and then setting

$$x_{k+1} = x_k + \Delta x_k,$$

where $J(f)(x_k) = \left(\frac{\partial f_i}{\partial x_j}(x_k)\right)$ is the Jacobian matrix of f at x_k.

In general it is very costly to compute $J(f)(x_k)$ at each iteration and then to solve the corresponding linear system. If the method converges, the consecutive vectors x_k should differ only a little, as also the corresponding matrices $J(f)(x_k)$. Thus, we are led to several variants of Newton's method.

Variant 2. This variant consists in keeping the same matrix for p consecutive steps (where p is some fixed integer ≥ 2):

$$\begin{aligned}
x_{k+1} &= x_k - (f'(x_0))^{-1}(f(x_k)), && 0 \leq k \leq p-1\\
x_{k+1} &= x_k - (f'(x_p))^{-1}(f(x_k)), && p \leq k \leq 2p-1\\
&\vdots\\
x_{k+1} &= x_k - (f'(x_{rp}))^{-1}(f(x_k)), && rp \leq k \leq (r+1)p-1\\
&\vdots
\end{aligned}$$

Variant 3. Set $p = \infty$, that is, use *the same matrix $f'(x_0)$ for all iterations*, which leads to iterations of the form

$$x_{k+1} = x_k - (f'(x_0))^{-1}(f(x_k)), \quad k \geq 0,$$

Variant 4. Replace $f'(x_0)$ by a particular matrix A_0 which is easy to invert:

$$x_{k+1} = x_k - A_0^{-1} f(x_k), \quad k \geq 0.$$

In the last two cases, if possible, we use an LU factorization of $f'(x_0)$ or A_0 to speed up the method. In some cases, it may even possible to set $A_0 = I$.

The above considerations lead us to the definition of a *generalized Newton method*, as in Ciarlet [41] (Chapter 7). Recall that a linear map $f \in \mathcal{L}(E; F)$ is called an *isomorphism* iff f is continuous, bijective, and f^{-1} is also continuous.

Definition 41.1. If X and Y are two normed vector spaces and if $f\colon \Omega \to Y$ is a function from some open subset Ω of X, a *generalized Newton method* for finding zeros of f consists of

(1) A sequence of families $(A_k(x))$ of linear isomorphisms from X to Y, for all $x \in \Omega$ and all integers $k \geq 0$;

(2) Some starting point $x_0 \in \Omega$;