COURSE 13

5.3. Numerical methods for solving nonlinear systems

Let $D \subseteq \mathbb{R}^n$, $f_i : D \to \mathbb{R}$, $i = 1, ..., n$ and the system

$$f_i(x_1, ..., x_n) = 0, \ i = 1, ..., n; \ (x_1, ..., x_n) \in D. \tag{1}$$

The system (1) can be written as

$$f(x) = 0, \ x \in D, \text{ with } f = (f_1, ..., f_n).$$

5.3.1. Successive approximation method

We rewrite the system (1) as

$$x_i = \varphi_i(x_1, ..., x_n), \quad i = 1, ..., n; \ (x_1, ..., x_n) \in D$$

or

$$x = \varphi(x), \quad \text{with } x = (x_1, ..., x_n) \in D \text{ and } \varphi = (\varphi_1, ..., \varphi_n), \tag{2}$$

where $\varphi_i : D \to \mathbb{R}$ are continuous functions on D such that for any point $(x_1, ..., x_n) \in D$ to have $(\varphi_1(x_1, ..., x_n), ..., \varphi_n(x_1, ..., x_n)) \in D$.

Considering the starting point $x^{(0)}$ we generate the sequence

$$x^{(0)}, x^{(1)}, ..., x^{(n)}, ... \tag{3}$$

with

$$x^{(m+1)} = \varphi(x^{(m)}),\ m = 0, 1, ...$$

If the sequence (3) is convergent and α is its limit, then α is the solution of system (1). We have

$$\lim_{m\to\infty} x^{(m+1)} = \varphi(\lim_{m\to\infty} x^{(m)}),$$

namely,

$$\alpha = \varphi(\alpha).$$

The convergence of the method, using Picard-Banach theorem: if $\varphi : \mathbb{R}^n \to \mathbb{R}^n$ verifies the contraction condition

$$\|\varphi(x) - \varphi(y)\| \leq l \|x - y\|,\ x, y \in \mathbb{R}^n; 0 < l < 1,$$

then there exists an unique element $\alpha \in \mathbb{R}^n$, which is solution of equation (2) and limit of the sequence (3). The approximation error is:

$$\left\|\alpha - x^{(n)}\right\| \leq \frac{l^n}{1-l}\left\|x^{(1)} - x^{(0)}\right\|.$$

Example 1 *Choosing* $x^{(0)} = (0.1, 0.1, -0.1)$, *solve the system*

$$\begin{cases} 3x_1 - \cos(x_2 x_3) - \frac{1}{2} &= 0 \\ x_1^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1.06 &= 0 \\ e^{-x_1 x_2} + 20x_3 + \frac{10\pi - 3}{3} &= 0 \end{cases}.$$

Sol. We have

$$\begin{cases} x_1^{(1)} = \frac{1}{3}\cos(x_2^{(0)} x_3^{(0)}) + \frac{1}{6} \\ x_2^{(1)} = \frac{1}{9}\sqrt{(x_1^{(0)})^2 + \sin x_3^{(0)} + 1.06} - 0.1 \\ x_3^{(1)} = -\frac{1}{20}e^{-x_1^{(0)} x_2^{(0)}} - \frac{10\pi - 3}{60} \end{cases},$$

and

$$\begin{cases} x_1^{(m)} = \frac{1}{3}\cos(x_2^{(m-1)}x_3^{(m-1)}) + \frac{1}{6} \\ x_2^{(m)} = \frac{1}{9}\sqrt{(x_1^{(m-1)})^2 + \sin x_3^{(m-1)} + 1.06} - 0.1 \\ x_3^{(m)} = -\frac{1}{20}e^{-x_1^{(m-1)}x_2^{(m-1)}} - \frac{10\pi-3}{60} \end{cases}.$$

The sequence converges to (0.5,0,-0.5236).

Accelerating Convergence

One way to accelerate convergence of the fixed-point iteration is to use the latest estimates $x_1^{(k)}, x_2^{(k)}, ..., x_{i-1}^{(k)}$ instead of $x_1^{(k-1)}, x_2^{(k-1)}, ..., x_{i-1}^{(k-1)}$ to compute $x_i^{(k)}$, as in the Gauss-Seidel method for linear systems.

