Course Mechatronics System Modelling

Modelling a Solar Panel - Homework 1

Stefano Tonini 248413

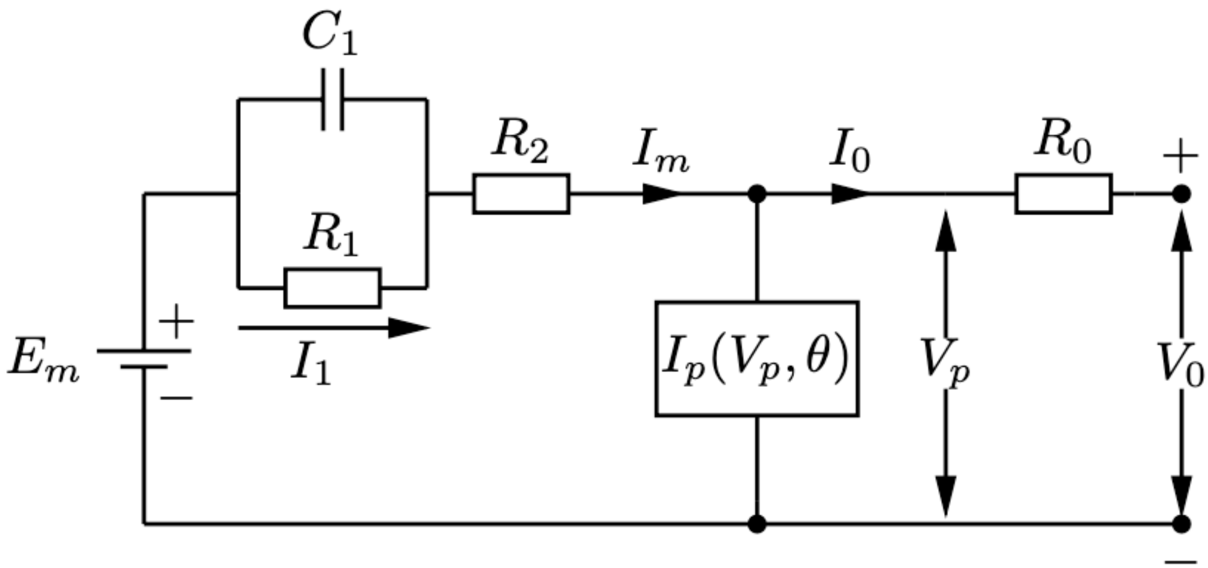
Initialization

```
> restart:
  with(plots):
  with(LinearAlgebra):
  with(GraphTheory):

  interface(rtablesize=20):
  plots[setcolors]("spring"):
  c_set := ColorTools:-GetPalette("spring");

  FONT_TYPE     := "Helvetica":
  FONT_SIZE_LAB :=  14:
  FONT_SIZE_TIT :=  18:
  plots[setoptions](axes            = boxed,
                    size            = [800,200],
                    axis[2]         = [gridlines = [linestyle=dot]
  ],
                    legendstyle     = [location="top",font=
  [FONT_TYPE,FONT_SIZE_LAB]],
                    labelfont       = [FONT_TYPE,FONT_SIZE_LAB],
                    axesfont        = [FONT_TYPE,FONT_SIZE_LAB],
                    titlefont       = [FONT_TYPE,bold,
  FONT_SIZE_TIT],
                    captionfont     = [FONT_TYPE,italic,
  FONT_SIZE_TIT],
                    labeldirections = [horizontal , vertical]
  );
```

$c\_set :=$ ⟨Palette Spring: Blue Rose YellowGreen BlueGreen Violet Cobalt Yellow PurpleRed GreenBlue PaleGreen Orange Purple Green SeaBlue PaleYellow PaleBlueGreen⟩ **(1.1)**

Battery Model

Ceraolo Battery Model



Parameters and variables:
Qe extracted (or intake) charges.

- I0 extracted current.
- I1 current on the overvoltage branch.
- V0 battery pins voltage.
- θ battery temperature.
- θa ambient temperature.

```
> # Define independent functions and variables
```

```
> MAXi0 := proc(Ic)
  max(0,Ic);
  end proc:
```

```
> # Capacity
  Capacity := proc(I_curr,theta) :
   local I__capacity := MAXi0(I_curr);
   (Cn/(Ac+(1-Ac)*(I__capacity/In)^delta))*((theta-theta__f)/
  (theta__n-theta__f))^epsilon;
  end proc:
```

```
> # Parameters

  params := [
  SOC = 1 - Q__e/Capacity(0,theta),
  DOC = 1 - Q__e/Capacity(I__1,theta),
  Em = Em0-KE*(273+theta)*(1-SOC),
```

```
R0 = R00*(1 + A0*(1 - SOC)),
R1 = -R10*log(DOC),
R2 = R20*( exp(A21(1 - SOC))/( 1 + exp (A22*I__m/In)))]
```

$$params := \left[SOC = 1 - \frac{Q_e\, Ac}{Cn \left(\frac{\theta - \theta_f}{\theta_n - \theta_f}\right)^{\epsilon}}, DOC = 1 - \frac{Q_e \left(Ac + (1 - Ac) \left(\frac{\max(0, I_1)}{In}\right)^{\delta}\right)}{Cn \left(\frac{\theta - \theta_f}{\theta_n - \theta_f}\right)^{\epsilon}}, Em = Em0 - KE\,(273 + \theta)\,(1 - SOC), R0 = R00\,(1 + A0\,(1 - SOC)), R1 = -R10 \ln(DOC), R2 = \frac{R20\, \mathrm{e}^{A21\,(1 - SOC)}}{1 + \mathrm{e}^{\frac{A22\, I_m}{In}}}\right] \tag{2.1.1}$$

```
> # Insert data optimized using DE
data := [R00 = 0.0920607, R10 = 9.71807, R20 =  0.0506281,
C1 = 1.0,
Em0 = 12.623, KE = 0.623657,
A0 = 0.347071,
A21 = 3.62106,
A22 = 7.10553,
Cn = 4.54474,
Ac = 0.0077,
delta =  0.64741,
In = 1.90781,
#C(0,θ) = 570,
#C(max(0, I__1), θ) = 570,
tau1 = 44499,
theta = 25,
theta__n = 25,
theta__f = 24,
epsilon = 1

]
```

$$data := [R00 = 0.0920607, R10 = 9.71807, R20 = 0.0506281, C1 = 1.0, Em0 = 12.623, KE = 0.623657, A0 = 0.347071, A21 = 3.62106, A22 = 7.10553, Cn = 4.54474, Ac = 0.0077, \delta = 0.64741, In = 1.90781, \tau 1 = 44499, \theta = 25, \theta_n = 25, \theta_f = 24, \epsilon = 1] \tag{2.1.2}$$

```
> subs(data,params)
```

$$\Bigg[SOC = 1 - 0.001694266338\, Q_e, DOC = 1 - 0.2200345894\, Q_e \left(0.0077 + 0.6531632611 \max\left(0, I_1\right)^{0.64741}\right), Em = -173.226786 + 185.849786\, SOC, R0 = 0.1240122992 - 0.03195159921\, SOC, R1 = -9.71807 \ln(DOC), R2 = \frac{0.0506281\, \mathrm{e}^{3.62106(1 - SOC)}}{1 + \mathrm{e}^{\frac{3.724443210\, I}{m}}}\Bigg] \quad (2.1.3)$$

Define the differential equations using the simplified model in the paper

```
> # Model simplified Equation
  eqns := [Q__e = I__0*t,
           I__1 =  I__0*(1-exp(-t/tau1)),
           V0 = Em - (R1*(1 - exp(-t/tau1)) + R0 + R2)*I__0
           ]
  :
  <%>
```

$$\begin{bmatrix} Q_e = I_0\, t \\ I_1 = I_0 \left(1 - \mathrm{e}^{-\frac{t}{\tau 1}}\right) \\ V0 = Em - \left(R1 \left(1 - \mathrm{e}^{-\frac{t}{\tau 1}}\right) + R0 + R2\right) I_0 \end{bmatrix} \quad (2.1.4)$$

```
> # Semplification done in the paper
  I__m := subs(data,I__0);
```

$$I_m := I_0 \quad (2.1.5)$$

```
> sol1 := solve(subs(params,data,eqns),[Q__e,I__1,V0])
```

$$sol1 := \Bigg[\Bigg[Q_e = I_0\, t, I_1 = -1.\, I_0\, \mathrm{e}^{-0.00002247241511\, t} + I_0, V0 = -\frac{1}{1. + \mathrm{e}^{3.724443210\, I_0}} \Big(1.000000000 \times 10^{-11} \Big(9.718070000 \times 10^{11}\, I_0\, \mathrm{e}^{-0.00002247241511\, t} \ln(DOC)\, \mathrm{e}^{3.724443210\, I_0} + 9.718070000 \times 10^{11}\, I_0\, \mathrm{e}^{-0.00002247241511\, t} \ln(DOC) - 9.718070000 \times 10^{11}\, I_0 \ln(DOC)\, \mathrm{e}^{3.724443210\, I_0} \quad (2.1.6)$$

$$- 3.195159921 \times 10^{9} I_0 e^{3.724443210 I_0} SOC - 9.718070000 \times 10^{11} I_0 \ln(DOC)$$
$$+ 5.062810000 \times 10^{9} I_0 e^{3.62106(1 - SOC)} + 1.240122992 \times 10^{10} I_0 e^{3.724443210 I_0}$$
$$- 3.195159921 \times 10^{9} I_0 SOC - 1.858497860 \times 10^{13} SOC e^{3.724443210 I_0}$$
$$+ 1.240122992 \times 10^{10} I_0 + 1.732267860 \times 10^{13} e^{3.724443210 I_0} - 1.858497860$$
$$\times 10^{13} SOC + 1.732267860 \times 10^{13} \Big) \Big) \Big] \Big]$$

```
> eqns__battery := subs(sol1[1,1],sol1[1,2],data,params)
```

$$eqns_{battery} := \Big[SOC = 1 - 0.001694266338 I_0 t, DOC = 1 - 0.2200345894 I_0 t \Big(0.0077 + 0.6531632611 \max\big(0, -1. I_0 e^{-0.00002247241511 t} + I_0\big)^{0.64741}\Big), Em = -173.226786 + 185.849786 SOC, R0 = 0.1240122992 - 0.03195159921 SOC, R1 = -9.71807 \ln(DOC), R2 = \frac{0.0506281 e^{3.62106(1 - SOC)}}{1 + e^{3.724443210 I_0}} \Big] \quad (2.1.7)$$

```
> simplify(subs(%,sol1[1,3]))
```

$$V0 = \frac{1}{1000.000000 + 1000.000000 e^{3.72444321 I_0}} \Big(12623. + I_0 \Big(-9718.07 e^{-0.00002247241511 t + 3.72444321 I_0} - 9718.07 e^{-0.00002247241511 t} + 9718.07 e^{3.72444321 I_0} + 9718.07\Big) \ln\Big(1 - 0.001694266338 I_0 t - 0.1437185100 I_0 t \max\big(0, I_0 (1. - 1. e^{-0.00002247241511 t})\big)^{0.64741}\Big) + \big(12623. - 0.05413451899 I_0^2 t + (-92.06069999 - 314.8790363 t) I_0\big) e^{3.72444321 I_0} - 0.05413451899 I_0^2 t + (-1984.39563 - 314.8790363 t) I_0\Big) \quad (2.1.8)$$

