ansatz*

$$U(x,t) = \sum_{j=1}^{N} \xi_j(t) \varphi_j(x), \tag{4}$$

and seek to determine the (time dependent) coefficient vector

$$\xi(t) = \begin{bmatrix} \xi_1(t) \\ \xi_2(t) \\ \vdots \\ \xi_N(t) \end{bmatrix} = \begin{bmatrix} U(x_1,t) \\ U(x_2,t) \\ \vdots \\ U(x_N,t) \end{bmatrix},$$

of nodal values of $U(x,t)$, in such a way that (3) is satisfied.

Consider *very carefully* the structure of $U(x,t)$ in (4): For every fixed time, t, we note that $U(x,t)$, as a function of x, is a continuous, piecewise linear function with weights given by $\xi(t)$.

2.1.4. *Construction of Space Discrete System of ODE.* We substitute (4) into (3):

$$\sum_{j=1}^{N} \dot{\xi}_j(t) \left(\int_{x_0}^{x_{N+1}} \varphi_j v \, dx \right) \; + \; \sum_{j=1}^{N} \xi_j(t) \left(\int_{x_0}^{x_{N+1}} a \, \varphi_j' v' \, dx \right) \; = \; \int_{x_0}^{x_{N+1}} f v \, dx, \tag{5}$$

$$0 < t < T, \quad \forall v \in \overset{\circ}{V}_h \; .$$

Since $\{\varphi_i\}_{i=1}^N \subset \overset{\circ}{V}_h$ is a *basis* for $\overset{\circ}{V}_h$, (5) is equivalent to

$$\sum_{j=1}^{N} \dot{\xi}_j(t) \left(\int_{x_0}^{x_{N+1}} \varphi_j \varphi_i \, dx \right) \; + \; \sum_{j=1}^{N} \xi_j(t) \left(\int_{x_0}^{x_{N+1}} a \, \varphi_j' \varphi_i' \, dx \right) \; = \; \int_{x_0}^{x_{N+1}} f \varphi_i \, dx, \tag{6}$$

$$0 < t < T, \quad i = 1, \ldots, N,$$

which is an N-dimensional system of linear, ordinary differential equations. Introducing the notation

$$m_{ij} = \int_{x_0}^{x_{N+1}} \varphi_j(x)\varphi_i(x)\,dx,$$

$$a_{ij}(t) = \int_{x_0}^{x_{N+1}} a(x,t)\,\varphi_j'(x)\varphi_i'(x)\,dx,$$

$$b_i(t) = \int_{x_0}^{x_{N+1}} f(x,t)\varphi_i(x)\,dx,$$

we can write the system of linear, ordinary differential equations (6), as:

$$\left\{\begin{array}{ccccccccccccc}
m_{11}\,\dot{\xi}_1(t) & + & \dots & + & m_{1\mathrm{N}}\,\dot{\xi}_{\mathrm{N}}(t) & + & a_{11}(t)\,\xi_1(t) & + & \dots & + & a_{1\mathrm{N}}(t)\,\xi_{\mathrm{N}}(t) & = & b_1(t), \\
m_{21}\,\dot{\xi}_1(t) & + & \dots & + & m_{2\mathrm{N}}\,\dot{\xi}_{\mathrm{N}}(t) & + & a_{21}(t)\,\xi_1(t) & + & \dots & + & a_{2\mathrm{N}}(t)\,\xi_{\mathrm{N}}(t) & = & b_2(t), \\
\vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\
m_{\mathrm{N}1}\,\dot{\xi}_1(t) & + & \dots & + & m_{\mathrm{NN}}\,\dot{\xi}_{\mathrm{N}}(t) & + & a_{\mathrm{N}1}(t)\,\xi_1(t) & + & \dots & + & a_{\mathrm{NN}}(t)\,\xi_{\mathrm{N}}(t) & = & b_{\mathrm{N}}(t),
\end{array}\right.$$

$$0 < t < T.$$

In *matrix form*, this reads

$$M\,\dot{\xi}(t) \,+\, A(t)\,\xi(t) \,=\, b(t), \quad 0 < t < T, \tag{7}$$

where $M = \begin{bmatrix} m_{11} & \dots & m_{1\mathrm{N}} \\ \vdots & \ddots & \vdots \\ m_{\mathrm{N}1} & \dots & m_{\mathrm{NN}} \end{bmatrix}$ is the *mass matrix*,

