$$I_{Ph\_const\_irr} := I_{SC\_STC}\left(1+\frac{TC_{I\_SC}\left(\vartheta_C-\vartheta_{STC}\right)}{100}\right) \tag{3.1.3}$$

Open-circuit voltage with constant solar irradiance $\left(E_{\mathrm{G}}=E_{\mathrm{STC}}\right)$:

```
> U__OC_STC_const_irr := U__OC_STC * (1 + TC__U_OC/100 *
  (vartheta__C - vartheta__STC));
```

$$U_{OC\_STC\_const\_irr} := U_{OC\_STC}\left(1+\frac{TC_{U\_OC}\left(\vartheta_C-\vartheta_{STC}\right)}{100}\right) \tag{3.1.4}$$

Starting values for the jacobian matrix

Starting value for the open circuit voltage:

```
> U__OC_0 := evalf(eval(U__OC_STC_const_irr + m * N_C * U__T *
  ln(E__G/E__STC), param)):
```

Starting value for the reverse saturation current:

```
> I__S_0 := evalf(eval(I__Ph * exp(- U__OC_0 / (m * N_C * U__T)
  ), param)):
```

Jacobian matrix

Preparing the vector of functions and zero crossings for Jacobian matrix:

```
> f__1 := exp( (U__OC_theta_phi - U__OC_STC_const_irr)/(m * N_C
  * U__T) ) - (I__Ph - I__S_theta)/(I__Ph_const_irr -
  I__S_theta):
> f__2 := I__S_theta - I__Ph * (exp( U__OC_theta_phi/(m * N_C *
  U__T) ) - 1)^(-1):
> f := < f__1, f__2 >:
> x__R := < I__S_theta, U__OC_theta_phi >:
```

Jacobian matrix:

```
> J := Jacobian(convert(f, list), convert(x__R, list)):
```

Preparing the Jacobian matrix for the Newton-Raphson method:

```
> J__inv := MatrixInverse(J):
> J__inv_num := eval(J__inv, param):
> f__num := eval(f, param):
```

Iterations

```
> x__R_0 := < I__S_0, U__OC_0 >;
```

$$x_{R\_0} := \begin{bmatrix} 3.935566032\ 10^{-11} \\ 22.78137801 \end{bmatrix} \tag{3.4.1}$$

```
> x__R_1 := evalf( x__R_0 - eval( J__inv_num . f__num ,
  [I__S_theta = x__R_0(1), U__OC_theta_phi = x__R_0(2)] ) );
```

$$x_{R\_1} := \begin{bmatrix} 3.93556604183892\ 10^{-11} \\ 22.7813780076877 \end{bmatrix} \tag{3.4.2}$$

```
> x__R_2 := evalf( x__R_1- eval( J__inv_num . f__num ,
  [I__S_theta = x__R_1(1), U__OC_theta_phi = x__R_1(2)] ) );
```

$$x_{R\_2} := \begin{bmatrix} 3.93556604183619\ 10^{-11} \\ 22.7813780046823 \end{bmatrix} \tag{3.4.3}$$