Equation of the Voltage of the Battery (discharge)

```
> V0plot := simplify(evalf(op(2,%)))
```

$$V0plot := \frac{1}{1000.000000 + 1000.000000 e^{3.72444321 I_0}} \Big(12623. + I_0 \Big(-9718.07 e^{-0.00002247241511 t + 3.72444321 I_0} - 9718.07 e^{-0.00002247241511 t} \quad (2.1.9)$$

$$+ 9718.07\, e^{3.72444321\, I_0} + 9718.07\Big) \ln\Big(1. - 0.001694266338\, I_0\, t$$

$$- 0.1437185100\, I_0\, t \max\Big(0., I_0\, \big(1. - 1.\, e^{-0.00002247241511\, t}\big)\Big)^{0.64741}\Big) + \Big(12623.$$

$$- 0.05413451899\, I_0^2\, t + (-92.06069999 - 314.8790363\, t)\, I_0\Big)\, e^{3.72444321\, I_0}$$

$$- 0.05413451899\, I_0^2\, t + (-1984.39563 - 314.8790363\, t)\, I_0\Big)$$

Define V0 as a function of time and current

```
> V0__func := simplify(unapply(V0plot, t, I__0));
```

$$V0_{func} := (t, I_0) \mapsto \frac{1}{1000.000000 + 1000.000000 \cdot e^{3.72444321 \cdot I_0}} \Big(12623. + I_0 \cdot \Big(\tag{2.1.10}$$

$$-9718.07 \cdot e^{-0.00002247241511 \cdot t + 3.72444321 \cdot I_0} - 9718.07 \cdot e^{-0.00002247241511 \cdot t} + 9718.07$$

$$\cdot e^{3.72444321 \cdot I_0} + 9718.07\Big) \cdot \ln\Big(1. - 0.001694266338 \cdot I_0 \cdot t - 0.1437185100 \cdot I_0 \cdot t$$

$$\cdot \max\Big(0., I_0 \cdot \big(1. - 1. \cdot e^{-0.00002247241511 \cdot t}\big)\Big)^{0.64741}\Big) + \Big(12623. - 0.05413451899 \cdot I_0^2 \cdot t$$

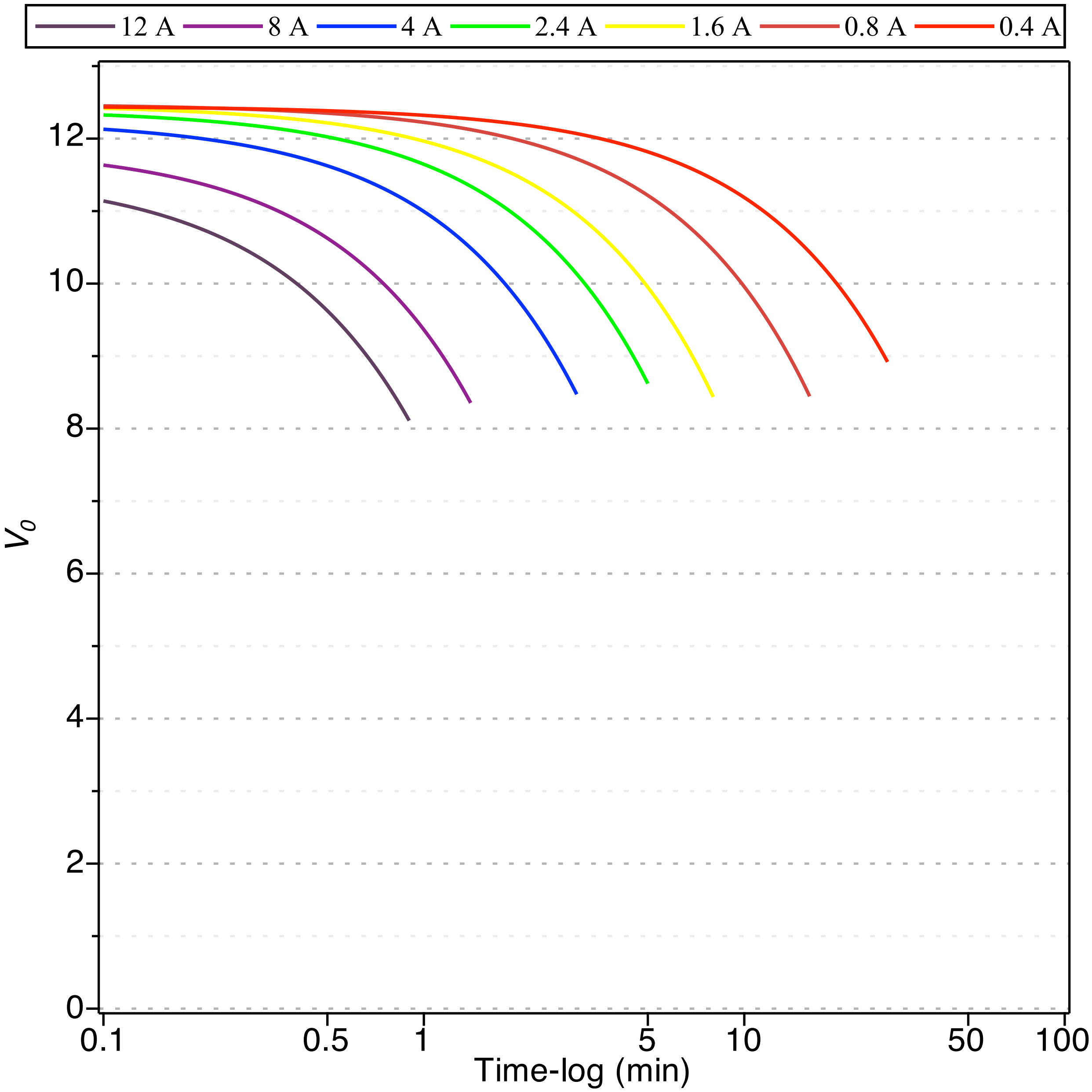
$$+ (-92.06069999 - 314.8790363 \cdot t) \cdot I_0\Big) \cdot e^{3.72444321 \cdot I_0} - 0.05413451899 \cdot I_0^2 \cdot t$$

$$+ (-1984.39563 - 314.8790363 \cdot t) \cdot I_0\Big)$$

Plot the dischrage of the battery like in the paper "Hybrid vehicle optimization: lead acid battery modellization"

```
> display(semilogplot(V0__func(t,12), t=10^(-1)..0.9,color=
  violet, legend = "12 A"),

          semilogplot(V0__func(t,8), t=10^(-1)..1.4,color =
  purple, legend = "8 A"),
          semilogplot(V0__func(t,4), t=10^(-1)..3,color = blue,
  legend = "4 A"),
  semilogplot(V0__func(t,2.4), t=10^(-1)..5,color = green,
  legend = "2.4 A"),
  semilogplot(V0__func(t,1.6), t=10^(-1)..8,color = yellow,
  legend = "1.6 A"),
          semilogplot(V0__func(t,0.8), t=10^(-1)..16,color =
  orange, legend = "0.8 A"),
  semilogplot(V0__func(t,0.4), t=10^(-1)..28,color = red, legend
  = "0.4 A"),
          labels = ["Time-log (min)", V__0],
          view = [10^(-1)..100, 0..13],
          size = [1000, 500]
```

```
    );
```



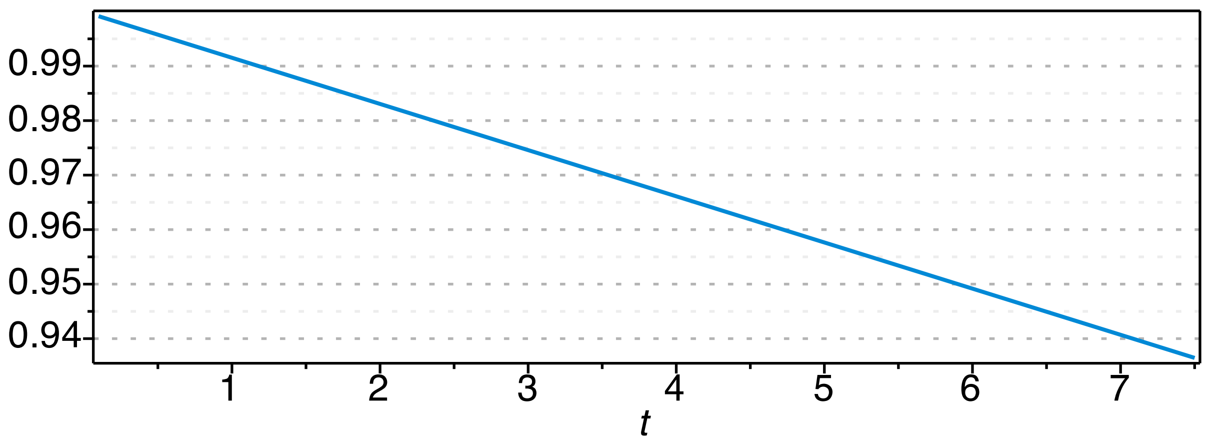
The discharghe plot obtained are comparable with the curve shown in the paper: "Hybrid vehicle optimization: lead acid battery modellization"

Define the equation of the State of Charge of the battery

```
> SOC1 := unapply(rhs(subs(data,params,eqns__battery[1])), [t,
  I__0]);
```

$$SOC1 := (t, I_0) \mapsto 1 - 0.001694266338 \cdot I_0 \cdot t \quad \textbf{(2.1.11)}$$

```
> display(plot(SOC1(t,5),t=10^(-1)..7.5));
```



