COURSE 10

4.2.3. Factorization methods - LU methods

The matrix A can be factored into the product of a lower triangular matrix L and an upper triangular matrix U, namely $A = LU$.

$$Ax = b \Longleftrightarrow LUx = b,$$

where

$$L = \begin{pmatrix} l_{11} & 0 & \dots & 0 \\ l_{21} & l_{22} & \dots & 0 \\ \vdots & & & \\ l_{n1} & l_{n2} & \dots & l_{nn} \end{pmatrix} \qquad U = \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ \vdots & & & \\ 0 & 0 & \dots & u_{nn} \end{pmatrix}.$$

We solve the systems in two stages:

First stage: Solve $Lz = b$,

Second stage: Solve $Ux = z$.

Methods for computing matrices L and U : **Doolittle method** where all diagonal elements of L have to be 1; **Crout method** where all diagonal elements of U have to be 1 and **Choleski method** where $l_{ii} = u_{ii}$ for $i = 1, ..., n$.

Remark 1 *LU factorizations are modified forms of Gauss elimination method.*

Doolittle method

We consider $a_{kk} \neq 0$, $k = \overline{1, n-1}$. Denote

$$l_{i,k} := \frac{a_{i,k}^{(k-1)}}{a_{k,k}^{(k-1)}}, \quad i = \overline{k+1, n}$$

$$t^{(k)} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ l_{k+1,k} \\ \cdots \\ l_{n,k} \end{bmatrix},$$

having zeros for the first k-th lines, and

$$M_k = I_n - t^{(k)} e_k \in \mathcal{M}_{n\times n}(\mathbb{R}) \tag{1}$$

where $e_k = \begin{pmatrix} 0 & ... & 1 & ... & 0 \end{pmatrix}$ is the k-unit vector of dimension n,(has 1 on the k-th position) and $I_n = \begin{pmatrix} 1 & 0 & ... & 0 \\ 0 & 1 & ... & 0 \\ ... & & & \\ 0 & 0 & ... & 1 \end{pmatrix}$ is the identity matrix of order n.

$a_{i,k}^{(0)}$ are elements of A; $a_{i,k}^{(1)}$ are elements of $M_1 \cdot A$; ...; $a_{i,k}^{(k-1)}$ are elements of $M_{k-1} \ldots \cdot M_1 \cdot A$.

Definition 2 *The matrix M_k is called* **the Gauss matrix**, *the components $l_{i,k}$ are called* **the Gauss multiplies** *and the vector $t^{(k)}$ is* **the Gauss vector**.

Remark 3 *If $A \in \mathcal{M}_{n\times n}(\mathbb{R})$, then the Gauss matrices $M_1, \ldots, M_{n-1}$ can be determined such that*

$$U = M_{n-1} \cdot M_{n-2} \ldots M_2 \cdot M_1 \cdot A$$

is an upper triangular matrix. Moreover, if we choose

$$L = M_1^{-1} \cdot M_2^{-1} \dots M_{n-1}^{-1}$$

then

$$A = L \cdot U.$$

Example 4 *Find LU factorization for the matrix*

$$A = \begin{pmatrix} 2 & 1 \\ 6 & 8 \end{pmatrix}.$$

Solve the system $\left\{ \begin{array}{c} 2x_1 + x_2 = 3 \\ 6x_1 + 8x_2 = 9 \end{array} \right.$.

