4. Numerical Methods

4.1. Band Matrix Methods for differential equations

Assume we want to solve the differential equation

$$y''(x) + ay'(x) + by(x) = g(x) \tag{4.1}$$

over some interval with initial conditions given. If we want to solve this numerically we first have to get rid of the abstract infinitely small differences called differentials. We approximate these with finite size differences so that

$$\begin{aligned}
&y'(x) \approx \frac{y(x+\Delta x) - y(x)}{\Delta x} \\
&y''(x) \approx \left(\frac{y(x+\Delta x) - y(x)}{\Delta x} - \frac{y(x) - y(x-\Delta x)}{\Delta x} \right) \Big/ \Delta x \\
&= \frac{y(x+\Delta x) - 2y(x) + y(x-\Delta x)}{\Delta x^2}
\end{aligned} \tag{4.2}$$

Using this we can solve the equation in the following way. Say we want to solve the equation in the interval [*p*,*q*] for x and we know that the first and second derivatives are zero at *p*. We divide the interval for x into *n* equal parts and use the following notation

$$\begin{aligned}
&p + \frac{k(q-p)}{n} \equiv x_k \\
&x_k - x_{k-1} = \frac{q-p}{n} \equiv \Delta
\end{aligned} \tag{4.3}$$

This gives us the following equations.

$$
\begin{aligned}
(y_{-1}-2y_0+y_1)\Delta^{-2} &= 0 \\
(-y_0+y_1)\Delta^{-1} &= 0 \\
(y_{-1}-2y_0+y_1)\Delta^{-2}+a(-y_0+y_1)\Delta^{-1}+by_0 &= g(x_0) \\
(y_0-2y_1+y_2)\Delta^{-2}+a(-y_1+y_2)\Delta^{-1}+by_1 &= g(x_1) \\
&\cdots \\
(y_{n-1}-2y_n+y_{n+1})\Delta^{-2}+a(-y_n+y_{n+1})\Delta^{-1}+by_n &= g(x_n)
\end{aligned}
\tag{4.4}
$$

This is n+3 linear equations for the n+3 unknown y. Writing this as a system we have

$$
\mathbf{A}\begin{bmatrix} y_{-1} \\ y_0 \\ \vdots \\ y_n \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ g(x_0) \\ \vdots \\ g(x_n) \end{bmatrix} \Rightarrow \begin{bmatrix} y_{-1} \\ y_0 \\ \vdots \\ y_n \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} y(x_{-1}) \\ y(x_0) \\ \vdots \\ y(x_n) \\ y(x_{n+1}) \end{bmatrix} = \mathbf{A}^{-1}\begin{bmatrix} 0 \\ 0 \\ g(x_0) \\ \vdots \\ g(x_n) \end{bmatrix}
\tag{4.5}
$$

Example; Let us solve the following equation numerically

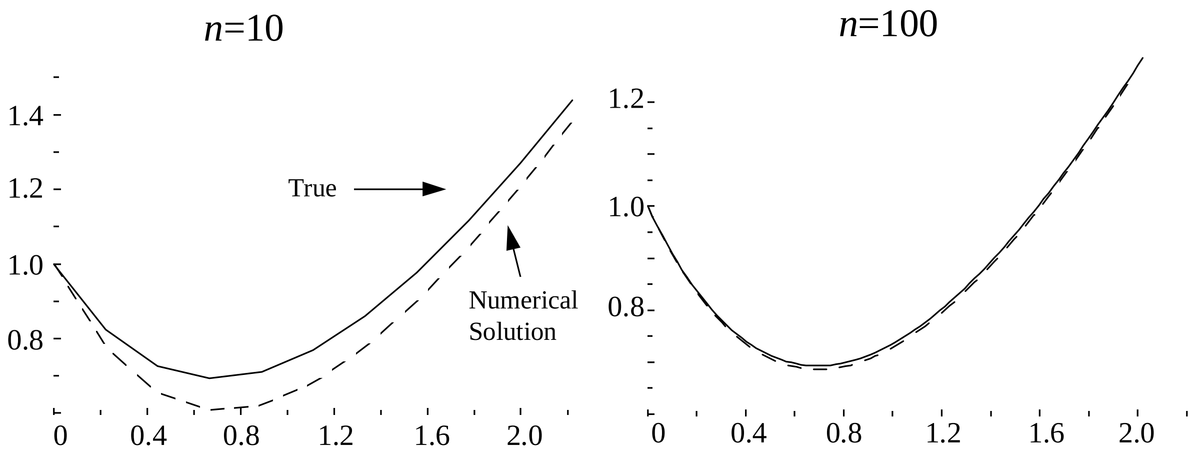
$$
\begin{aligned}
&y'(x)+y(x)=x \\
&x_0=0, x_n=2, y(x_0)=1.
\end{aligned}
\tag{4.6}
$$

The system resulting from the approximation can be written as

$$
\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 1-\Delta^{-1} & \Delta^{-1} & 0 & \cdots & 0 & 0 \\ 0 & 1-\Delta^{-1} & \Delta^{-1} & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 1-\Delta^{-1} & \Delta^{-1} \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}
\tag{4.7}
$$

where the first equation is the initial condition.

The solutions tend to be closer to the algebraic solution the larger is n, but is fairly close here also for small values of n. This method is very well suited for computers.