Example 2 *Choosing* $x^{(0)} = (0.1, 0.1, -0.1)$, $\varepsilon = 10^{-5}$ *solve the system*

$$\begin{cases} 3x_1 - \cos(x_2x_3) - \frac{1}{2} & = 0 \\ x_1^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1.06 & = 0 \\ e^{-x_1x_2} + 20x_3 + \frac{10\pi-3}{3} & = 0 \end{cases}.$$

Sol. We have

$$\begin{cases} x_1^{(1)} = \frac{1}{3}\cos(x_2^{(0)}x_3^{(0)}) + \frac{1}{6} \\ x_2^{(1)} = \frac{1}{9}\sqrt{(x_1^{(0)})^2 + \sin x_3^{(0)} + 1.06} - 0.1 \\ x_3^{(1)} = -\frac{1}{20}e^{-x_1^{(0)}x_2^{(0)}} - \frac{10\pi-3}{60} \end{cases},$$

and

$$\begin{cases} x_1^{(k)} = \frac{1}{3}\cos(x_2^{(k-1)}x_3^{(k-1)}) + \frac{1}{6} \\ x_2^{(k)} = \frac{1}{9}\sqrt{(x_1^{(k-1)})^2 + \sin x_3^{(k-1)} + 1.06} - 0.1 \\ x_3^{(k)} = -\frac{1}{20}e^{-x_1^{(k-1)}x_2^{(k-1)}} - \frac{10\pi-3}{60} \end{cases}.$$

The sequence converges to $(0.5, 0, -0.52359877)$ (5 iterations).

Using the method for accelerating the convergence the component equations for this problem become

$$\begin{cases} x_1^{(k)} = \frac{1}{3}\cos(x_2^{(k-1)}x_3^{(k-1)}) + \frac{1}{6} \\ x_2^{(k)} = \frac{1}{9}\sqrt{(x_1^{(k)})^2 + \sin x_3^{(k-1)} + 1.06} - 0.1 \\ x_3^{(k)} = -\frac{1}{20}e^{-x_1^{(k)}x_2^{(k)}} - \frac{10\pi-3}{60} \end{cases}.$$

In this case the convergence is indeed accelerated (4 iterations) by using the Gauss-Seidel method. However, this method does not *always* accelerate the convergence.

Example 3 *Use succesive approximation's method with* $x^{(0)} = \begin{pmatrix} 1 \\ 3 \end{pmatrix}$ *to compute* $x^{(2)}$ *for the following nonlinear systems:*

$$\begin{cases} x_1 - x_2^2 + 2x_2 = 0 \\ 2x_1 + x_2 - 6 = 0 \end{cases}.$$

Use also the method for accelerating the convergence.

5.3.2. Newton's method for solving nonlinear systems

Consider the system (1) written as

$$f(x) = 0,\ x \in D, D \subseteq \mathbb{R}^n.$$

Let $\alpha \in D$ be a solution of this system and $x^{(p)}$ an approximation of it.

The Newton's method for nonlinear systems:

$$x^{(p+1)} = x^{(p)} - J^{-1}(x^{(p)}) f(x^{(p)}),\ p = 0, 1, ... \tag{4}$$

where

$$J(x^{(p)}) = f'(x^{(p)}) = \left(\frac{\partial f_i}{\partial x_i}(x^{(p)}) \right)_{i,\ j=1,...,n}$$

is the Jacobian matrix.

If the sequence $(x^{(p)})_{p \in \mathbb{N}}$ is convergent and α is its limit then by (4) it follows that α is solution of the system. Regarding the convergence of the sequence we have:

Theorem 4 *Let $f : D \subseteq R^n \to R^n$ and consider a given norm $\|\cdot\|$ on R^n. If*

- there exists a solution $\alpha \in D$, such that $f(\alpha) = 0$;

- f is differentiable on D, with f' Lipschitz continuous, i.e., $\exists L > 0$ s.t.

$$\left\| f'(x) - f'(y) \right\| \leq L \|x - y\|, \forall x, y \in D;$$

- the Jacobian of f is nonsingular at α: $\exists f'(\alpha)^{-1} : R^n \to R^n$,

then there exists an open neighborhood $D_0 \subseteq D$ of α such that for any initial approximation $x_0 \in D_0$ the sequence generated by the Newton's method remains in D_0, converges to the solution α and there exists a constant $K > 0$ such that

$$\left\| x_{k+1} - \alpha \right\| \leq K \|x_k - \alpha\|^2, \ \forall k \geq 0.$$

Example 5 *Solve the system*

$$\begin{cases} x_1^3 + 3x_2^2 - 21 = 0 \\ x_1^2 + 2x_2 + 2 = 0 \end{cases}$$

using $x^{(0)} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$, $\varepsilon = 10^{-6}$.