Sol.

$$M_1 = I_2 - t^{(1)} e_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 \\ \frac{6}{2} \end{pmatrix} \begin{pmatrix} 1 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ 3 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -3 & 1 \end{pmatrix}.$$

We have

$$U = M_1 A = \begin{pmatrix} 1 & 0 \\ -3 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 \\ 6 & 8 \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 0 & 5 \end{pmatrix}$$

$$L = M_1^{-1} = \begin{pmatrix} 1 & 0 \\ 3 & 1 \end{pmatrix}.$$

So

$$A = \begin{pmatrix} 2 & 1 \\ 6 & 8 \end{pmatrix} = L \cdot U = \begin{pmatrix} 1 & 0 \\ 3 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 \\ 0 & 5 \end{pmatrix}.$$

We have

$$L \cdot U \cdot x = \begin{pmatrix} 3 \\ 9 \end{pmatrix}$$

$$Ux = z$$

and

$$\begin{pmatrix} 1 & 0 \\ 3 & 1 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} 3 \\ 9 \end{pmatrix} \Rightarrow z = \begin{pmatrix} 3 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 1 \\ 0 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \end{pmatrix} \Rightarrow x = \begin{pmatrix} 1.5 \\ 0 \end{pmatrix}.$$

4.3. Iterative methods for solving linear systems

Because of round-off errors, direct methods become less efficient than iterative methods for large systems (>100 000 variables).

An iterative scheme for linear systems consists of converting the system

$$Ax = b \tag{2}$$

to the form

$$x = \tilde{b} - Bx.$$

After an initial guess for $x^{(0)}$, the sequence of approximations of the solution $x^{(0)}, x^{(1)}, ..., x^{(k)}, ...$is generated by computing

$$x^{(k)} = \tilde{b} - Bx^{(k-1)}, \quad \text{for } k = 1, 2, 3, ...$$

4.3.1. Jacobi iterative method

Consider the $n \times n$ linear system,

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n = b_2 \\ \qquad\qquad\qquad ... \\ a_{n1}x_1 + a_{n2}x_2 + ... + a_{nn}x_n = b_n, \end{cases}$$

where we assume that the diagonal terms a_{11}, a_{22}, ..., a_{nn} are all nonzero.

We begin our iterative scheme by solving each equation for one of the variables:

$$\begin{cases} x_1 = u_{12}x_2 + ... + u_{1n}x_n + c_1 \\ x_2 = u_{21}x_1 + ... + u_{2n}x_n + c_2 \\ ... \\ x_n = u_{n1}x_1 + ... + u_{nn-1}x_{n-1} + c_n, \end{cases}$$

where $u_{ij} = -\frac{a_{ij}}{a_{ii}}$, $c_i = \frac{b_i}{a_{ii}}$, $i = 1, ..., n$.

Let $x^{(0)} = (x_1^{(0)}, x_2^{(0)}, ..., x_n^{(0)})$ be an initial approximation of the solution. The $k+1$-th approximation is:

$$\begin{cases} x_1^{(k+1)} = u_{12}x_2^{(k)} + ... + u_{1n}x_n^{(k)} + c_1 \\ x_2^{(k+1)} = u_{21}x_1^{(k)} + u_{23}x_3^{(k)} + ... + u_{2n}x_n^{(k)} + c_2 \\ ... \\ x_n^{(k+1)} = u_{n1}x_1^{(k)} + ... + u_{nn-1}x_{n-1}^{(k)} + c_n, \end{cases}$$

for $k = 0, 1, 2, ...$

An algorithmic form:

$$x_i^{(k)} = \frac{b_i - \sum\limits_{j=1, j\neq i}^{n} a_{ij}x_j^{(k-1)}}{a_{ii}}, \; i = 1, 2, ..., n, \text{ for } k \geq 1.$$

The iterative process is terminated when a convergence criterion is satisfied.

Stopping criterions: $\left|x^{(k)} - x^{(k-1)}\right| < \varepsilon$ or $\frac{\left|x^{(k)} - x^{(k-1)}\right|}{\left|x^{(k)}\right|} < \varepsilon$, with $\varepsilon > 0$ - a prescribed tolerance.