$A(t) = \begin{bmatrix} a_{11}(t) & \dots & a_{1\mathrm{N}}(t) \\ \vdots & \ddots & \vdots \\ a_{\mathrm{N}1}(t) & \dots & a_{\mathrm{NN}}(t) \end{bmatrix}$ is the (possibly time dependent) *stiffness matrix*, and

$$b(t) = \begin{bmatrix} b_1(t) \\ \\ \vdots \\ \\ b_N(t) \end{bmatrix}$$ is the (possibly time dependent) *load vector*.

2.2. **Time Discretization.** To discretize (7) in time, let $0 = t_0 < t_1 < t_2 < \cdots < t_L = T$ be discrete time levels with corresponding time steps $k_n = t_n - t_{n-1}$, $n = 1, \ldots, L$. Further, let ξ^n denote the *approximation* of $\xi(t_n)$, $n = 1, \ldots, L$.

There are different possible choices of *initial data*, $\xi^0 = \xi(0)$, to (7): the simplest is

$$\xi^0 = \begin{bmatrix} \xi_1(0) \\ \xi_2(0) \\ \vdots \\ \xi_N(0) \end{bmatrix} = \begin{bmatrix} u_0(x_1) \\ u_0(x_2) \\ \vdots \\ u_0(x_N) \end{bmatrix},$$

which corresponds to letting $U(x,0) = \sum_{j=1}^N \xi_j(0)\varphi_j(x)$ be the *nodal interpolant* of $u_0(x) = u(x,0)$. (An alternative would be to choose $U(x,0)$ as the $L_2([x_{\min},\ x_{\max}])$-projection of u_0, but then we would need to *compute* ξ^0.)

We now *integrate* (7) (element-wise) over one time interval $[t_{n-1},\ t_n]$:

$$\int_{t_{n-1}}^{t_n} M\,\dot{\xi}(t)\,dt + \int_{t_{n-1}}^{t_n} A(t)\,\xi(t)\,dt = \int_{t_{n-1}}^{t_n} b(t)\,dt.$$

Since M is a constant matrix, we get:

$$M(\xi(t_n) - \xi(t_{n-1})) + \int_{t_{n-1}}^{t_n} A(t)\,\xi(t)\,dt = \int_{t_{n-1}}^{t_n} b(t)\,dt. \tag{8}$$

Given an approximation, ξ^{n-1}, of $\xi(t_{n-1})$, approximating the integrals in (8) using *right end-point quadrature* gives the *backward Euler method* defining ξ^n by

$$M(\xi^n - \xi^{n-1}) + A(t_n)\xi^n k_n = b(t_n)k_n,$$

i.e.,

$$M\,\frac{\xi^n - \xi^{n-1}}{k_n} + A(t_n)\xi^n = b(t_n). \tag{9}$$

For solving (7) using the *backward Euler method* we can now state the following *algorithm*:

Given $\xi^0 = \xi(0)$. For $n = 1, \ldots, L$: Solve the linear system of equations

$$(M + k_n A_n)\xi^n = M\xi^{n-1} + k_n b_n. \tag{10}$$

In (10) we have introduced the notation

$$A_n = A(t_n), \quad b_n = b(t_n).$$

Remark. Observe the similarity between (7) and (9): We may alternatively view the backward Euler method as approximating the derivative by a difference quotient, and evaluating the other terms at the right end-point of the time interval.