We have

$$x^{(p+1)} = x^{(p)} - J^{-1}(x^{(p)})f(x^{(p)}),\ p = 0, 1, ... \tag{5}$$

We compute

$$J(x) = f'(x) = \left(\frac{\partial}{\partial x_i} f_i(x)\right)_{i,\ j=1,...,n} = \begin{pmatrix} 3x_1^2 & 6x_2 \\ 2x_1 & 2 \end{pmatrix}$$

$$f = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} = \begin{pmatrix} x_1^3 + 3x_2^2 - 21 \\ x_1^2 + 2x_2 + 2 \end{pmatrix}.$$

We have

$$x^{(p+1)} = x^{(p)} - J^{-1}(x^{(p)})f(x^{(p)}), \tag{6}$$

i.e.,

$$\begin{aligned} x^{(1)} &= \begin{pmatrix} 1 \\ -1 \end{pmatrix} - \begin{pmatrix} 3 & -6 \\ 2 & 2 \end{pmatrix}^{-1} \begin{pmatrix} 1 + 3 - 21 \\ 1 - 2 + 2 \end{pmatrix} \\ &= \begin{pmatrix} 1 \\ -1 \end{pmatrix} - \begin{pmatrix} 0.(1) & 0.(3) \\ -0.(1) & 0.1(6) \end{pmatrix} \begin{pmatrix} -17 \\ 1 \end{pmatrix} \end{aligned}$$

and it follows

$$x^{(1)} \approx \begin{pmatrix} 2.55 \\ -3.05 \end{pmatrix}.$$

Continuing in this way we obtain the approx. solution $\begin{pmatrix} 1.64 \\ -2.35 \end{pmatrix}$

5.3.3. Broyden's method for solving nonlinear systems

One disadvantage of Newton's method for solving nonlinear systems of equations is the need to compute the Jacobian matrix and to solve a linear system of size $n \times n$ that involves this matrix.

To illustrate this disadvantage, we will evaluate the computational effort associated with one iteration of Newton's method: the Jacobian matrix associated with a nonlinear system $f(x) = 0$, of size n, requires evaluating the n^2 partial derivatives of the n components of the function f . The computational effort for one Newton iteration involves at

least n^2+n scalar function evaluations with n^2 for the Jacobian and n for f , and $\mathcal{O}(n^3)$ arithmetic operations to solve the nonlinear system.

Thus, it is reasonable to focus on reducing the number of evaluations and avoiding solving a linear system at each step. Broyden's method avoids the complete recalculation of the Jacobian matrix at each iteration, replacing it with an approximation that is efficiently updated at each step. Therefore, in Broyden's method, the computational cost is significantly reduced compared to the classical Newton method.

Let

$$f(x) = 0, \quad f : \Omega \to \mathbb{R}^n$$

$$\begin{cases} f_1(x_1, \ldots, x_n) = 0 \\ \vdots \\ f_n(x_1, \ldots, x_n) = 0 \end{cases}$$

$$x^{(k+1)} = x^{(k)} - [f'(x^{(k)})]^{-1} f(x^{(k)})$$

$$x^{(k+1)} = x^{(k)} + w^{(k)}$$

$$w^{(k)} = -[f'(x^{(k)})]^{-1} f(x^{(k)}).$$

Note that $w^{(k)}$ is the solution of a system with n equations and k unknowns: $f'(x^{(k)})w^{(k)} = -f(x^{(k)})$

Broyden's method is a secant type method and is one of the most efficient methods for solving nonlinear equations.

Broyden's Algorithm

The initial approximation to the Jacobian is denoted as B_0.

1. Solve $B_k S_k = -f(x_k)$ for S_k

2. $x_{k+1} = x_k + S_k$

3. $y_k = f(x_{k+1}) - f(x_k)$

4. $B_{k+1} = B_k + \frac{(y_k - B_k S_k) S_k^T}{S_k^T S_k}$

Step 4. is motivated by the fact that B_{k+1} changes according to the secant method rule:

$$B_{k+1}(x_{k+1} - x_k) = f(x_{k+1}) - f(x_k).$$

This ensures that the sequence of matrices B_k retains information about the behavior of the function f along various directions generated by the algorithm, without the need to evaluate f again to get derivative information.

To find the values of B_k, one needs to solve a linear system with a computational cost of $\mathcal{O}(n^3)$. In practice, one uses a factorization of B_k rather than the entire matrix.

Example 6 *Solve the system* $f(x) = \begin{bmatrix} x_1 + 2x_2 - 2 \\ x_1^2 + 4x_2^2 - 4 \end{bmatrix}$, $f(x) = 0$.

Sol. Let $x_0 = \begin{bmatrix} 1 & 2 \end{bmatrix}^T, f(x_0) = \begin{bmatrix} 3 & 13 \end{bmatrix}^T$

$B_0 = J(f(x_0)) = \begin{bmatrix} 1 & 2 \\ 2 & 16 \end{bmatrix}$.