Example 5 *Solve the following system using the Jacobi iterative method. Use* $\varepsilon = 10^{-3}$ *and* $x^{(0)} = (0\ 0\ 0\ 0)$ *as the starting vector.*

$$\begin{cases} 7x_1 - 2x_2 + x_3 = 17 \\ x_1 - 9x_2 + 3x_3 - x_4 = 13 \\ 2x_1 + 10x_3 + x_4 = 15 \\ x_1 - x_2 + x_3 + 6x_4 = 10. \end{cases}$$

These equations can be rearranged to give

$$\begin{aligned} x_1 &= (17 + 2x_2 - x_3)/7 \\ x_2 &= (-13 + x_1 + 3x_3 - x_4)/9 \\ x_3 &= (15 - 2x_1 - x_4)/10 \\ x_4 &= (10 - x_1 + x_2 - x_3)/6 \end{aligned}$$

and, for example,

$$\begin{aligned} x_1^{(1)} &= (17 + 2x_2^{(0)} - x_3^{(0)})/7 \\ x_2^{(1)} &= (-13 + x_1^{(0)} + 3x_3^{(0)} - x_4^{(0)})/9 \\ x_3^{(1)} &= (15 - 2x_1^{(0)} - x_4^{(0)})/10 \\ x_4^{(1)} &= (10 - x_1^{(0)} + x_2^{(0)} - x_3^{(0)})/6. \end{aligned}$$

Substitute $x^{(0)} = (0,0,0,0)$ into the right-hand side of each of these equations to get

$$x_1^{(1)} = (17 + 2 \cdot 0 - 0)/7 = 2.428\ 571\ 429$$
$$x_2^{(1)} = (-13 + 0 + 3 \cdot 0 - 0)/9 = -1.444\ 444\ 444$$
$$x_3^{(1)} = (15 - 2 \cdot 0 - 0)/10 = 1.5$$
$$x_4^{(1)} = (10 - 0 + 0 - 0)/6 = 1.666\ 666\ 667$$

and so $x^{(1)} = (2.428\ 571\ 429, -1.444\ 444\ 444, 1.5, 1.666\ 666\ 667)$.
The Jacobi iterative process:

$$x_1^{(k+1)} = \left(17 + 2x_2^{(k)} - x_3^{(k)}\right)/7$$
$$x_2^{(k+1)} = \left(-13 + x_1^{(k)} + 3x_3^{(k)} - x_4^{(k)}\right)/9$$
$$x_3^{(k+1)} = \left(15 - 2x_1^{(k)} - x_4^{(k)}\right)/10$$
$$x_4^{(k+1)} = \left(10 - x_1^{(k)} + x_2^{(k)} - x_3^{(k)}\right)/6, \qquad k \geq 1.$$

We obtain a sequence that converges to

$$\mathbf{x}^{(9)} = (2.000127203, -1.000100162, 1.000118096, 1.000162172).$$

4.3.2. Gauss-Seidel iterative method

Almost the same as Jacobi method, except that each x-value is improved using the most recent approx. of the other variables.

For a $n \times n$ system, the $k+1$-th approximation is:

$$\begin{cases} x_1^{(k+1)} = u_{12}x_2^{(k)} + ... + u_{1n}x_n^{(k)} + c_1 \\ x_2^{(k+1)} = u_{21}x_1^{(k+1)} + u_{23}x_3^{(k)} + ... + u_{2n}x_n^{(k)} + c_2 \\ ... \\ x_n^{(k+1)} = u_{n1}x_1^{(k+1)} + ... + u_{nn-1}x_{n-1}^{(k+1)} + c_n, \end{cases}$$

with $k = 0, 1, 2, ...;$ $u_{ij} = -\frac{a_{ij}}{a_{ii}}$, $c_i = \frac{b_i}{a_{ii}}$, $i = 1, ..., n$ (as in Jacobi method).

Algorithmic form:

$$x_i^{(k)} = \frac{b_i - \sum\limits_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum\limits_{j=i+1}^{n} a_{ij}x_j^{(k-1)}}{a_{ii}}$$

for each $i = 1, 2, ...n,$ and for $k \geq 1$.