4.2. Non-Linear Equations

Suppose we are faced with the problem of solving

$$0 = \begin{bmatrix} g_1(x_1^*, \ldots x_n^*) \\ \vdots \\ g_n(x_1^*, \ldots x_n^*) \end{bmatrix} \equiv \mathbf{g}\left(\mathbf{x}^*\right) \tag{4.8}$$

for $\mathbf{x}^*$. This could for example be the equation implicitly defining a steady state of difference or differential equation system. Starting from some initial value $\mathbf{x}_n$ we can then make a first order Taylor approximation. Define

$$\mathbf{Dg}(\mathbf{x}) \equiv \begin{bmatrix} \partial g_1(x_1, \ldots x_n)/\partial x_1 & \ldots & \partial g_1(x_1, \ldots x_n)/\partial x_n \\ \vdots & \ddots & \vdots \\ \partial g_n(x_1, \ldots x_n)/\partial x_1 & \ldots & \partial g_n(x_1, \ldots x_n)/\partial x_1 \end{bmatrix} \tag{4.9}$$

Then we have

$$
\begin{aligned}
&0=\mathbf{g}\left(\mathbf{x}^{*}\right)\approx\mathbf{g}\left(\mathbf{x}_{\mathbf{0}}\right)+\mathbf{Dg}\left(\mathbf{x}_{\mathbf{0}}\right)\left(\mathbf{x}^{*}-\mathbf{x}_{\mathbf{0}}\right)\\
&\Rightarrow\left(\mathbf{x}^{*}-\mathbf{x}_{\mathbf{0}}\right)\approx-\mathbf{Dg}\left(\mathbf{x}_{\mathbf{0}}\right)^{-1}\mathbf{g}\left(\mathbf{x}_{\mathbf{0}}\right)
\end{aligned}
\tag{4.10}
$$

This gives us a better approximation to $\mathbf{x}^*$ than our first $\mathbf{x}_n$ guess. We can then repeat this until we are sufficiently close to **g(x)**=0. Each repetition we thus update our value for **x** according to

$$
\begin{aligned}
&0=\mathbf{g}\left(\mathbf{x}^{*}\right)\approx\mathbf{g}\left(\mathbf{x}_{0}\right)+\mathbf{Dg}\left(\mathbf{x}_{0}\right)\left(\mathbf{x}^{*}-\mathbf{x}_{0}\right)\\
&\Rightarrow\left(\mathbf{x}^{*}-\mathbf{x}_{0}\right)\approx-\mathbf{Dg}\left(\mathbf{x}_{0}\right)^{-1}\mathbf{g}\left(\mathbf{x}_{0}\right)\\
&\mathbf{x}_{k+1}=\mathbf{x}_{k}-\mathbf{Dg}\left(\mathbf{x}_{k}\right)^{-1}\mathbf{g}\left(\mathbf{x}_{k}\right)
\end{aligned}
\tag{4.11}
$$

Example;

$$
\begin{aligned}
&\mathbf{g}(\mathbf{x})\equiv\begin{bmatrix} x_1^3 e^{x_2}-2\\ x_1^3-3x_2^2+1\end{bmatrix}\\
&\Rightarrow\mathbf{Dg}(\mathbf{x})=\begin{bmatrix} 3x_1^2 e^{x_2} & x_1^3 e^{x_2}\\ 3x_1^2 & -6x_2\end{bmatrix}
\end{aligned}
\tag{4.12}
$$

Starting with **x=**{2,2} I got the following results

k	$\{x_{1,k},x_{2,k}\}$	$\mathbf{g}(\mathbf{x}_k)$
0	2,2	{57.112, -3}
1	{1.714, 1.464}	{19.741, -0.395}
2	{1.400, 1.104}	{6.281, 0.086}
3	{1.146, 0.891}	{1.665, 0.120}
4	{1.003, 0.809}	{0.269, 0.046}
5	{0.968, 0.797}	{0.011, 0.003}
6	{0.966, 0.796}	{0.000, 0.000}

In computer applications we may sometimes also want to approximate the matrix of derivatives with a discrete approximation as in (4.2).

4.3. Newton-Raphson

Suppose we are looking for an optimum of the function $f(\mathbf{x}), \mathbf{x} \in R^n$. A standard way to do this numerically is to apply the *Newton-Raphson* algoritm. If the optimum is interior and *f* is differentiable, it satifies the necessary first order conditions that the gradient is zero

$$Df(\mathbf{x}^*) = \mathbf{0}. \tag{4.13}$$

Now apply a first order linear approximation to the gradient from some initial point **x**

$$\mathbf{0} = Df(\mathbf{x}^*) \approx Df(\mathbf{x}) + D^2 f(\mathbf{x})\left(\mathbf{x} - \mathbf{x}^*\right). \tag{4.14}$$

By rearranging terms we get an approximation to **x***

$$\mathbf{x}^* \approx \mathbf{x} - D^2 f(\mathbf{x})^{-1} Df(\mathbf{x}). \qquad (4.15).$$

From this we can construct a search algorithm that hopefully makes better and better approximations

$$\mathbf{x}_{s+1} = \mathbf{x}_s - D^2 f(\mathbf{x}_s)^{-1} Df(\mathbf{x}_s). \tag{4.16}$$

If we don’t have analytic solutions to the gradient and Hessian we can use numerical approximations, for example

$$
\begin{aligned}
\frac{\partial f(\mathbf{x})}{\partial x_1} &\approx \frac{f\left(\mathbf{x}+\begin{bmatrix}\varepsilon\\ \mathbf{0}\end{bmatrix}\right)-f(\mathbf{x})}{\varepsilon}\\
\frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_1} &\approx \frac{\dfrac{\partial f\left(\mathbf{x}+\begin{bmatrix}0\\ \varepsilon\\ \mathbf{0}\end{bmatrix}\right)}{\partial x_1}-\dfrac{\partial f(\mathbf{x})}{\partial x_1}}{\varepsilon}
\end{aligned}
\tag{4.17}
$$

which is called the forward difference method.

One should be very careful with these methods. The first problem is only local optima will be found with this method. Secondly, the method will surely converge only in a neighbourhood of **x*.** So often it is crucial for convergence which starting point is chosen.