Solving the system $\begin{bmatrix} 1 & 2 \\ 2 & 16 \end{bmatrix} s_0 = \begin{bmatrix} -3 \\ -13 \end{bmatrix}$,

we get $s_0 = \begin{bmatrix} -1.83 & -0.58 \end{bmatrix}^T$, so

$x_1 = x_0 + s_0 = \begin{bmatrix} -0.83 \\ 1.42 \end{bmatrix}$

We have $f(x_1) = \begin{bmatrix} 0 \\ 4.72 \end{bmatrix}, y_0 = \begin{bmatrix} -3 \\ -8.28 \end{bmatrix}$

$$B_1 = \begin{bmatrix} 1 & 2 \\ 2 & 16 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -2.34 & -0.74 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ -0.34 & 15.3 \end{bmatrix}$$

Solving the system $\begin{bmatrix} 1 & 2 \\ -0.34 & 15.3 \end{bmatrix} s_1 = \begin{bmatrix} 0 \\ -4.72 \end{bmatrix}$

we get that $s_1 = \begin{bmatrix} 0.59 & -0.30 \end{bmatrix}^T$ so

$$x_2 = x_1 + s_1 = \begin{bmatrix} -0.24 \\ 1.120 \end{bmatrix}.$$

We have $f(x_2) = \begin{bmatrix} 0 \\ 1.08 \end{bmatrix}, y_1 = \begin{bmatrix} 0 \\ -3.64 \end{bmatrix}$

So $B_2 = \begin{bmatrix} 1 & 2 \\ -0.34 & 15.3 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -1.46 & -0.73 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 1.12 & 14.5 \end{bmatrix}$

The iterations converge to the solution $x^* = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$.

Iteration	x_k	y_k
1	1.0000e+00	2.0000e+00
2	-8.3333e-01	1.4167e+00
3	-2.4060e-01	1.1203e+00
4	-6.5226e-02	1.0326e+00
5	-6.8059e-03	1.0034e+00
6	-2.1425e-04	1.0001e+00
7	-7.2652e-07	1.0000e+00

6. Numerical methods for solving differential equations

We consider a Cauchy problem:

$$\begin{aligned} y' &= f(t,y) \\ y(t_0) &= y_0 \end{aligned} \tag{7}$$

with f defined on $D = \{(t,y) \in \mathbb{R}^2 \mid |t-t_0| \leq a,\ |y-y_0| \leq b\}$, $a, b \in \mathbb{R}_+$, continuous and derivable.

6.1. Taylor interpolation method

Let $f \in C^p(D)$ and y be a solution of the problem (7). We attach Taylor interpolation formula to y, with respect to t_0:

$$y = T_p y + R_p y,$$

where

$$(T_p y)(t) = y(t_0) + \frac{t-t_0}{1!} y'(t_0) + \ldots + \frac{(t-t_0)^p}{p!} y^{(p)}(t_0),$$

and the remainder term:

$$(R_p y)(t) = \frac{(t-t_0)^{p+1}}{(p+1)!} y^{(p+1)}(\xi), \quad \xi \in (t_0, t). \tag{8}$$

We know only $y(t_0) = y_0$ and $y'(t_0) = f(t_0, y_0)$ in this polynomial, so we have to compute $y^{(k)}(t_0)$, $k = 2, ..., p$. Using equation (7) we get

$$y'' = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial y} y', ...$$

and so on. Taking the values of these derivatives in t_0, the approximation of y is completely determined.

Denoting $y^{(k)} = f^{(k-1)}$, Taylor polynomial can be written as

$$\begin{aligned}(T_p y)(t) =& y(t_0) + \frac{(t-t_0)}{1!} f(t_0, y(t_0)) + \frac{(t-t_0)^2}{2!} f'(t_0, y(t_0)) \\ &+ ... + \frac{(t-t_0)^p}{p!} f^{(p-1)}(t_0, y(t_0)).\end{aligned} \tag{9}$$

If $\left|y^{(p+1)}(t)\right| \leq M_{p+1}$, we get the error delimitation:

$$|y(t) - (T_p y)(t)| \leq \frac{|t-t_0|^{p+1}}{(p+1)!} M_{p+1}, \ t \in I. \tag{10}$$

So, we've proved the following theorem:

Theorem 7 *If $f \in C^p(D)$ then the solution y of the Cauchy problem (7) can be approximated by Taylor polynomial (9) with delimitation of the error given in (10).*

Remark 8 *Disadvantage: for large values of p, the derivatives $f^{(k)}$, $k = 1, ..., p$, are more and more complicated to compute. It is very used in practical applications for small values of p.*