Charging the battery

Define the parameters in order to simulate the charge of the battery

```
> # Parameters

  params2 := [
  SOCc = 0.94 + Q__e/Capacity(0,theta),  # The value 0.94 of SOC it used to define the battery completly discharged
  DOCc = 0.94 + Q__e/Capacity(I__1,theta),
  E__mc = Em0-KE*(273+theta)*(1-SOCc),

  R__0c = R00*(1 + A0*(1 - SOCc)),
  R__1c = -R10*log(DOCc),
  R__2c = R20*( exp(A21(1 - SOCc))/( 1 + exp (A22*I__m/In)))
  ];
```

$$params2 := \left[SOCc = 0.94 + \frac{Q_e\, Ac}{Cn \left(\frac{\theta - \theta_f}{\theta_n - \theta_f} \right)^{\epsilon}}, DOCc = 0.94 + \frac{Q_e \left(Ac + (1 - Ac) \left(\frac{\max(0, I_1)}{In} \right)^{\delta} \right)}{Cn \left(\frac{\theta - \theta_f}{\theta_n - \theta_f} \right)^{\epsilon}}, E_{mc} = Em0 - KE\,(273 + \theta)\,(1 - SOCc), R_{0c} = R00\,(1 + A0\,(1 - SOCc)), R_{1c} = -R10 \ln(DOCc), R_{2c} \right. \tag{2.1.1.1}$$

$$= \left[\frac{R20\, e^{A21(1-SOCc)}}{1+e^{\frac{A22\, I_0}{In}}} \right]$$

```
> subs(data,params2)
```

$$\left[SOCc = 0.94 + 0.001694266338\, Q_e,\ DOCc = 0.94 + 0.2200345894\, Q_e \left(0.0077 \right.\right. \quad (2.1.1.2)$$

$$\left. + 0.6531632611 \max\left(0, I_1\right)^{0.64741}\right),\ E_{mc} = -173.226786$$

$$+ 185.849786\, SOCc,\ R_{0c} = 0.1240122992 - 0.03195159921\, SOCc,\ R_{1c} =$$

$$\left. -9.71807 \ln(DOCc),\ R_{2c} = \frac{0.0506281\, e^{3.62106(1-SOCc)}}{1+e^{3.724443210\, I_0}} \right]$$

Define the differential equations using the given simplified model

```
> # Model simplified Equation
  eqns__2 := [Q__e = I__0*t,
           I__1 =  I__0*(1-exp(-t/tau1)),
           V0 = E__mc - (R__1c*(1 - exp(-t/tau1)) + R__0c + 
  R__2c)*I__0
           ]
  :
  <%>
```

$$\left[\begin{array}{c} Q_e = I_0\, t \\ I_1 = I_0 \left(1 - e^{-\frac{t}{\tau 1}}\right) \\ V0 = E_{mc} - \left(R_{1c}\left(1 - e^{-\frac{t}{\tau 1}}\right) + R_{0c} + R_{2c}\right) I_0 \end{array} \right] \quad (2.1.1.3)$$

```
> solc := solve(subs(params2,data,eqns__2),[Q__e,I__1,V0])
```

$$solc := \left[\left[Q_e = I_0\, t,\ I_1 = -1.\, I_0\, e^{-0.00002247241511\, t} + I_0,\ V0 = \right.\right. \quad (2.1.1.4)$$

$$-\frac{1}{1. + e^{3.724443210\, I_0}} \left(1.000000000 \times 10^{-11} \left(9.718070000\right.\right.$$

$$\times 10^{11}\, I_0\, e^{-0.00002247241511\, t} \ln(DOCc)\, e^{3.724443210\, I_0} + 9.718070000$$

$$\times 10^{11}\, I_0\, e^{-0.00002247241511\, t} \ln(DOCc) - 9.718070000$$

$$\times 10^{11} I_0 \ln(DOCc)\, e^{3.724443210 I_0} - 3.195159921 \times 10^{9} I_0\, e^{3.724443210 I_0} SOCc$$

$$- 9.718070000 \times 10^{11} I_0 \ln(DOCc) + 5.062810000 \times 10^{9} I_0\, e^{3.62106(1 - SOCc)}$$

$$+ 1.240122992 \times 10^{10} I_0\, e^{3.724443210 I_0} - 3.195159921 \times 10^{9} I_0\, SOCc$$

$$- 1.858497860 \times 10^{13} SOCc\, e^{3.724443210 I_0} + 1.240122992 \times 10^{10} I_0$$

$$+ 1.732267860 \times 10^{13} e^{3.724443210 I_0} - 1.858497860 \times 10^{13} SOCc$$

$$+ 1.732267860 \times 10^{13} \Big) \Big) \Big] \Big]$$

```
> eqns__battery2 := subs(solc[1,1],solc[1,2],data,params2)
```

$$eqns_{battery2} := \Big[SOCc = 0.94 + 0.001694266338\, I_0\, t,\ DOCc = 0.94 \tag{2.1.1.5}$$

$$+ 0.2200345894\, I_0\, t \Big(0.0077 + 0.6531632611 \max\Big(0,$$

$$-1.\, I_0\, e^{-0.00002247241511\, t} + I_0\Big)^{0.64741}\Big),\ E_{mc} = -173.226786$$

$$+ 185.849786\, SOCc,\ R_{0c} = 0.1240122992 - 0.03195159921\, SOCc,\ R_{1c} =$$

$$-9.71807 \ln(DOCc),\ R_{2c} = \frac{0.0506281\, e^{3.62106(1 - SOCc)}}{1 + e^{3.724443210 I_0}} \Big]$$

```
> simplify(subs(eqns__battery2,solc[1,3]))
```

$$V0 = \frac{1}{1. + e^{3.72444321 I_0}} \Big(1.4720128 + I_0 \Big(-9.71807\, e^{-0.00002247241511\, t + 3.72444321 I_0} \tag{2.1.1.6}$$

$$- 9.71807\, e^{-0.00002247241511\, t} + 9.71807\, e^{3.72444321 I_0} + 9.71807\Big) \ln\Big(0.94$$

$$+ 0.001694266338\, I_0\, t + 0.1437185100\, I_0\, t \max\Big(0, I_0\, (1.$$

$$- 1.\, e^{-0.00002247241511\, t})\Big)^{0.64741}\Big) + \Big(1.4720128 + 0.00005413451899\, I_0^2\, t + ($$

$$-0.09397779594 + 0.3148790363\, t)\, I_0\Big)\, e^{3.72444321 I_0} + 0.00005413451899\, I_0^2\, t$$

$$+ (-1.986312726 + 0.3148790363\, t)\, I_0\Big)$$

Equation of the Voltage of the Battery (charge)

```
> V0plotc := simplify(evalf(op(2,%)))
```

(2.1.1.7)

$$V0plotc := \frac{1}{1. + e^{3.72444321\, I_0}} \Big(1.4720128 + I_0 \Big(-9.71807\, e^{-0.00002247241511\, t + 3.72444321\, I_0} - 9.71807\, e^{-0.00002247241511\, t} + 9.71807\, e^{3.72444321\, I_0} + 9.71807\Big) \ln\Big(0.94 + 0.001694266338\, I_0\, t + 0.1437185100\, I_0\, t \max\big(0., I_0\, (1. - 1.\, e^{-0.00002247241511\, t})\big)^{0.64741}\Big) + \Big(1.4720128 + 0.00005413451899\, I_0^2\, t + (-0.09397779594 + 0.3148790363\, t)\, I_0\Big)\, e^{3.72444321\, I_0} + 0.00005413451899\, I_0^2\, t + (-1.986312726 + 0.3148790363\, t)\, I_0\Big) \tag{2.1.1.7}$$

Define V0 as a function of time and current

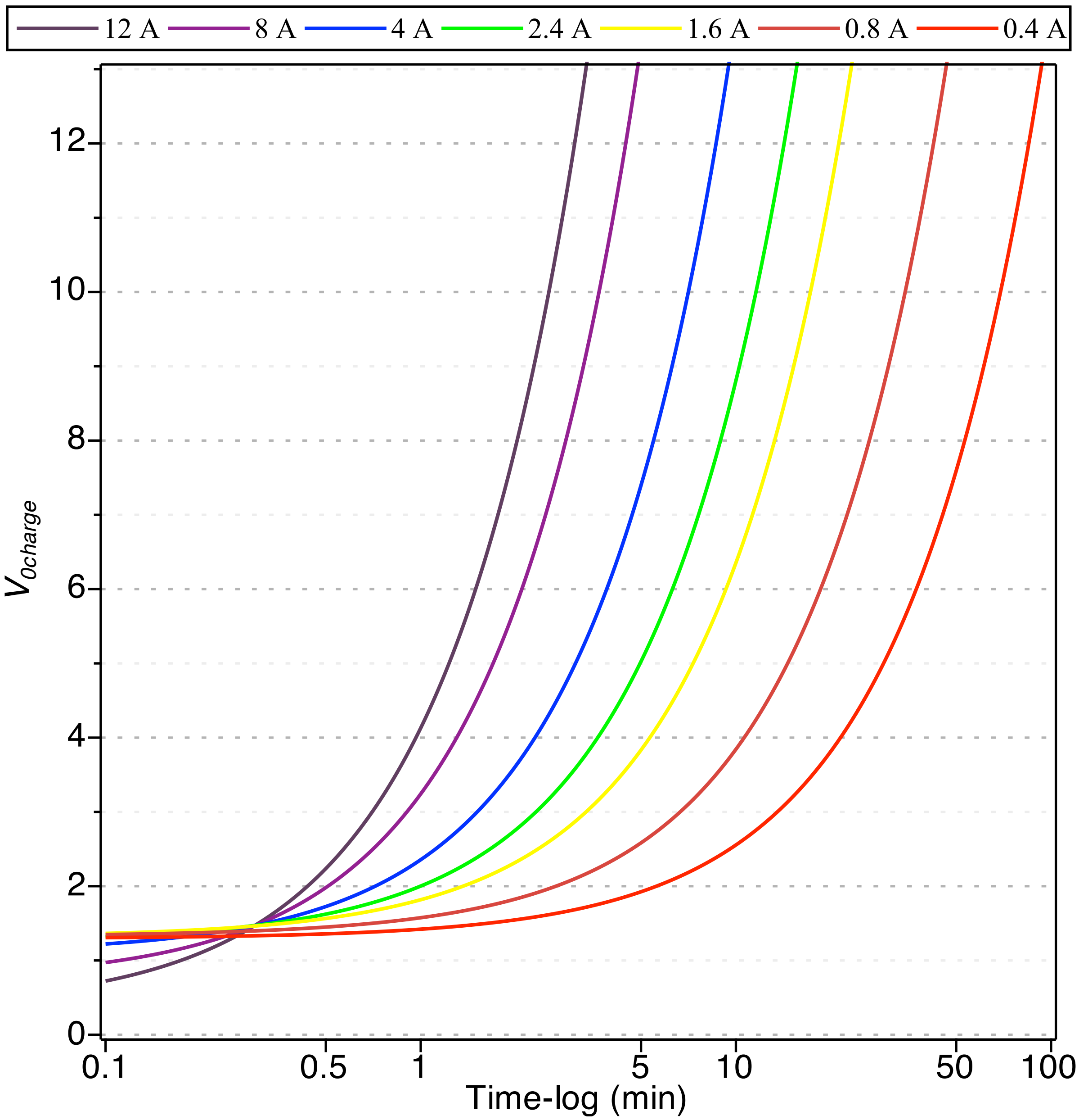
```
> V0__funcc := simplify(unapply(V0plotc, t, I__0));
```

$$V0_{funcc} := (t, I_0) \mapsto \frac{1}{1. + e^{3.72444321\cdot I_0}} \Big(1.4720128 + I_0\cdot \Big(-9.71807\cdot e^{-0.00002247241511\cdot t + 3.72444321\cdot I_0} - 9.71807\cdot e^{-0.00002247241511\cdot t} + 9.71807\cdot e^{3.72444321\cdot I_0} + 9.71807\Big)\cdot \ln\Big(0.94 + 0.001694266338\cdot I_0\cdot t + 0.1437185100\cdot I_0\cdot t\cdot \max\big(0., I_0\cdot (1. - 1.\cdot e^{-0.00002247241511\cdot t})\big)^{0.64741}\Big) + \Big(1.4720128 + 0.00005413451899\cdot I_0^2\cdot t + (-0.09397779594 + 0.3148790363\cdot t)\cdot I_0\Big)\cdot e^{3.72444321\cdot I_0} + 0.00005413451899\cdot I_0^2\cdot t + (-1.986312726 + 0.3148790363\cdot t)\cdot I_0\Big) \tag{2.1.1.8}$$

