Project 1 - FYS3150

René Ask and Benedicte Nyheim
(Dated: August 25, 2019)

I. METHOD

We're going to solve the differential equation

$$-u''(x) = f(x), \quad x \in (0,1), \quad u(0) = u(1) = 0. \tag{1}$$

We'll approximate this differential equation by a function $v(x) \approx u(x)$ by the approximation scheme

$$-\frac{-v_{i+1} + v_{i-1} - 2v_i}{h^2} = f_i, \quad i = 1, 2, ..., n, \tag{2}$$

which may be rearranged into

$$2v_i - v_{i+1} - v_{i-1} = f_i h^2 \equiv \tilde{b_i}. \tag{3}$$

From (3) we can write

$$\begin{pmatrix} 2v_1 - v_2 \\ -v_1 + 2v_2 - v_3 \\ \vdots \\ 2v_n - v_{n-1} \end{pmatrix} = \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \cdots & 0 \\ \vdots & & & & \\ 0 & \cdots & 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} = \begin{pmatrix} \tilde{b_1} \\ \tilde{b_2} \\ \vdots \\ \tilde{b_n} \end{pmatrix} \tag{4}$$

To this end we will develop an algorithm based on the LU-decomposition $A = LU$:

$$A = \begin{pmatrix} b_1 & c_1 & 0 & \cdots & \cdots & \cdots \\ a_1 & b_2 & c_2 & 0 & \cdots & \cdots \\ 0 & a_2 & b_3 & 0 & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{n-2} & b_{n-1} & c_{n-1} \\ 0 & 0 & \cdots & 0 & a_{n-1} & b_n \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & \cdots & \cdots & 0 \\ \ell_2 & 1 & \cdots & \cdots & \cdots & 0 \\ 0 & \ell_3 & 1 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & \cdots & \ell_{n-1} & 1 & 0 \\ 0 & 0 & \cdots & \cdots & \ell_n & 1 \end{pmatrix} \begin{pmatrix} d_1 & u_1 & \cdots & \cdots & \cdots & 0 \\ 0 & d_2 & u_2 & \cdots & \cdots & 0 \\ 0 & 0 & d_3 & u_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots & d_{n-1} & u_{n-1} \\ 0 & 0 & \cdots & \cdots & 0 & d_n \end{pmatrix} = LU, \tag{5}$$

and performing matrix multiplication we get

$$LU = \begin{pmatrix} d_1 & u_1 & \cdots & \cdots & \cdots & 0 \\ \ell_2 d_1 & \ell_2 u_1 + d_2 & u_2 & \cdots & \cdots & 0 \\ 0 & \ell_3 d_2 & \ell_3 u_2 + d_3 & u_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \ell_{n-1} d_{n-2} & \ell_{n-1} u_{n-2} + d_{n-1} & u_{n-1} \\ 0 & 0 & \cdots & \cdots & \ell_n d_{n-1} & \ell_n u_{n-1} + d_n \end{pmatrix}, \tag{6}$$

which yields the following general relations:

$$b_i = d_i, \qquad c_i = u_i, \qquad \text{for} \qquad i = 1, \tag{7}$$

$$\ell_i = \frac{a_{i-1}}{d_{i-1}}, \qquad \text{for} \qquad 1 < i < n, \tag{8}$$

$$d_i = b_i - \ell_i u_{i-1}, \qquad \text{for} \qquad 1 < i < n, \tag{9}$$

$$\ell_n = \frac{a_{n-1}}{d_{n-1}}, \qquad d_n = b_n - \ell_n u_{n-1}, \qquad \text{for} \qquad i = n. \tag{10}$$

requiring $2n$ multiplications and n additions. In other words, the total floating point operations involved in finding LU is $3n$.

We can then write $A\boldsymbol{v} = LU\boldsymbol{v} = L\boldsymbol{y} = \tilde{\boldsymbol{b}}$ where $\boldsymbol{y} \equiv U\boldsymbol{v}$. Explicitly, we can write this as

$$L\boldsymbol{y} = \begin{pmatrix} 1 & 0 & \cdots & \cdots & \cdots & 0 \\ \ell_2 & 1 & \cdots & \cdots & \cdots & 0 \\ 0 & \ell_3 & 1 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & \cdots & \ell_{n-1} & 1 & 0 \\ 0 & 0 & \cdots & \cdots & \ell_n & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} y_1 \\ \ell_2 y_1 + y_2 \\ \ell_3 y_2 + y_3 \\ \vdots \\ \ell_{n-1} y_{n-2} + y_{n-1} \\ \ell_n y_{n-1} + y_n \end{pmatrix} = \begin{pmatrix} \tilde{b_1} \\ \tilde{b_2} \\ \vdots \\ \vdots \\ \tilde{b_n} \end{pmatrix} \tag{11}$$

which yields the following procedure:

$$y_1 = \tilde{b_1}, \tag{12}$$

$$y_i = \tilde{b_i} - \ell_i y_{i-1}, \qquad \text{for} \qquad i = 2, 3, ..., n. \tag{13}$$

giving $n-1$ multiplications and $n-1$ additions. Thus the added computational cost is $2(n-1)$.

Finally, to determine $\boldsymbol{v}$, we perform back-substitution by solving $U\boldsymbol{v} = \boldsymbol{y}$. Writing it out explicitly yields

$$\begin{pmatrix} d_1 & u_1 & \cdots & \cdots & \cdots & 0 \\ 0 & d_2 & u_2 & \cdots & \cdots & 0 \\ 0 & 0 & d_3 & u_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots & d_{n-1} & u_{n-1} \\ 0 & 0 & \cdots & \cdots & 0 & d_n \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ \vdots \\ v_n \end{pmatrix} = \begin{pmatrix} d_1 v_1 + u_1 v_2 \\ d_2 v_2 + u_2 v_3 \\ \vdots \\ \vdots \\ d_{n-1} v_{n-1} + u_{n-1} v_n \\ d_n v_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ \vdots \\ y_{n-1} \\ y_n \end{pmatrix}, \tag{14}$$

which yields the following procedure:

$$v_n = \frac{y_n}{d_n}, \tag{15}$$

$$v_i = \frac{y_i - u_i v_{i+1}}{d_i}, \qquad \text{for} \qquad i = n-1, n-2, ..., 1. \tag{16}$$

Here we got $2n-1$ multiplications and $n-1$ additions, so the number of floating point operations are $3n-2$. Putting all of these together we get $4(2n-1) \approx 8n$ floating point operations.

Now, assuming that $b_1 = b_2 = \cdots = b_n \equiv b$ and $a_1 = c_1$, $a_2 = c_2, \cdots, a_{n-1} = c_{n-1}$. Furthermore, assume $a_1 = a_2 = \cdots = a_n \equiv a$ and $c_1 = c_2 = \cdots = c_n \equiv c$. But since $a = c$, we can ignore the last assumption. These assumptions implies certain simplifications of the algorithm for LU-decomposition:

$$d_1 = b, \qquad u_1 = c = a, \tag{17}$$

$$u_i = c_i = a, \qquad \text{for} \qquad 1 < i \leq n, \tag{18}$$

$$\ell_i = \frac{a_{i-1}}{d_{i-1}} = \frac{a}{d_{i-1}}, \qquad \text{for} \qquad 1 < i \leq n, \tag{19}$$

$$d_i = b_i - \ell_i u_i = b - \ell_i a = b - \frac{a^2}{d_{i-1}} \qquad \text{for} \qquad 1 < i \leq n. \tag{20}$$