For the equidistant points $t_i = t_0 + ih$, $i = 0, ..., N$; $N \in \mathbb{N}$; $h = \frac{b-a}{N}$, **Taylor interpolation method of order** n can be written as

$$y_{i+1} = y_i + hT_n(t_i, y_i), \tag{11}$$

with

$$T_n(t_i, y_i) = f(t_i, y_i) + \frac{h}{2!}f'(t_i, y_i) + ... + \frac{h^{n-1}}{n!}f^{(n-1)}(t_i, y_i). \tag{12}$$

Example 9 *Apply Taylor's method of orders (a) two and (b) four with $N = 10$ to the initial-value problem $y' = y - t^2 + 1$, $0 \leq t \leq 2$, $y(0) = 0.5$.*

Sol. (a) For the method of order $n = 2$,

$$T_2(t_i, y_i) = f(t_i, y_i) + \frac{h}{2} f'(t_i, y_i),$$

we need the first derivative of $f(t, y(t)) = y(t) - t^2 + 1$ with respect to the variable t. Because $y' = y - t^2 + 1$ we have

$$f'(t, y(t)) = \frac{d}{dt}(y(t) - t^2 + 1) = y' - 2t = y - t^2 + 1 - 2t$$

so

$$\begin{aligned} T_2(t_i, y_i) &= f(t_i, y_i) + \frac{h}{2} f'(t_i, y_i) = y_i - t_i^2 + 1 + \frac{h}{2}(y_i - t_i^2 + 1 - 2t_i) \\ &= (1 + \frac{h}{2})(y_i - t_i^2 + 1) - ht_i \end{aligned}$$

and

$$y_{i+1} = y_i + hT_2(t_i, y_i).$$

As $N = 10$ we have $h = 0.2$, and $t_i = 0.2i$ for each $i = 1, 2, ..., 10$.

Thus the second-order method becomes

$$\begin{aligned} y_0 &= 0.5, \\ y_{i+1} &= y_i + h\left[(1+\frac{h}{2})(y_i - t_i^2 + 1) - ht_i\right] \\ &= y_i + 0.2\left[(1+\frac{0.2}{2})(y_i - 0.04i^2 + 1) - 0.04i\right] \end{aligned}$$

and

$$\begin{aligned} y(0.2) &\approx y_1 = 0.83 \\ y(0.4) &\approx y_2 = 1.21 \\ &\ldots \end{aligned}$$

(b) For Taylor's method of order four we need the first three derivatives of $f(t, y(t))$ with respect to t. Again using $y' = y - t^2 + 1$ we have

$$f'(t, y(t)) = y - t^2 + 1 - 2t$$

$$\begin{aligned} f''(t, y(t)) &= \frac{d}{dt}(y - t^2 + 1 - 2t) = f'(t, y(t)) = y' - 2t - 2 \\ &= y - t^2 + 1 - 2t - 2 = y - t^2 - 2t - 1 \end{aligned}$$

$$f'''(t,y(t)) = \frac{d}{dt}(y - t^2 - 2t - 1) = y' - 2t - 2$$
$$= y - t^2 - 2t - 1$$

so

$$T_4(t_i, y_i) = f(t_i, y_i) + \frac{h}{2}f'(t_i, y_i) + \frac{h^2}{6}f''(t_i, y_i) + \frac{h^3}{24}f'''(t_i, y_i)$$
$$= y_i - t_i^2 + 1 + \frac{h}{2}(y_i - t_i^2 + 1 - 2t_i) + \frac{h^2}{6}(y - t^2 - 2t - 1)$$
$$+ \frac{h^3}{24}(y - t^2 - 2t - 1)$$
$$= (1 + \frac{h}{2} + \frac{h^2}{6} + \frac{h^3}{24})(y_i - t_i^2) - (1 + \frac{h}{3} + \frac{h^2}{12})(ht_i)$$
$$+ 1 + \frac{h}{2} - \frac{h^2}{6} - \frac{h^3}{24}$$

and Taylor method of fourth order is

$$y_0 = 0.5,$$
$$y_{i+1} = y_i + hT_4(t_i, y_i), \quad i = 0, 1, ..., N - 1$$

with

$$y(0.2) \approx y_1 = 0.829$$
$$y(0.4) \approx y_2 = 1.214$$
$$\ldots$$

Comparing these results with those of Taylor's method of order 2 we have that the fourth-order results are vastly superior.