Stopping criterions: $\left| x^{(k)} - x^{(k-1)} \right| < \varepsilon$, or $\frac{\left| \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)} \right|}{\left| \mathbf{x}^{(k)} \right|} < \varepsilon$, with ε - a prescribed tolerance, $\varepsilon > 0$.

Remark 6 *Because the new values can be immediately stored in the location that held the old values, the storage requirements for $\mathbf{x}$ with the Gauss-Seidel method is half than that for Jacobi method and the rate of convergence is faster.*

Example 7 *Solve the following system using the Gauss-Seidel iterative method. Use $\varepsilon = 10^{-3}$ and $\mathbf{x}^{(0)} = (0\ 0\ 0\ 0)$ as the starting vector.*

$$\begin{cases} 7x_1 - 2x_2 + x_3 = 17 \\ x_1 - 9x_2 + 3x_3 - x_4 = 13 \\ 2x_1 + 10x_3 + x_4 = 15 \\ x_1 - x_2 + x_3 + 6x_4 = 10 \end{cases}$$

We have

$$
\begin{aligned}
x_1 &= (17 + 2x_2 - x_3)/7 \\
x_2 &= (-13 + x_1 + 3x_3 - x_4)/9 \\
x_3 &= (15 - 2x_1 - x_4)/10 \\
x_4 &= (10 - x_1 + x_2 - x_3)/6,
\end{aligned}
$$

and, for example,

$$
\begin{aligned}
x_1^{(1)} &= (17 + 2x_2^{(0)} - x_3^{(0)})/7 \\
x_2^{(1)} &= (-13 + x_1^{(1)} + 3x_3^{(0)} - x_4^{(0)})/9 \\
x_3^{(1)} &= (15 - 2x_1^{(1)} - x_4^{(0)})/10 \\
x_4^{(1)} &= (10 - x_1^{(1)} + x_2^{(1)} - x_3^{(1)})/6,
\end{aligned}
$$

which provide the following Gauss-Seidel iterative process:

$$x_1^{(k+1)} = \left(17 + 2x_2^{(k)} - x_3^{(k)}\right)/7$$
$$x_2^{(k+1)} = \left(-13 + x_1^{(k+1)} + 3x_3^{(k)} - x_4^{(k)}\right)/9$$
$$x_3^{(k+1)} = \left(15 - 2x_1^{(k+1)} - x_4^{(k)}\right)/10$$
$$x_4^{(k+1)} = \left(10 - x_1^{(k+1)} + x_2^{(k+1)} - x_3^{(k+1)}\right)/6, \quad \textit{for } k \geq 1.$$

Substitute $\mathbf{x}^{(0)} = (0,0,0,0)$ *into the right-hand side of each of these equations to get*

$$x_1^{(1)} = (17 + 2\cdot 0 - 0)/7 = 2.428\ 571\ 429$$
$$x_2^{(1)} = (-13 + 2.428\ 571\ 429 + 3\cdot 0 - 0)/9 = -1.1746031746$$
$$x_3^{(1)} = (15 - 2\cdot 2.428\ 571\ 429 - 0)/10 = 1.0142857143$$
$$x_4^{(1)} = (10 - 2.428\ 571\ 429 - 1.1746031746 - 1.0142857143)/6$$
$$= 0.8970899472$$

and so

$$\mathbf{x}^{(1)} = (2.428571429 - 1.1746031746, 1.0142857143, 0.8970899472).$$

Similar procedure generates a sequence that converges to

$$\mathbf{x}^{(5)} = (2.000025, -1.000130, 1.000020.0.999971).$$

4.3.3. Relaxation method

In case of convergence, the Gauss-Seidel method is faster than Jacobi method. The convergence can be more improved using **relaxation method (SOR method) (**SOR=Succesive Over Relaxation)

Algorithmic form of the method:

$$x_i^{(k)} = \frac{\omega}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)}\right) + (1-\omega)x_i^{(k-1)}$$

for each $i = 1, 2, ...n,$ and for $k \geq 1$.