Plot the charge of the battery with different values of current I0

```
> display(semilogplot(V0__funcc(t,12), t=10^(-1)..3.5,color=
  violet, legend = "12 A"),

          semilogplot(V0__funcc(t,8), t=10^(-1)..5.2,color =
  purple, legend = "8 A"),
          semilogplot(V0__funcc(t,4), t=10^(-1)..10,color =
  blue, legend = "4 A"),
  semilogplot(V0__funcc(t,2.4), t=10^(-1)..17,color = green,
  legend = "2.4 A"),
  semilogplot(V0__funcc(t,1.6), t=10^(-1)..25,color = yellow,
  legend = "1.6 A"),
          semilogplot(V0__funcc(t,0.8), t=10^(-1)..50,color =
  orange, legend = "0.8 A"),
```

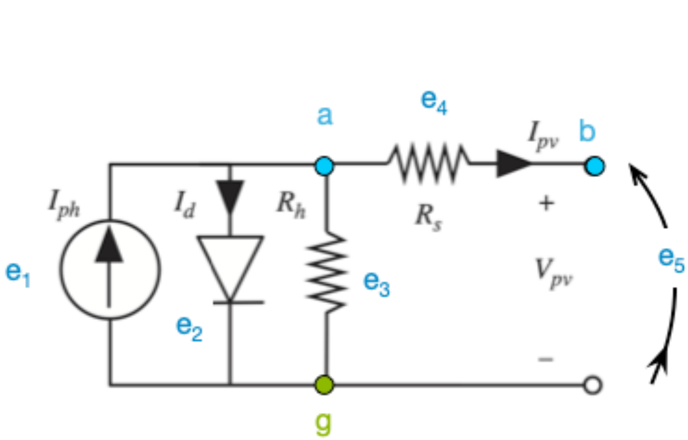
```
semilogplot(V0__funcc(t,0.4), t=10^(-1)..100,color = red,
legend = "0.4 A"),
        labels = ["Time-log (min)", V__0charge],
        view = [10^(-1)..100, 0..13],
        size = [1000, 500]
      );
```



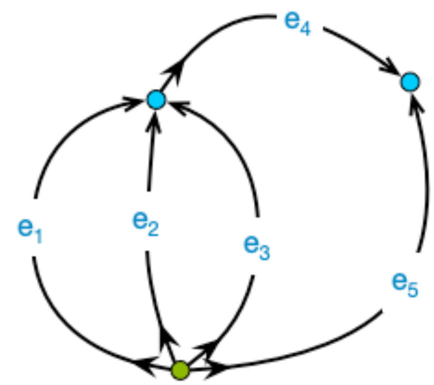
PV Panel Model

Solar Cell equivalent model: single diode

Single Diode Model



Linear Graph - DDM



The diode equation writes :

$$I_d = I_s \left(e^{\frac{V_d}{\eta V_{th}}} - 1 \right)$$

where $V_{th} = \frac{k_B T}{q}$ is the thermal voltage.

```
> eq_Vth := V__th = k__B*T/q
```

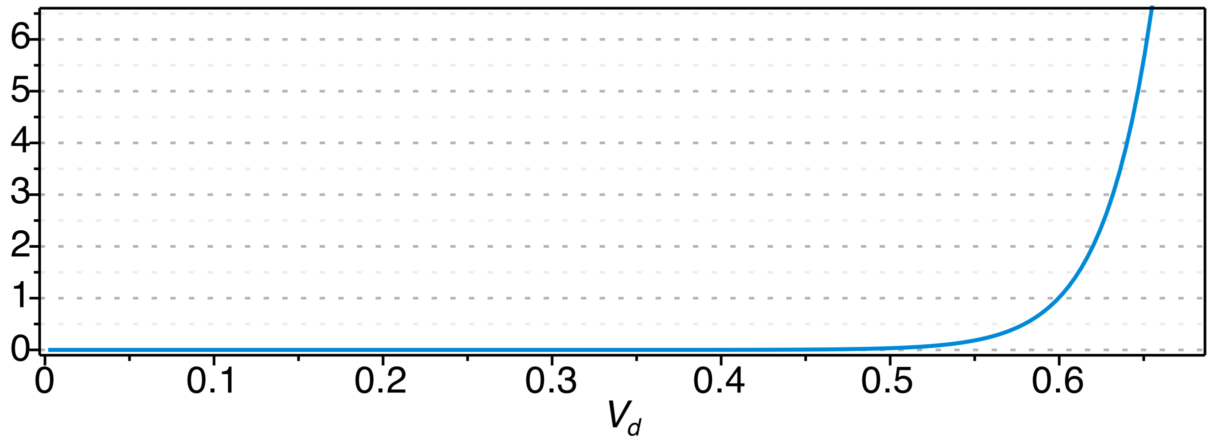
$$eq\_Vth := V_{th} = \frac{k_B T}{q} \qquad (3.1.1)$$

```
> data := k__B=1.38062*10^(-23), T =300, q =1.60219*10^(-19),
  I__ds = 1.2172e-9, Rs = 9.2/1000, Rh =12.7, eta = 1.13:
  subs(%,eq_Vth );

  data_full := %, %%;

  I__d = I__ds*(exp(V__d/(eta*V__th)) - 1);

  plot(subs(%%,data_full,eta=1,rhs(%)), V__d = 0..0.75);
```

$$V_{th} = 0.02585124111$$

$$data\_full := V_{th} = 0.02585124111, k_B = 1.380620000 \times 10^{-23}, T = 300, q = 1.602190000 \times 10^{-19}, I_{ds} = 1.2172 \times 10^{-9}, Rs = 0.009200000000, Rh = 12.7, \eta = 1.13$$

$$I_d = I_{ds} \left(e^{\frac{V_d}{\eta V_{th}}} - 1 \right)$$



Linear Graph of the electrical system

List of **vertices** (or nodes)

```
> V := [a,b,g];
```

$$V := [a, b, g] \tag{3.2.1}$$

The list of **edges** (correspond to physical elements) and we must also specify the .
The order is defined according to the arrow in the second picture.

```
> E := [[a,g], # element Eph
        [g,a], # element D1: diode 1
        [g,a], # element Rh: shunt resistance
        [a,b], # element Rs: series resistance
        [g,b]  # element V: output
        ];
```

$$E := [[a, g], [g, a], [g, a], [a, b], [g, b]] \tag{3.2.2}$$

Through and across variables

Through variables (current)

```
> tau_vars   := [i__ph(t),i__d(t), i__Rh(t),i__Rs(t),i__pv(t)];
```

$$tau\_vars := [i_{ph}(t), i_d(t), i_{Rh}(t), i_{Rs}(t), i_{pv}(t)] \tag{3.2.3}$$

Across variables (voltages)

```
> alpha_vars := [v__ph(t),v__d(t), v__Rh(t),v__Rs(t),v__pv(t)];
```

$$alpha\_vars := [v_{ph}(t), v_d(t), v_{Rh}(t), v_{Rs}(t), v_{pv}(t)] \tag{3.2.4}$$

Create a procedure to get the incidence matrix providing the list of nodes and the list of edges

```
> build_incidence_matrix := proc(V::list,E::list(list),$)
```

```
    local im, i, j, v, e;

    # create empty incidence matrix
    im := Matrix(1..nops(V),1..nops(E),fill=0):

    for j from 1 to nops(V) do
      v := V[j];  # j-th node

      # loop over the edges and check if the node is in
      for i from 1 to nops(E) do
        #print(type(E[i],list);
        #if nops(E[i]) > 1 then
        #  e := E[i][1]; # i-th edge
        #else
          e := E[i]; # i-th edge
        #end ;
        #print(e,v=e[1]);
        if has(e,v) and (v =  e[1] ) then  # check v is in e and
  v is first element in e
          im[j,i] := +1;
        elif  has(e,v) and (v = e[2] ) then # check v is in e
  and v is second element in e
          im[j,i] := -1;
        else
          im[j,i] := 0; #  v is not in e
        end;
      end do:
    end do:

    return im;

  end proc:
```

Incidence matrix

```
> IM0  := build_incidence_matrix(V,E);
  DF := DataFrame(IM0, columns = [seq(e__||i,i=1..nops(E))],
                       rows    = convert(V,list),
                       datatypes=[seq(integer,i=1..nops(E))]);
```

$$IM0 := \begin{bmatrix} 1 & -1 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 \\ -1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

(3.2.5)

$$DF := \begin{array}{c} \\ a \\ b \\ g \end{array} \begin{bmatrix} e_1 & e_2 & e_3 & e_4 & e_5 \\ 1 & -1 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 \\ -1 & 1 & 1 & 0 & 1 \end{bmatrix} \tag{3.2.5}$$

As expected is singular

```
> GaussianElimination(IM0)
```

$$\begin{bmatrix} 1 & -1 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.2.6}$$

I remove the ground node and get a linearly independent set of rows

```
> A_mat := IM0[1..-2,1..-1];
```

$$A\_mat := \begin{bmatrix} 1 & -1 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 \end{bmatrix} \tag{3.2.7}$$

