Newton-Raphson Algorithm

Algorithm 1: Newton-Raphson

input : $\mathbf{G}$, $\mathbf{F}(\mathbf{x})$, $\mathbf{b}_{DC}$ (MNA equations)
and $\mathbf{x}_0$ (Initial guess)

output : $\mathbf{x}_{DC}$ (DC solution)

$\mathrm{ErrTh} = 1e^{-15}$ *Deciding an Error-Threshold*;

$\mathrm{MaxItr} = 50$ *Maximum number of NR iterations*;

$\mathbf{x}^{(1)} = \mathbf{x}_0$ *Reading-in the initial guess*;

$\Delta\mathbf{x}^{(1)} = \mathbf{0}$ *Initialization*;

for $n < \mathrm{MaxItr}$ **do**

  $\mathbf{\Phi}^{(n)} = \mathbf{G}\mathbf{x}^{(n)} + \mathbf{F}(\mathbf{x}^{(n)}) - \mathbf{b}_{DC}$ *Compute the nonlinear DC equations*;

  if $\|\mathbf{\Phi}^{(n)}\| < \mathrm{ErrTh}$ & $\|\Delta\mathbf{x}^{(n)}\| < \mathrm{ErrTh}$ **then**

    $\mathbf{x}_{DC} = \mathbf{x}^{(n)}$;

    break; *Stops the **for** loop*;

  $\mathbf{J}^{(n)} = \mathbf{G} + \dfrac{\partial}{\partial\mathbf{x}}\mathbf{F}(\mathbf{x}^{(n)})$ *Compute the Jacobian*;

  $\Delta\mathbf{x}^{(n)} = -\left(\mathbf{J}^{(n)}\right)^{-1}\mathbf{\Phi}^{(n)}$ *using: **LUPQ***;

  $\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \Delta\mathbf{x}^{(n)}$;

return $\mathbf{x}_{DC}$;

Note:

- $\mathbf{x}^{(n)}$: denotes $\mathbf{x}$ in the n-th iteration,
- MaxItr : By limiting the maximum number for NR iterations, we avoid the possibility of going into an infinite loop when NR is not converging,