For $0 < \omega < 1$ the procedure is called **under relaxation method,** that can be used to obtain convergence for systems which are not convergent by Gauss-Siedel method.

For $\omega > 1$ the procedure is called **over relaxation method,** that can be used to accelerate the convergence for systems which are convergent by Gauss-Siedel method.

By Kahan's Theorem follows that the method converges for $0 < \omega < 2$.

Remark 8 *For $\omega = 1$, relaxation method is Gauss-Seidel method.*

Example 9 *Solve the following system, using relaxation iterative method. Use $\varepsilon = 10^{-3}$, $\mathbf{x}^{(0)} = (1\ 1\ 1)$ and $\omega = 1.25$,*

$$\begin{array}{rcrcrcr} 4x_1 & + & 3x_2 & & & = & 24 \\ 3x_1 & + & 4x_2 & - & x_3 & = & 30 \\ & - & x_2 & + & 4x_3 & = & -24 \end{array}$$

We have

$$\begin{aligned} x_1^{(k)} &= 7.5 - 0.937x_2^{(k-1)} - 0.25x_1^{(k-1)} \\ x_2^{(k)} &= 9.375 - 9.375x_1^{(k)} + 0.3125x_3^{(k-1)} - 0.25x_2^{(k-1)} \\ x_3^{(k)} &= -7.5 + 0.3125x_2^{(k)} - 0.25x_3^{(k-1)}, \quad \textit{for } k \geq 1. \end{aligned}$$

The solution is $(3, 4, -5)$.

4.3.4 The matriceal formulations of the iterative methods

Split the matrix A into the sum

$$A = D + L + U,$$

where D is the diagonal of A, L the lower triangular part of A, and U the upper triangular part of A. That is,

$$D = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_{nn} \end{bmatrix}, \qquad L = \begin{bmatrix} 0 & \cdots & & 0 \\ a_{21} \vdots & & & \ddots \\ \vdots & \ddots & \ddots & \\ a_{n1} & \cdots & a_{n,n-1} & 0 \end{bmatrix},$$

$$U = \begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ \vdots & \ddots & \ddots & \vdots \\ & & \ddots & a_{n-1,n} \\ 0 & \cdots & & 0 \end{bmatrix}$$

The system $Ax = b$ can be written as

$$(D + L + U)\mathbf{x} = \mathbf{b}.$$

The **Jacobi method** in matriceal form is given by:

$$D\mathbf{x}^{(k)} = -(L+U)\mathbf{x}^{(k-1)} + \mathbf{b}$$

the **Gauss-Seidel method** in matriceal form is given by:

$$(D+L)\mathbf{x}^{(k)} = -U\mathbf{x}^{(k-1)} + \mathbf{b}$$

and **the relaxation method** in matriceal form is given by:

$$(D+\omega L)\mathbf{x}^{(k)} = ((1-\omega)D - \omega U)\mathbf{x}^{(k-1)} + \omega\mathbf{b}$$

Convergence of the iterative methods

Remark 10 *The convergence (or divergence) of the iterative process in the Jacobi and Gauss-Seidel methods does not depend on the initial guess, but depends only on the character of the matrices themselves. However, a good first guess in case of convergence will make for a relatively small number of iterations.*

A sufficient condition for convergence:

Theorem 11 (**Convergence Theorem**) *If A is strictly diagonally dominant, then the Jacobi, Gauss-Seidel and relaxation methods converge for any choice of the starting vector $\mathbf{x}^{(0)}$.*

Example 12 *Consider the system of equations*

$$\begin{bmatrix} 3 & 1 & 1 \\ -2 & 4 & 0 \\ -1 & 2 & -6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \\ 2 \end{bmatrix}.$$

The coefficient matrix of the system is strictly diagonally dominant since

$$\begin{aligned} |a_{11}| &= |3| = 3 > |1| + |1| = 2 \\ |a_{22}| &= |4| = 4 > |-2| + |0| = 2 \\ |a_{33}| &= |-6| = 6 > |-1| + |2| = 3. \end{aligned}$$

Hence, if the Jacobi or Gauss-Seidel method are used to solve the system of equations, they will converge for any choice of the starting vector $\mathbf{x}^{(0)}$.