Suppose we need to determine an approximation to an intermediate point, for example, at $t = 1.25$. If we use linear interpolation on the Taylor method of order four approximations at $t = 1.2$ and $t = 1.4$, we have

$$y(1.25) \approx \left(\frac{1.25 - 1.4}{1.2 - 1.4}\right) 3.17 + \left(\frac{1.25 - 1.2}{1.4 - 1.2}\right) 3.73 = 3.31.$$

The true value is $y(1.25) = 3.3173285$, so this approximation has an error of 0.0007525.

We can significantly improve the approximation by using cubic Hermite interpolation. To determine this approximation for $y(1.25)$ requires

approximations to $y'(1.2)$ and $y'(1.4)$, as well as approximations to $y(1.2) = 3.17$ and $y(1.4) = 3.73$. The derivative approximations are available from the differential equation, because $y\prime(t) = f(t, y(t))$, i.e.,

$$y'(t) = y(t) - t^2 + 1,$$

so $y'(1.2) = y(1.2) - (1.2)^2 + 1 \approx 2.739$ and $y'(1.4) = y(1.4) - (1.4)2 + 1 \approx 2.772$. Divided difference table:

t	$y(t)$			
1.2	3.17	2.73	0.111	-0.307
1.2	3.17	**2.76**	0.050	
1.4	3.73	2.77		
1.4	3.73			

The cubic Hermite polynomial is

$$\begin{aligned} y(t) &\approx 3.17 + (t - 1.2)2.73 + (t - 1.2)^2 0.111 + (t - 1.2)^2(t - 1.4)(-0.307) \\ &= 3.317 \end{aligned}$$

a result that is accurate to within 0.0000286. This is about the average of the errors at 1.2 and at 1.4, and only 4% of the error obtained using linear interpolation. This improvement in accuracy certainly justifies the added computation required for the Hermite method.

6.2. Euler's method

Consider the equation (7) and an equidistant partition of the interval $[a, b]$: $t_i = t_0 + ih$, $h = \frac{b-a}{N}$; $i = 0, 1, ..., N$; $N \in \mathbb{N}^*$.

Remark 10 *The method (11), for* $n = 1$, *i.e.,*

$$y_{i+1} = y_i + hf(t_i, y_i), \ i = 0, 1, ..., N$$

is called **Euler's method.**

Geometrical interpretation: by Euler's method, the graph of the solution y is approximated by the polygonal line with vertices (t_k, y_k), $k = 0, 1, ...,$ hence this method is also called *the method of polygonal lines.*

Definition 11 *A function* $f(t, y)$ *is said to satisfy a Lipschitz condition in the variable* y *on a set* $D \subset \mathbb{R}^2$, *if a constant* $L > 0$ *exists with*

$$|f(t, y_1) - f(t, y_2)| \leq L|y_1 - y_2|,$$

whenever(t, y_1) *and* (t, y_2) *are in* D. *The constant L is called a Lipschitz constant for* f.

Theorem 12 *Let* $y(t)$ *be solution of the problem*

$$y' = f(t, y),\ a \leq t \leq b,\ y(a) = \alpha,$$

and $y_0, y_1, ..., y_n$ *the approximations generated by Euler's method for the parameters* N, $t_i = a + ih$, $i = 0, ..., N$, $h = \frac{b-a}{N}$.

Suppose that f *is Lipschitz continuous on* D, *of constant* L, *where* $D = \{(t, y) : a \leq t \leq b, -\infty < y < \infty\}$, *i.e.,*

$$|f(t, y_1) - f(t, y_2)| \leq L|y_1 - y_2|,\ \ \forall (t, y_1)(t, y_2) \in D.$$

If there exists the constant M *such that*

$$\left|y''(t)\right| \leq M,\ \ \forall t \in [a, b],$$

then

$$|y(t_i) - y_i| \leq \frac{hM}{2L}\left(e^{L(t_i - a)} - 1\right),\ \ i = 0, 1, ..., N.$$

Algorithm for Euler's method:

$h \leftarrow \frac{b-a}{N}$

$\alpha \leftarrow y_0$

for $i = 0, 1, ..., N-1$

$$y_{i+1} \leftarrow y_i + hf(t_i, y_i)$$

end

Note that y_i is a close approximation of $y(t_i)$.

Example 13 *Approximate the solution of the Cauchy problem:*

$$y'(t) = 2t - y$$
$$y(0) = -1$$

on the equidistant nodes $t_i = a + ih$, $i = 0, ..., N; h = \frac{b-a}{N}$, *with* $a = 0$, $b = 1$, $N = 10$, *using Euler's method.*

Solution. We have $h = \frac{1}{10}$, $f(t, y) = 2t - y$ and we get

$$\begin{aligned} y(0.1) &\approx y_1 = y_0 + 0.1f(0, -1) = -0.9 \\ y(0.2) &\approx y_2 = y_1 + 0.1f(0.1, -0.9) = -0.79 \\ &\dots \\ y_{10} &= 0.348678. \end{aligned}$$

Example 14 *Use Euler's method to approximate the solution to*

$$y' = y - t^2 + 1, \quad 0 \le t \le 2, y(0) = 0.5,$$

at $t = 2$ *with* $h = 0.5$.