Compute the number of nodes, vertices, brances and chords

```
> nv := nops(V);
  nb := nv-1;
  ne := nops(E);
  nc := ne-nb;
```

$$nv := 3$$
$$nb := 2$$
$$ne := 5$$
$$nc := 3 \tag{3.2.8}$$

Reduce in the echelon form and extract Ib and Ac matrices

```
> Af_mat := ReducedRowEchelonForm(A_mat);
  Ib_mat := Af_mat[1..nb,1..nb];
  Ac_mat := Af_mat[1..nb,nb+1..-1];

  nc := ne-nb;
```

$$Af\_mat := \begin{bmatrix} 1 & -1 & -1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

$$Ib\_mat := \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix}$$

$$Ac\_mat := \begin{bmatrix} -1 & 0 & -1 \\ 0 & 1 & 1 \end{bmatrix}$$

$$nc := 3 \tag{3.2.9}$$

Matrix Bf is the **circuit matrix** that contains the set of fundamental circuits or loops (i.e. independent circuits)

```
> NullSpace(Af_mat); # find the orthogonal
  tmp_Bf := Transpose(Matrix([op(%)]));
```

$$\left\{ \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \\ 1 \end{bmatrix} \right\}$$

$$tmp\_Bf := \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 1 \end{bmatrix} \tag{3.2.10}$$

Proof that they are orthogonal
It represents a general conservation principle, applicable to any type of physical system represented by a linear graph.
This Principle of Orthogonality can be used to derive Tellegen's Theorem or the Principle of Virtual Work for mechanical systems.

```
> Af_mat.Transpose(tmp_Bf);
```

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.2.11}$$

We can get the matrix also
Af_mat = <Ib_mat | Ac_mat>;
Bf_mat = <Bb_mat | Ic_mat>;
Af_mat.Bf_mat^T = 0 --> <Ib_mat | Ac_mat><Bb_mat | Ic_mat>^T = 0 -->Ib_mat.Bb_mat^T = -Ac_mat.Ic_mat^T --> Bb_mat^T = -Ac_mat --> Bb_mat = -Ac_mat^T

```
> Ic_mat :=  IdentityMatrix(nc):
  Bb_mat := -Transpose(Ac_mat):

  Bf_mat := <Bb_mat|Ic_mat>:

  Ic_mat, Bb_mat,Bf_mat ;

  #check that it is correct
  Bf_mat = tmp_Bf;
  Bf_mat := tmp_Bf;
```

$$\left[\begin{array}{ccc}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{array}\right], \left[\begin{array}{cc}1 & 0\\0 & -1\\1 & -1\end{array}\right], \left[\begin{array}{ccccc}1 & 0 & 1 & 0 & 0\\0 & -1 & 0 & 1 & 0\\1 & -1 & 0 & 0 & 1\end{array}\right]$$

$$\left[\begin{array}{ccccc}1 & 0 & 1 & 0 & 0\\0 & -1 & 0 & 1 & 0\\1 & -1 & 0 & 0 & 1\end{array}\right] = \left[\begin{array}{ccccc}1 & 0 & 1 & 0 & 0\\1 & 1 & 0 & 0 & 0\\1 & 0 & 0 & -1 & 1\end{array}\right]$$

$$Bf\_mat := \left[\begin{array}{ccccc}1 & 0 & 1 & 0 & 0\\1 & 1 & 0 & 0 & 0\\1 & 0 & 0 & -1 & 1\end{array}\right] \quad \textbf{(3.2.12)}$$

Cutset or node equations

Obtain the cutset equations

```
> cutset_eqns  := Af_mat.<tau_vars>;

  #latex(cutset_eqns)
```

$$cutset\_eqns := \left[\begin{array}{c}i_{ph}(t) - i_{d}(t) - i_{Rh}(t) - i_{pv}(t)\\i_{Rs}(t) + i_{pv}(t)\end{array}\right] \quad \textbf{(3.2.13)}$$

We can also exploit the echelon form to solve the branch through variables as function of chords through variables

```
> Ib_mat.<tau_vars[1..nb]> = -Ac_mat.<tau_vars[nb+1..-1]>;

  #sol_tau_b := op(solve(cutset_eqns,tau_vars[1..nb])): # verificare
  sol_tau_b := op(solve(cutset_eqns,[i__ph(t),i__Rs(t)])):
  <sol_tau_b>
```

$$\left[\begin{array}{c}i_{ph}(t) - i_{d}(t)\\0\end{array}\right] = \left[\begin{array}{c}i_{Rh}(t) + i_{pv}(t)\\-i_{Rs}(t) - i_{pv}(t)\end{array}\right]$$

$$\left[\begin{array}{c}i_{ph}(t) = i_{d}(t) + i_{Rh}(t) + i_{pv}(t)\\i_{Rs}(t) = -i_{pv}(t)\end{array}\right] \quad \textbf{(3.2.14)}$$

```
> tau_vars[1..nb];
```

$$\left[i_{ph}(t), i_{d}(t)\right] \quad \textbf{(3.2.15)}$$

```
> Ib_mat;
```

(3.2.16)

$$\begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix} \tag{3.2.16}$$

```
> <tau_vars[1..nb]>;
```

$$\begin{bmatrix} i_{ph}(t) \\ i_{d}(t) \end{bmatrix} \tag{3.2.17}$$

```
> Ib_mat.<tau_vars[1..nb]> = -Ac_mat.<tau_vars[nb+1..-1]>;
```

$$\begin{bmatrix} i_{ph}(t) - i_{d}(t) \\ 0 \end{bmatrix} = \begin{bmatrix} i_{Rh}(t) + i_{pv}(t) \\ -i_{Rs}(t) - i_{pv}(t) \end{bmatrix} \tag{3.2.18}$$

Circuit equations

Circuit or loop equations

```
> circuit_eqns := Bf_mat.<alpha_vars>;

  #latex(circuit_eqns)
```

$$circuit\_eqns := \begin{bmatrix} v_{ph}(t) + v_{Rh}(t) \\ v_{ph}(t) + v_{d}(t) \\ v_{ph}(t) - v_{Rs}(t) + v_{pv}(t) \end{bmatrix} \tag{3.2.19}$$

We can also solve the chords across variables as function of the branch across variables

```
> sol_alpha_c := op(solve(circuit_eqns,alpha_vars[1..nc])):
  <sol_alpha_c>;
```

$$\begin{bmatrix} v_{ph}(t) = v_{Rs}(t) - v_{pv}(t) \\ v_{d}(t) = -v_{Rs}(t) + v_{pv}(t) \\ v_{Rh}(t) = -v_{Rs}(t) + v_{pv}(t) \end{bmatrix} \tag{3.2.20}$$

Constitutive equations of each physical element

Elements constitutive (or terminal) equations

```
> const_elem_eqns :=
    <i__ph(t) = i__ph__law(t),
     i__d(t)   = I__ds*(exp(v__d(t)/(eta*V__th)) - 1),
     v__Rs(t)  = Rs*i__Rs(t),
     v__Rh(t)  = Rh*i__Rh(t),
```

```
   v__pv(t)  = v__pv__law(t)
  >: <%>
```

$$\begin{bmatrix} i_{ph}(t) = i_{ph_{law}}(t) \\ i_d(t) = I_{ds}\left(\mathrm{e}^{\frac{v_d(t)}{\eta V_{th}}} - 1\right) \\ v_{Rs}(t) = Rs\, i_{Rs}(t) \\ v_{Rh}(t) = Rh\, i_{Rh}(t) \\ v_{pv}(t) = v_{pv_{law}}(t) \end{bmatrix} \quad \textbf{(3.2.21)}$$

```
> i__s1(t),i__s2(t); # can be constnat or function of time (in
  this case via temperatures)
```

$$i_{s1}(t), i_{s2}(t) \quad \textbf{(3.2.22)}$$

```
> circuit_eqns;
  cutset_eqns;
```

$$\begin{bmatrix} v_{ph}(t) + v_{Rh}(t) \\ v_{ph}(t) + v_d(t) \\ v_{ph}(t) - v_{Rs}(t) + v_{pv}(t) \end{bmatrix}$$

$$\begin{bmatrix} i_{ph}(t) - i_d(t) - i_{Rh}(t) - i_{pv}(t) \\ i_{Rs}(t) + i_{pv}(t) \end{bmatrix} \quad \textbf{(3.2.23)}$$

```
> #latex(const_elem_eqns)
```

List of equations of photovoltaic cell

```
> pvc_eqns := <const_elem_eqns,circuit_eqns,cutset_eqns>
```

(3.2.24)

$$pvc\_eqns := \left[\begin{array}{c} i_{ph}(t) = i_{ph_{law}}(t) \\ i_{d}(t) = I_{ds}\left(e^{\frac{v_{d}(t)}{\eta V_{th}}} - 1\right) \\ v_{Rs}(t) = Rs\, i_{Rs}(t) \\ v_{Rh}(t) = Rh\, i_{Rh}(t) \\ v_{pv}(t) = v_{pv_{law}}(t) \\ v_{ph}(t) + v_{Rh}(t) \\ v_{ph}(t) + v_{d}(t) \\ v_{ph}(t) - v_{Rs}(t) + v_{pv}(t) \\ i_{ph}(t) - i_{d}(t) - i_{Rh}(t) - i_{pv}(t) \\ i_{Rs}(t) + i_{pv}(t) \end{array}\right] \quad \textbf{(3.2.24)}$$

Analytical solution of equaivalent circuit of a photovoltaic cell

```
> pvc_vars := [op(tau_vars),op(alpha_vars)];
```

$$pvc\_vars := \left[i_{ph}(t), i_{d}(t), i_{Rh}(t), i_{Rs}(t), i_{pv}(t), v_{ph}(t), v_{d}(t), v_{Rh}(t), v_{Rs}(t), v_{pv}(t)\right] \quad \textbf{(3.2.25)}$$

Check that variables are equal in number to the equations

```
> RowDimension(pvc_eqns) = nops(tau_vars)+nops(alpha_vars) ;
```

$$10 = 10 \quad \textbf{(3.2.26)}$$

```
> sol_pvc := fullsimplify(solve(convert(pvc_eqns,set),convert
  (pvc_vars,set))):
  <op(%)>:
```

