ÜBUNGEN ZU

NUMERIK PARTIELLER DIFFERENTIALGLEICHUNGEN

für den 16. 11. 2005

Send your programs to `zulehner@numa.uni-linz.ac.at` **by 9 a.m.**

---

19. Write a function `ImplementNeumannBC(`$\downarrow$`i,`$\downarrow$`g,`$\updownarrow$`vector)` to implement a Neumann boundary condition
$$\frac{\partial u}{\partial n}(x_i) = g_N(x_i),$$
for a given value $\mathtt{g} = g_N(x_i)$ at the boundary node x_i identified by its node number $= i$ by updating the input `vector` $= \underline{f}_h$, previously computed by using `LoadVector` for the case of homogenous Neumann conditions only.

20. Write a function `ImplementDirichletBC(`$\downarrow$`i,`$\downarrow$`g,`$\updownarrow$`matrix,`$\updownarrow$`vector)` to implement a Dirichlet boundary condition
$$u(x_i) = g_D(x_i),$$
for a given value $\mathtt{g} = g_D(x_i)$ at the boundary node x_i identified by its index $\mathtt{i} = i$ by updating the input `matrix` $= K_h$ and `vector` $= \underline{f}_h$, previously computed by using `StiffnessMatrix`, `LoadVector` and `ImplementNeumannBC` for the case of Neumann boundary conditions only.

    Hint: Assume that the following equation is obtained after applying `StiffnessMatrix`, `LoadVector` and `ImplementNeumannBC`:
$$\begin{pmatrix} K_{00} & K_{01} & K_{02} \\ K_{10} & K_{11} & K_{12} \\ K_{20} & K_{21} & K_{22} \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \end{pmatrix},$$
and we want to fix the value at x_0:
$$u_0 = u(x_0) = g_G(x_0) = g_0.$$
This can be done by replacing the first equation by the simple equation $K_{00}u_0 = K_{00}g_0$ and by substituting u_0 by g_0 in the other equations. This results in the new updated system
$$\begin{pmatrix} K_{00} & 0 & 0 \\ 0 & K_{11} & K_{12} \\ 0 & K_{21} & K_{22} \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} K_{00}\, g_0 \\ f_1 - K_{10}\, g_0 \\ f_2 - K_{20}\, g_0 \end{pmatrix}.$$

21. Write a function `Mult(`$\downarrow$`matrix,`$\downarrow$`vector,`$\uparrow$`product)` which returns the matrix-vector product $Ku(=$ `product`$)$ of the matrix $K(=$ `matrix`$)$ and the vector $u(=$ `vector`$)$, where K is a given tridiagonal matrix, implemented by the data type `Matrix`, and u is a given vector. In C++ this could also be done either by writing an appropriate member function for the class `Matrix` or by overloading the operator $*$.

22. Define a data type `Preconditioner` by using `struct` in C or `class` in C++, which contains all information needed for the Jacobi preconditioner $C_h = D_h = \text{diag}(K_h)$. Write a function (or a member function of the class `Preconditioner` in C++), which solves the equation

$$C_h \underline{w}_h = \underline{r}_h$$

for $C_h = D_h$ and a given vector $\underline{r}_h$.

Test your functions for a simple example.

23. Let

$$\hat{K} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}, \quad \hat{M} = \begin{pmatrix} 1/3 & 1/6 \\ 1/6 & 1/3 \end{pmatrix}, \quad \hat{D} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Show that

$$\frac{1}{6}\,\hat{D} \le \hat{M} \quad \text{and} \quad \hat{K} \le 2\,\hat{D}.$$

24. Consider the one-dimensional boundary value problem

$$\begin{aligned} -u''(x) &= f(x) \quad x \in (0,1), \\ u(0) &= g_0, \\ u'(1) &= g_1. \end{aligned}$$

Let K_h be the stiffness matrix obtained by the finite element with the Courant element on a subdivision given by $0 = x_0 < x_1 < \cdots < x_{N_h} = 1$.

Show that

$$\frac{\min_k h_k^2}{6c_F^2}\, D_h \le K_h \le 2\, D_h,$$

with $D_h = \text{diag}(K_h)$, c_F the constant from Friedrichs' inequality, and $h_k = x_k - x_{k-1}$.

Hint: Use

$$(D_h \underline{v}_h, \underline{v}_h)_{\ell_2} = D_h^{(1)} v_1^2 + \sum_{k=2}^{N_h} \left(D_h^{(k)} \begin{pmatrix} v_{k-1} \\ v_k \end{pmatrix}, \begin{pmatrix} v_{k-1} \\ v_k \end{pmatrix} \right)_{\ell_2}$$

with

$$D_h^{(1)} = K_h^{(1)} = \frac{1}{h_1} \quad \text{and} \quad D_h^{(k)} = \text{diag}(K_h^{(k)}) = \frac{1}{h_k}\,\text{diag}(\hat{K}) = \frac{1}{h_k}\,\hat{D}.$$