Example 13 *Consider the linear system*

$$4x_1 + x_2 = 3$$
$$2x_1 + 5x_2 = 1.$$

Perform two iterations of Jacobi, Gauss-Seidel and relaxation methods to this system, beginning with the vector $x = [3, 11]$ *and for* $\omega = 1.25$.

(Solutions of the system are $7/9$ *and* $-1/9$*).*

5. Numerical methods for solving nonlinear equations in $\mathbb{R}$

The roots of the iterative methods are traced back to Egyptians and Babylonians (cc. 1800 B.C.).

Example. Kepler's Equation: consider a two-body problem like a satellite orbiting the earth or a planet revolving around the sun. Kepler discovered that the orbit is an ellipse and the central body F (earth, sun) is in a focus of the ellipse. The speed of the satellite P is not uniform: near the earth it moves faster than far away. It is used Kepler's law to predict where the satellite will be at a given time. If we want to know the position of the satellite for t = 9 minutes, then we have to solve the equation $f(E) = E - 0.8sinE - 2\pi/10 = 0$.

Let $f : \Omega \to \mathbb{R}$, $\Omega \subset \mathbb{R}$. Consider the equation

$$f(x) = 0, \quad x \in \Omega. \tag{3}$$

We attach a mapping $F : D \to D,\ D \subset \Omega^n$ to this equation.

Let $(x_0, ..., x_{n-1}) \in D$. Using F and the numbers $x_0, x_1, ..., x_{n-1}$ we construct iteratively the sequence

$$x_0, x_1, ..., x_{n-1}, x_n, ... \tag{4}$$

with

$$x_i = F\left(x_{i-n}, ..., x_{i-1}\right), \quad i = n, \ldots . \tag{5}$$

The problem consists in choosing F and $x_0, ..., x_{n-1} \in D$ such that the sequence (4) to be convergent to the solution of the equation (3).

Definition 14 *The procedure of approximation the solution of equation (3) by the elements of the sequence (4), computed as in (5), is called* F-**method.**

The numbers $x_0, x_1, ..., x_{n-1}$ *are called* **the starting points** *and the* k*-th element of the sequence (4) is called an approximation of* k*-th order of the solution.*

If the set of starting points has only one element then the F-method is **an one-step method;** if it has more than one element then the F-method is **a multistep method**.

Definition 15 *If the sequence (4) converges to the solution of the equation (3) then the F-method is convergent, otherwise it is divergent.*

Definition 16 *Let $\alpha \in \Omega$ be a solution of the equation (3) and let $x_0, x_1, ..., x_{n-1}, x_n, ...$ be the sequence generated by a given F-method. The number p having the property*

$$\lim_{x_i \to \alpha} \frac{\alpha - F(x_{i-n}, ..., x_i)}{(\alpha - x_i)^p} = C \neq 0, \quad C = constant,$$

is called the order *of the F-method.*

We construct some classes of F-methods based on the interpolation procedures.

Let $\alpha \in \Omega$ be a solution of the equation (3) and $V(\alpha)$ a neighborhood of α. Assume that f has inverse on $V(\alpha)$ and denote $g := f^{-1}$. Since

$$f(\alpha) = 0$$

it follows that

$$\alpha = g(0).$$

This way, the approximation of the solution α is reduced to the approximation of $g(0)$.