Partial solution of the equations of the equivalent circuit to show the relationship of

```
> convert(convert(pvc_eqns,list)[1..-3],set) union convert
  (pvc_eqns[-1],set);
  convert(pvc_vars,set) minus {i__pv(t)};

  solve(%%,%);
  SDM_pv_eqn := collect(subs(%,pvc_eqns[-2]),I__ds);
```

$$\left\{ i_{Rs}(t) + i_{pv}(t), v_{ph}(t) + v_{Rh}(t), v_{ph}(t) + v_{d}(t), v_{ph}(t) - v_{Rs}(t) + v_{pv}(t), i_{d}(t)\right.$$

$$= I_{ds}\left(e^{\frac{v_d(t)}{\eta V_{th}}} - 1\right), i_{ph}(t) = i_{ph_{law}}(t), v_{Rh}(t) = Rh\, i_{Rh}(t), v_{Rs}(t) = Rs\, i_{Rs}(t), v_{pv}(t) = v_{pv_{law}}(t)\Bigg\}$$

$$\{i_{Rh}(t), i_{Rs}(t), i_d(t), i_{ph}(t), v_{Rh}(t), v_{Rs}(t), v_d(t), v_{ph}(t), v_{pv}(t)\}$$

$$\left\{i_{Rh}(t) = \frac{Rs\, i_{pv}(t) + v_{pv_{law}}(t)}{Rh}, i_{Rs}(t) = -i_{pv}(t), i_d(t) = I_{ds}\, e^{\frac{Rs\, i_{pv}(t) + v_{pv_{law}}(t)}{\eta V_{th}}} - I_{ds}, i_{ph}(t) = i_{ph_{law}}(t), v_{Rh}(t) = Rs\, i_{pv}(t) + v_{pv_{law}}(t), v_{Rs}(t) = -Rs\, i_{pv}(t), v_d(t) = Rs\, i_{pv}(t) + v_{pv_{law}}(t), v_{ph}(t) = -Rs\, i_{pv}(t) - v_{pv_{law}}(t), v_{pv}(t) = v_{pv_{law}}(t)\right\}$$

$SDM\_pv\_eqn :=$ **(3.2.27)**

$$\left[\left(-e^{\frac{Rs\, i_{pv}(t) + v_{pv_{law}}(t)}{\eta V_{th}}} + 1\right) I_{ds} + i_{ph_{law}}(t) - \frac{Rs\, i_{pv}(t) + v_{pv_{law}}(t)}{Rh} \cdots\right]$$

Output current of the cell

The pannel output current i__pv in term of the output voltage v_{pv} :

```
> simplify(solve(SDM_pv_eqn,{i__pv(t)}));
  tmp := rhs(%[1]);


  Iout := unapply(subs(i__ph__law(t) = I__ph, v__pv__law(t)=
  V__pv, tmp) ,[V__pv]);
```

$$\left\{i_{pv}(t) = \frac{1}{(Rh + Rs)\, Rs}\left(-(Rh + Rs)\, \eta\, V_{th}\, \mathrm{LambertW}\left(\frac{I_{ds}\, Rs\, Rh\, e^{\frac{\left(I_{ds}\, Rs + i_{ph_{law}}(t)\, Rs + v_{pv_{law}}(t)\right) Rh}{(Rh + Rs)\, \eta\, V_{th}}}}{(Rh + Rs)\, \eta\, V_{th}}\right) + Rs\left(I_{ds}\, Rh + i_{ph_{law}}(t)\, Rh - v_{pv_{law}}(t)\right)\right)\right\}$$

$$tmp := \frac{1}{(Rh + Rs)\,Rs}\left(-(Rh + Rs)\,\eta\,V_{th}\,\mathrm{LambertW}\left(\frac{I_{ds}\,Rs\,Rh\,\mathrm{e}^{\frac{\left(I_{ds}\,Rs + i_{ph_{law}}(t)\,Rs + v_{pv_{law}}(t)\right)Rh}{(Rh+Rs)\,\eta\,V_{th}}}}{(Rh + Rs)\,\eta\,V_{th}}\right) + Rs\left(I_{ds}\,Rh + i_{ph_{law}}(t)\,Rh - v_{pv_{law}}(t)\right)\right)$$

$$Iout := V_{pv} \mapsto \frac{1}{(Rh + Rs)\cdot Rs}\left(-(Rh + Rs)\cdot\eta\cdot V_{th}\cdot\mathrm{LambertW}\left(\frac{I_{ds}\cdot Rs\cdot Rh\cdot\mathrm{e}^{\frac{\left(I_{ds}\cdot Rs + I_{ph}\cdot Rs + V_{pv}\right)\cdot Rh}{(Rh+Rs)\cdot\eta\cdot V_{th}}}}{(Rh + Rs)\cdot\eta\cdot V_{th}}\right) + Rs\cdot\left(I_{ds}\cdot Rh + I_{ph}\cdot Rh - V_{pv}\right)\right) \tag{3.2.28}$$

Output voltage of the cell

We can also solve equation and give the expression of the output voltage V_{out} in term of the output current I_{out} :

```
> simplify(solve(SDM_pv_eqn,{v__pv__law(t)}));
  tmp := rhs(%[1]);


  Vout := unapply(subs(i__ph__law(t) = I__ph, i__pv(t)= I__pv,
  v__pv__law(t)= V__pv, tmp) ,[I__pv]);
```

$$\left\{v_{pv_{law}}(t) = -\mathrm{LambertW}\left(\frac{I_{ds}\,Rh\,\mathrm{e}^{\frac{Rh\left(I_{ds} - i_{pv}(t) + i_{ph_{law}}(t)\right)}{\eta\,V_{th}}}}{\eta\,V_{th}}\right)\eta\,V_{th} + (-Rh - Rs)\,i_{pv}(t) + \left(I_{ds} + i_{ph_{law}}(t)\right)Rh\right\}$$

$$tmp := -\mathrm{LambertW}\left(\frac{I_{ds}\,Rh\,\mathrm{e}^{\frac{Rh\left(I_{ds}-i_{pv}(t)+i_{ph_{law}}(t)\right)}{\eta\,V_{th}}}}{\eta\,V_{th}}\right)\eta\,V_{th}+(-Rh-Rs)\,i_{pv}(t)+\left(I_{ds}+i_{ph_{law}}(t)\right)Rh$$

$$Vout := I_{pv}\mapsto -\mathrm{LambertW}\left(\frac{I_{ds}\cdot Rh\cdot \mathrm{e}^{\frac{Rh\cdot\left(I_{ds}-I_{pv}+I_{ph}\right)}{\eta\cdot V_{th}}}}{\eta\cdot V_{th}}\right)\cdot\eta\cdot V_{th}+(-Rh-Rs)\cdot I_{pv}+\left(I_{ds}+I_{ph}\right)\cdot Rh \tag{3.2.29}$$

Short-circuit current I_{sc}

The analytical expression of the short-circuit current I_{sc} is obtained by setting $V_{out}=0$ which corresponds to $R_{load}=0$:

```
> Isc:=Iout(0);
```

$$Isc := \frac{1}{(Rh+Rs)\,Rs}\left(-(Rh+Rs)\,\eta\,V_{th}\,\mathrm{LambertW}\left(\frac{I_{ds}\,Rs\,Rh\,\mathrm{e}^{\frac{\left(I_{ds}\,Rs+I_{ph}\,Rs\right)Rh}{(Rh+Rs)\,\eta\,V_{th}}}}{(Rh+Rs)\,\eta\,V_{th}}\right)+Rs\left(I_{ds}\,Rh+I_{ph}\,Rh\right)\right) \tag{3.2.30}$$

Open-circuit voltage V_{oc}

The analytical expression of the open-circuit voltage V_{oc} is obtained by setting $I_{out}=0$ which corresponds to $R_{load}=\infty$:

```
> Voc := Vout(0);
```

$$Voc := -\mathrm{LambertW}\left(\frac{I_{ds}\,Rh\,\mathrm{e}^{\frac{Rh\left(I_{ds}+I_{ph}\right)}{\eta\,V_{th}}}}{\eta\,V_{th}}\right)\eta\,V_{th}+\left(I_{ds}+I_{ph}\right)Rh \tag{3.2.31}$$

Numerical example

Solar induced current:

```
> i__ph = I__ph0*I__r/I__r0;

  # Ir is the irradiance (light intensity), in W/m2, falling on
```

```
the cell.
# Iph0 is the measured solar-generated current for the
irradiance Ir0.
```

$$i_{ph} = \frac{I_{ph0}\, I_r}{I_{r0}} \quad \textbf{(3.3.1)}$$

The previous equation will be used in the next section to model the solar current

```
> data_Ns := k__B=1.38062*10^(-23), T =300, q =1.60219*10^(-19),

          I__ds = Np*1.2172e-9,
          Rs = 9.2/1000*(Ns/Np), Rh =12.7*(Ns/Np), eta = 1.13;
  subs(%,lhs(eq_Vth) = rhs(eq_Vth)*Ns );
```

$$data\_Ns := k_B = 1.380620000 \times 10^{-23}, T = 300, q = 1.602190000 \times 10^{-19}, I_{ds} = 1.2172 \times 10^{-9}\, Np, Rs = \frac{0.009200000000\, Ns}{Np}, Rh = \frac{12.7\, Ns}{Np}, \eta = 1.13$$

$$V_{th} = 0.02585124111\, Ns \quad \textbf{(3.3.2)}$$

```
> data_Ns_full  := subs(Ns = 56,Np = 2, [% , %%]);
```

$$data\_Ns\_full := \left[V_{th} = 1.447669502, k_B = 1.380620000 \times 10^{-23}, T = 300, q = 1.602190000 \times 10^{-19}, I_{ds} = 2.4344 \times 10^{-9}, Rs = 0.2576000000, Rh = 355.6000000, \eta = 1.13\right] \quad \textbf{(3.3.3)}$$

```
> I__ph1 := subs(Np = 2,Np * 6.1);
```

$$I_{ph1} := 12.2 \quad \textbf{(3.3.4)}$$

```
> Voc_num := evalf( subs( data_Ns_full,I__ph = I__ph1, Voc));
  Isc_num := evalf( subs( data_Ns_full,I__ph = I__ph1, Isc));
```

$$Voc\_num := 36.523249$$

$$Isc\_num := 12.19116859 \quad \textbf{(3.3.5)}$$

```
> Pow_vol:=V__pv->V__pv*Iout(V__pv);

  diff(Pow_vol(V__pv),V__pv) = 0;
  subs(data_Ns_full,I__ph = I__ph1,%);

  V__pv__max := fsolve(%,V__pv);
  I__pv__max := evalf( subs(data_Ns_full,I__ph = I__ph1,Iout
  (V__pv__max)) );
  Pow_max    := evalf( subs(data_Ns_full,I__ph = I__ph1,Pow_vol
  (V__pv__max)));
```