Sol. For this problem $f(t,y) = y - t^2 + 1$, so

$$\begin{aligned} y_0 &= y(0) = 0.5; \\ y_1 &= y_0 + 0.5(y_0 - (0.0)^2 + 1) = 1.25 \\ y_2 &= y_1 + 0.5(y_1 - (0.5)^2 + 1) = 2.25 \\ y_3 &= y_2 + 0.5(y_2 - (1.0)^2 + 1) = 3.375 \\ y(2) \approx y_4 &= y_3 + 0.5(y_3 - (1.5)^2 + 1) = 4.4375. \end{aligned}$$

Example 15 *Consider the initial-value problem* $y = y - t^2 + 1, 0 \le t \le 2, y(0) = 0.5$, *with* $h = 0.2$. *Use the inequality in Theorem 12 to find a bounds for the approximation errors and compare these to the actual errors.*

Sol. Because $f(t,y) = y - t^2 + 1$, we have

$$\frac{\partial f(t,y)}{\partial y} = 1, \text{for all } y$$

so $L = 1$. For this problem, the exact solution is $y(t) = (t+1)^2 - 0.5e^t$, so $y''(t) = 2 - 0.5e^t$

$$|y''(t)| \le 0.5e^2 - 2 := M, \quad t \in [0,2].$$

By Theorem 12 we get

$$|y(t_i) - y_i| \leq \frac{hM}{2L}\left(e^{L(t_i-a)} - 1\right) = 0.1(0.5e^2 - 2)\left(e^{t_i} - 1\right)$$

hence

$$|y(0.2) - y_1| \leq 0.1(0.5e^2 - 2)\left(e^{0.2} - 1\right) = 0.037,$$

$$\ldots$$

For example, at $t_1 = 0.2$ the actual error is 0.029. Note that even though the true bound for the second derivative of the solution was used, the error bound is considerably larger than the actual error, especially for increasing values of t.

The principal importance of the error-bound formula given in Theorem is that the bound depends linearly on the step size h. Consequently, diminishing the step size should give correspondingly greater accuracy to the approximations.

6.3. Runge-Kutta methods

There are determined the values a_1, α_1, β_1 such that $a_1 f(t+\alpha_1, y+\beta_1)$ to approximate Taylor polynomial, given by (12), with error $O\left(h^2\right)$, that is the error for Taylor method of second order,

$$T_2(t,y) = f(t,y) + \frac{h}{2} f'(t,y), \quad h = \frac{b-a}{N}, N\text{-given.}$$

We have

$$f'(t,y) = \frac{\partial f}{\partial t}(t,y) + \frac{\partial f}{\partial t}(t,y) \cdot y'(t)$$

so

$$T_2(t,y) = f(t,y) + \frac{h}{2}\frac{\partial f}{\partial t}(t,y) + \frac{h}{2}\frac{\partial f}{\partial y}(t,y) y'(t).$$

Expanding $f(t+\alpha_1, y+\beta_1)$ using Taylor series we get

$$\begin{aligned} a_1 f(t+\alpha_1, y+\beta_1) = a_1 f(t,y) + a_1\alpha_1 \frac{\partial f}{\partial t}(t,y) + a_1\beta_1 \frac{\partial f}{\partial y}(t,y) \\ + a_1 (R_1 f)(t+\alpha_1, y+\beta_1). \end{aligned}$$

Indentifying the coefficients of the terms $f(t,y)$, $\frac{\partial f}{\partial t}(t,y)$ and $\frac{\partial f}{\partial y}(t,y)$, we obtain

$$\begin{aligned} a_1 &= 1 \\ \alpha_1 &= \frac{h}{2} \\ \beta_1 &= \frac{h}{2} y'(t) = \frac{h}{2} f(t,y). \end{aligned}$$

In (11), for $n = 2$ we have

$$y_{i+1} = y_i + hT_2(t_i, y_i),$$

and replacing T_2 by $a_1 f(t + \alpha_1, y + \beta_1)$, it is obtained **"the midpoint method":**

$$\begin{aligned} y_0 &= \alpha \\ y_{i+1} &= y_i + hf\left(t_i + \frac{h}{2}, y + \frac{h}{2} f(t_i, y_i)\right), \quad i = 0, ..., N-1 \end{aligned}$$

For approximating Taylor polynomial of third order

$$T_3(t,y) = f(t,y) + \frac{h}{2} f'(t,y) + \frac{h^2}{6} f''(t,y),$$

the best suitable expression with four parameters is

$$a_1 f(t,y) + a_2 f(t+\alpha_2, y+\delta_2 f(t,y)), \tag{13}$$

that is not enough flexible to allow the indetification of the term $\frac{h^2}{6}\left(\frac{\partial f}{\partial y}(t,y)\right)^2 f(t,y)$ that appears in expansion of $\frac{h^2}{6} f''(t,y)$.