Definition 17 *The approximation of g by means of an interpolating method, and of α by the value of g at the point zero is called* **the inverse interpolation procedure.**

5.1. One-step methods

Let F be a one-step method, i.e., for a given x_i we have $x_{i+1} = F\left(x_i\right).$

Remark 18 *If* $p = 1$ *the convergence condition is* $\left|F'\left(x\right)\right| < 1$.

If $p > 1$ *there always exists a neighborhood of* α *where the* F*-method converges.*

All information on f are given at a single point, the starting value $\Rightarrow$ we are lead to Taylor interpolation.

Theorem 19 *Let* α *be a solution of equation (3),* $V(\alpha)$ *a neighborhood of* α*,* $x, x_i \in V(\alpha)$*,* f *fulfills the necessary continuity conditions. Then we have the following method, denoted by* F_m^T*, for approximating* α*:*

$$F_m^T(x_i) = x_i + \sum_{k=1}^{m-1} \frac{(-1)^k}{k!}[f(x_i)]^k g^{(k)}(f(x_i)), \tag{6}$$

where $g = f^{-1}$.

Proof. There exists $g = f^{-1} \in C^m[V(0)]$. Let $y_i = f(x_i)$ and consider Taylor interpolation formula

$$g(y) = (T_{m-1}g)(y) + (R_{m-1}g)(y),$$

with

$$(T_{m-1}g)(y) = \sum_{k=0}^{m-1} \tfrac{1}{k!} (y - y_i)^k \, g^{(k)}(y_i),$$

and $R_{m-1}g$ is the corresponding remainder.

Since $\alpha = g(0)$ and $g \approx T_{m-1}g$, it follows

$$\alpha \approx (T_{m-1}g)(0) = x_i + \sum_{k=1}^{m-1} \frac{(-1)^k}{k!} y_i^k g^{(k)}(y_i).$$

Hence,

$$x_{i+1} := F_m^T(x_i) = x_i + \sum_{k=1}^{m-1} \frac{(-1)^k}{k!} [f(x_i)]^k g^{(k)}(f(x_i))$$

is an approximation of α, and F_m^T is an approximation method for the solution α. ■

Concerning the order of the method F_m^T we state:

Theorem 20 *If* $g = f^{-1}$ *satisfies condition* $g^{(m)}(0) \neq 0$, *then* $\mathrm{ord}(F_m^T) = m$.

Remark 21 *We have an upper bound for the absolute error in approximating* α *by* x_{i+1} :

$$\left|\alpha - F_m^T(x_i)\right| \leq \frac{1}{m!}[f(x_i)]^m M_m g, \quad \textit{with } M_m g = \max_{y \in V(0)} \left|g^{(m)}(y)\right|.$$

Particular cases.

1) Case $m = 2$.

$$F_2^T(x_i) = x_i - \frac{f(x_i)}{f'(x_i)}.$$

This method is called **Newton's method** (**the tangent method**). Its order is 2.

2) Case $m = 3$.

$$F_3^T(x_i) = x_i - \frac{f(x_i)}{f'(x_i)} - \frac{1}{2}\left[\frac{f(x_i)}{f'(x_i)}\right]^2 \frac{f''(x_i)}{f'(x_i)},$$

with $\mathrm{ord}(F_3^T) = 3$. So, this method converges faster than F_2^T.

3) Case $m = 4$.

$$F_4^T(x_i) = x_i - \frac{f(x_i)}{f'(x_i)} - \frac{1}{2}\frac{f''(x_i)f^2(x_i)}{[f'(x_i)]^3} + \frac{\left(f'''(x_i)f'(x_i) - 3[f''(x_i)]^2\right)f^3(x_i)}{3![f'(x_i)]^5}.$$

Remark 22 *The higher the order of a method is, the faster the method converges. Still, this doesn't mean that a higher order method is more efficient (computation requirements). By the contrary, the most efficient are the methods of relatively low order, due to their low complexity (methods* F_2^T *and* F_3^T*).*