$$Pow\_vol := V_{pv} \mapsto V_{pv} \cdot Iout(V_{pv})$$

$$\frac{1}{(Rh+Rs)\,Rs}\left(-(Rh+Rs)\,\eta\,V_{th}\,\mathrm{LambertW}\left(\frac{I_{ds}\,Rs\,Rh\,\mathrm{e}^{\frac{(I_{ds}\,Rs+I_{ph}\,Rs+V_{pv})\,Rh}{(Rh+Rs)\,\eta\,V_{th}}}}{(Rh+Rs)\,\eta\,V_{th}}\right) + Rs\,\left(I_{ds}\,Rh+I_{ph}\,Rh-V_{pv}\right)\right) + \frac{V_{pv}\left(-\frac{Rh\,\mathrm{LambertW}\left(\frac{I_{ds}\,Rs\,Rh\,\mathrm{e}^{\frac{(I_{ds}\,Rs+I_{ph}\,Rs+V_{pv})\,Rh}{(Rh+Rs)\,\eta\,V_{th}}}}{(Rh+Rs)\,\eta\,V_{th}}\right)}{1+\mathrm{LambertW}\left(\frac{I_{ds}\,Rs\,Rh\,\mathrm{e}^{\frac{(I_{ds}\,Rs+I_{ph}\,Rs+V_{pv})\,Rh}{(Rh+Rs)\,\eta\,V_{th}}}}{(Rh+Rs)\,\eta\,V_{th}}\right)} - Rs\right)}{(Rh+Rs)\,Rs} = 0$$

$$-6.350413578\,\mathrm{LambertW}\left(3.830676137\times10^{-10}\,\mathrm{e}^{1.919744038+0.6108543039\,V_{pv}}\right) + 12.19116861 - 0.002810112810\,V_{pv} + 0.01090882302\,V_{pv}\left(-\frac{355.6000000\,\mathrm{LambertW}\left(3.830676137\times10^{-10}\,\mathrm{e}^{1.919744038+0.6108543039\,V_{pv}}\right)}{1+\mathrm{LambertW}\left(3.830676137\times10^{-10}\,\mathrm{e}^{1.919744038+0.6108543039\,V_{pv}}\right)} - 0.2576000000\right) = 0$$

$$V_{pv_{\max}} := 28.95339440$$

$$I_{pv_{\max}} := 11.39824241$$

$$Pow\_max := 330.0178078 \tag{3.3.6}$$

lAnother way to determine the maximum power and the corresponding voltage :

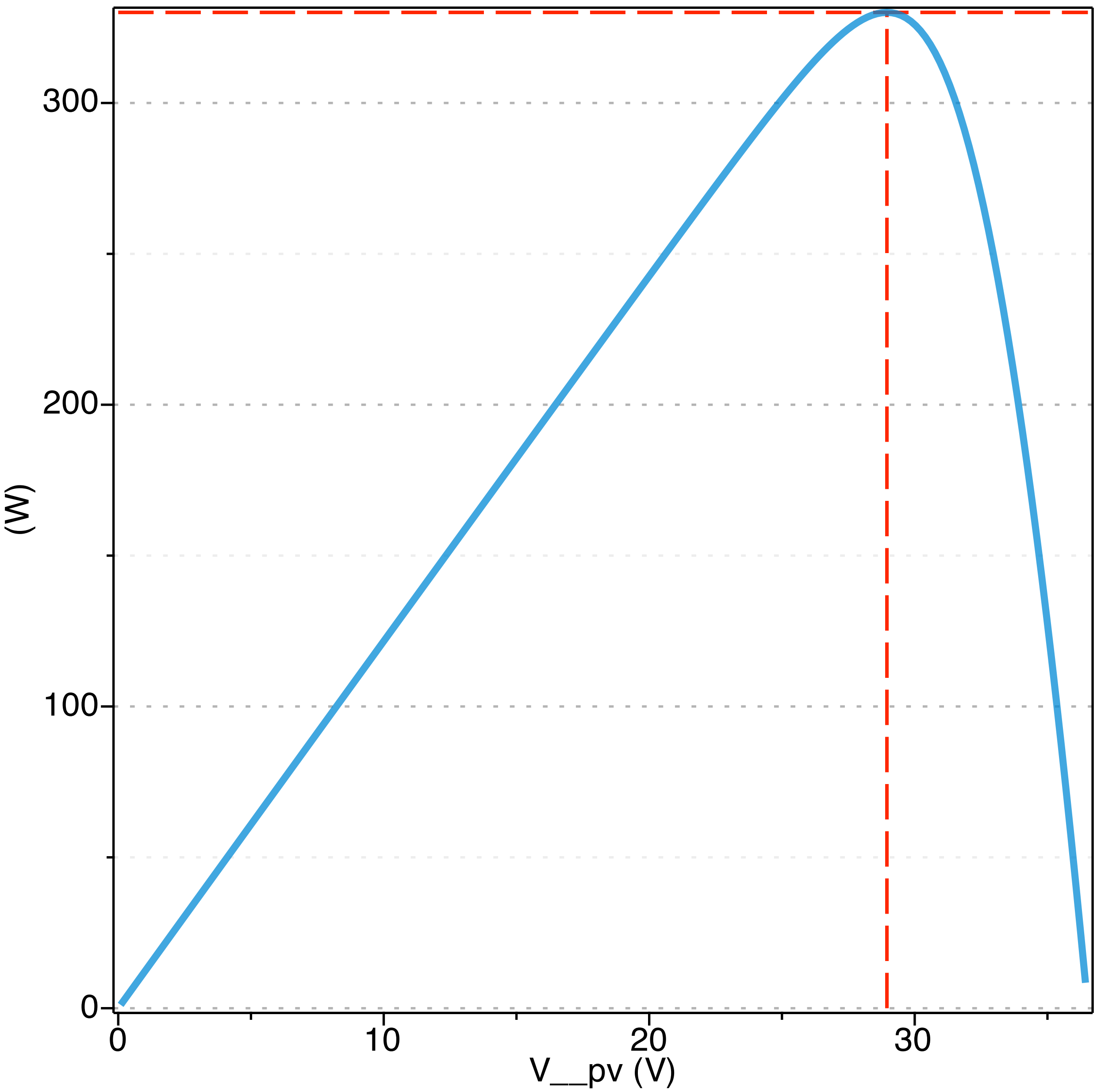
```
> MPP_vol:=Optimization:-Maximize(subs(data_Ns_full,I__ph =
  I__ph1,Pow_vol(V__pv)),{0<=V__pv,V__pv<=Voc_num});
```

$$MPP\_vol := \left[330.017807675594725,\ \left[V_{pv} = 28.9533944028819\right]\right] \tag{3.3.7}$$

Power versus voltage

```
> display( plot(Pow_max,  V__pv = 0..Voc_num,linestyle=dash,
  color="Red"),
```

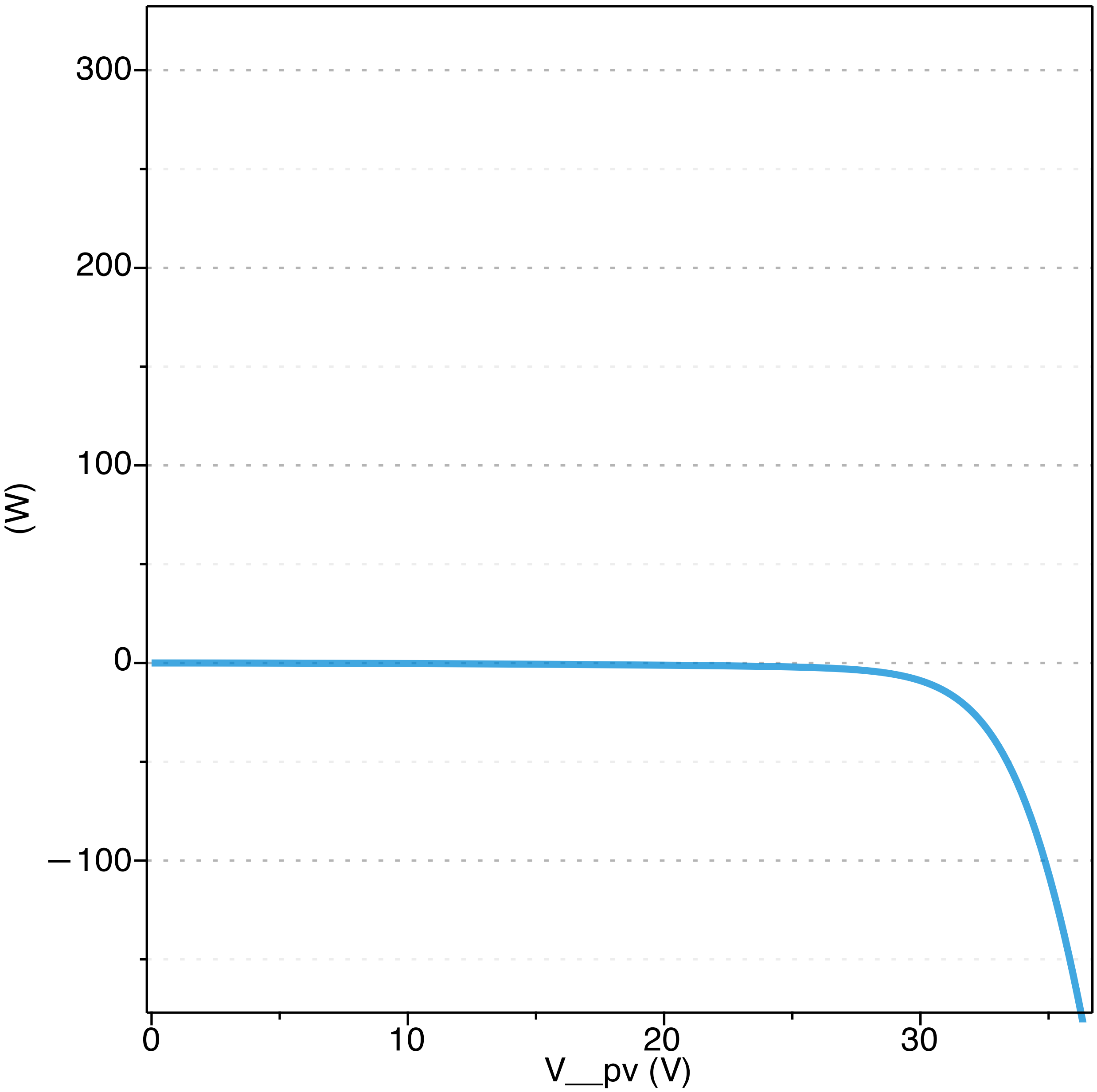
```
        pointplot([[V__pv__max,  0],[V__pv__max,  Pow_max]],
linestyle=dash,color="Red",connect=true),
        plot(subs(data_Ns_full,I__ph = I__ph1,Pow_vol(V__pv)
),V__pv = 0..Voc_num, thickness    = 3, transparency = 0.25),
        labels = ["V__pv (V)","(W)"],
        size   = [800,500])
```



Animate the effect of photo current variation vs voltage

```
> animate(plot,[subs(data_Ns_full,Pow_vol(V__pv)),V__pv = 0..
Voc_num],I__ph = 0..I__ph1,
        thickness    = 3, transparency = 0.25,
        labels = ["V__pv (V)","(W)"],
```