So, by expressions of the form (13) can be deduced methods of maximum second order. Being 4 parameters there is the possibility to get more methods. These are called **Runge-Kutta methods** of second order:

Modified Euler's method, for $a_1 = a_2 = \frac{1}{2}$, $\alpha_2 = \delta_2 = h$:

$$y_{i+1} = y_i + \frac{h}{2}\left[f(t_i, y_i) + f\left(t_{i+1}, y_i + hf(t_i, y_i)\right)\right], \quad i = 0, ..., N-1$$

Heun's method, for $a_1 = \frac{1}{4}$, $a_2 = \frac{3}{4}$, $\alpha_2 = \delta_2 = \frac{2}{3}h$:

$$y_{i+1} = y_i + \frac{h}{4}\left[f(t_i, y_i) + 3f\left(t_i + \frac{2}{3}h, y_i + \frac{2}{3}hf(t_i, y_i)\right)\right], \quad i = 0, ..., N-1$$

Midpoint method (previously obtained)

$$y_{i+1} = y_i + hf\left(t_i + \frac{h}{2}, y + \frac{h}{2} f\left(t_i, y_i\right)\right), \ i = 0, ..., N-1$$

Taylor's polynomial of third order $T_3(t, y)$ can be approximated with error $O\left(h^3\right)$ by an expression of the following form

$$f\left(t + \alpha_1, y + \delta_1 f\left(t + \alpha_2, y + \delta_2 f\left(t, y\right)\right)\right),$$

with 4 parameters to obtain third order Runge-Kutta methods.

Runge-Kutta method of fourth order (most used in practice):

$$
\begin{aligned}
y_0 &= \alpha \\
k_1 &= f\left(t_i, y_i\right) \\
k_2 &= f\left(t_i + \frac{h}{2}, y_i + \frac{1}{2}k_1\right) \\
k_3 &= f\left(t_i + \frac{h}{2}, y_i + \frac{1}{2}k_2\right) \\
k_4 &= f\left(t_{i+1}, y_i + k_3\right) \\
y_{i+1} &= y_i + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right), \qquad i = 0, ..., N-1.
\end{aligned}
$$

Algorithm for Runge-Kutta method of 4-th order:

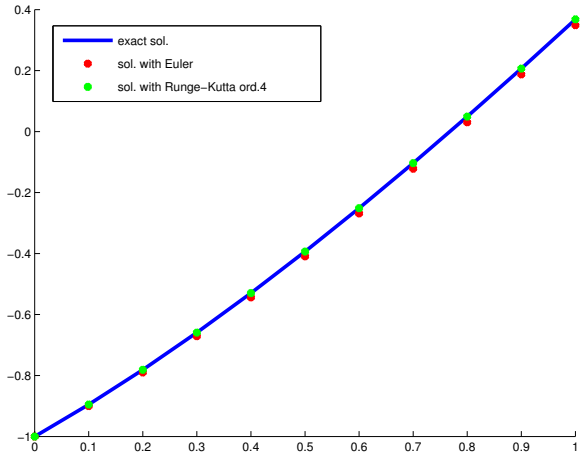
$h \leftarrow \frac{b-a}{N}$; $y_0 \leftarrow \alpha$

for $i = 0, ..., N$

$$
\begin{aligned}
k_1 &= f(t_i, y_i) \\
k_2 &= f\left(t_i + \frac{h}{2}, y_i + \frac{1}{2}k_1\right) \\
k_3 &= f\left(t_i + \frac{h}{2}, y_i + \frac{1}{2}k_2\right) \\
k_4 &= f\left(t_{i+1}, y_i + k_3\right) \\
y_{i+1} &= y_i + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)
\end{aligned}
$$

end

For the data from Example 13 we have the following graph. (The exact solution is $y(t) = e^{-t} + 2t - 2$.)



Example 16 *Use Runge-Kutta methods of order 2 and 4 to obtain an approximation to the solution of* $y' = 2t - y, y(0) = -1$, *with* $N = 10, h = 0.1$.