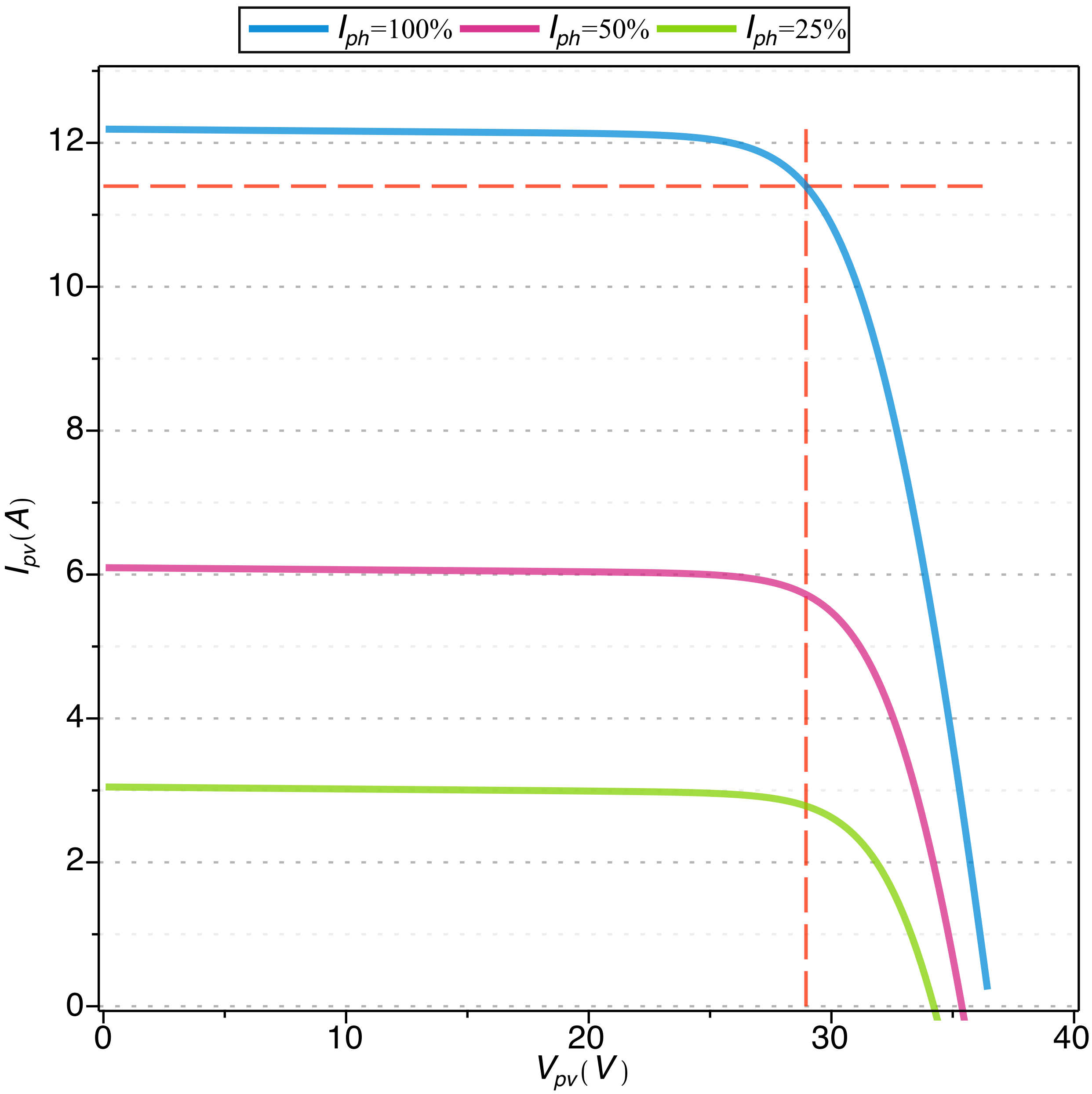
```
         size   = [800,500])
```



I-V curve of a photovoltaic panel

```
> display( plot(I__pv__max,  V__pv = 0..Voc_num,linestyle=dash,
  color="Red",thickness=1),
          pointplot([[V__pv__max,  0],[V__pv__max,  Isc_num]],
  linestyle=dash,thickness=1,color="Red",connect=true),
          plot(subs(data_Ns_full,I__ph = I__ph1,       Iout
  (V__pv)),V__pv = 0..Voc_num,color=c_set[1],
                legend=typeset(I__ph,"=100%")),
          plot(subs(data_Ns_full,I__ph = I__ph1*0.5,   Iout
  (V__pv)),V__pv = 0..Voc_num,color=c_set[2],
```

```
                legend=typeset(I__ph,"=50%")),
          plot(subs(data_Ns_full,I__ph = I__ph1*0.25,  Iout
(V__pv)),V__pv = 0..Voc_num, color=c_set[3],
                legend=typeset(I__ph,"=25%")),
          labels = ['V__pv (V)','I__pv (A)'],
          thickness= 3, transparency = 0.25,
          view = [0..40,0..13],
         size         = [800,500]);
```



1) Simulation with external load with voltage ramp for load

Simulation with constant external load

In this simulation, it is explored the behaviour of a photovoltaic (PV) panel system connected to a constant load over a 24-hour period. The purpose of this study is to understand how different factors such as irradiance and load demand affect the performance of the PV panel throughout the day.

Equations of pannel connected to a constant load

```
> eqns := [I__pv(t) - Iout(V__pv(t)),
           V__pv(t) =  Pow/I__pv(t)
           ]
  :
  <%>
```

$$\left[I_{pv}(t) - \frac{-(Rh+Rs)\,\eta\, V_{th}\,\mathrm{LambertW}\left(\frac{I_{ds}\, Rs\, Rh\, \mathrm{e}^{\frac{(I_{ds}\, Rs + I_{ph}\, Rs + V_{pv}}{(Rh+Rs)\,\eta\, V_{t}}}}{(Rh+Rs)\,\eta\, V_{th}} \dots \right.}{(Rh+Rs)\, Rs},\; V_{pv}(t) = \frac{Pow}{I_{pv}(t)} \dots \right] \quad (4.1.1)$$

```
> #sol := op(solve(eqns,[I__pv(t),V__pv(t)]))
> [I__pv(t),V__pv(t)];
  x2y := {seq(%[i] = op(0,%[i]),i=1..nops(%))};
```

$$[I_{pv}(t), V_{pv}(t)]$$

$$x2y := \{I_{pv}(t) = I_{pv}, V_{pv}(t) = V_{pv}\} \quad (4.1.2)$$

Simple case with constant Solar induced current I__ph = 6 and fixed value of Power request (Pow = [40, 60, 80, 100])

```
> # Power set to 40 W
  sol := fsolve(convert(subs(data_Ns_full,I__ph = 6, Pow = 40,
  x2y, eqns),set),subs( x2y, {I__pv(t),V__pv(t)}));
```

$$sol := \{I_{pv} = 5.976849671, V_{pv} = 6.692488886\} \quad (4.1.3)$$

```
> # Power set to 60 W
  sol := fsolve(convert(subs(data_Ns_full,I__ph = 6, Pow = 60,
  x2y, eqns),set),subs( x2y, {I__pv(t),V__pv(t)}));
```

$$sol := \{I_{pv} = 5.967399136, V_{pv} = 10.05463161\} \quad (4.1.4)$$

```
> # Power set to 80 W
  sol := fsolve(convert(subs(data_Ns_full,I__ph = 6, Pow = 80,
  x2y, eqns),set),subs( x2y, {I__pv(t),V__pv(t)}));
```

$$sol := \{I_{pv} = 5.957900949, V_{pv} = 13.42754784\} \quad (4.1.5)$$

```
> # Power set to 100 W
```

```
sol := fsolve(convert(subs(data_Ns_full,I__ph = 6, Pow = 100,
x2y, eqns),set),subs( x2y, {I__pv(t),V__pv(t)}));
```

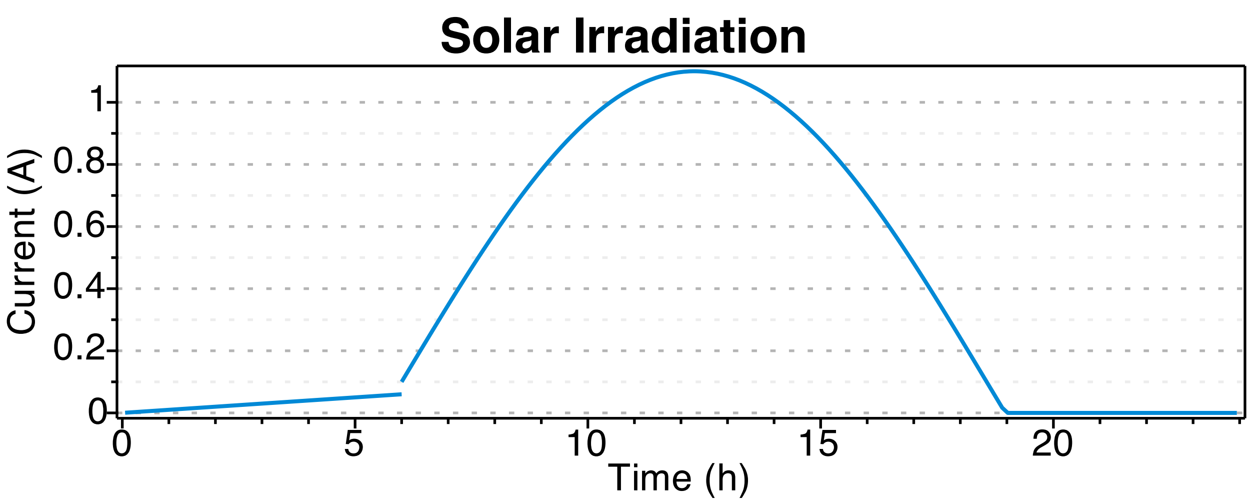
$$sol := \{I_{pv} = 5.948233542, V_{pv} = 16.81171381\} \quad \textbf{(4.1.6)}$$

Simulation with costant load and Solar Irradiance modelled as a piecewise sinusoidal function

> Model the solar Irradiance I__r as a sinusoidal funtion with a peak at 12:17

```
I__r := t -> piecewise(t < 6, 0.01*t,  t < 19, sin(t/4-1.5)
+0.1, 0);

# Plot Pow over a day
plot(I__r(t), t = 0 .. 24, title = "Solar Irradiation", labels
= ["Time (h)", "Current (A)"]);
```

$$I_r := t \mapsto \begin{cases} 0.01 \cdot t & t < 6 \\ \sin\left(\frac{t}{4} - 1.5\right) + 0.1 & t < 19 \\ 0 & otherwise \end{cases}$$



Compute the maximum solar peak

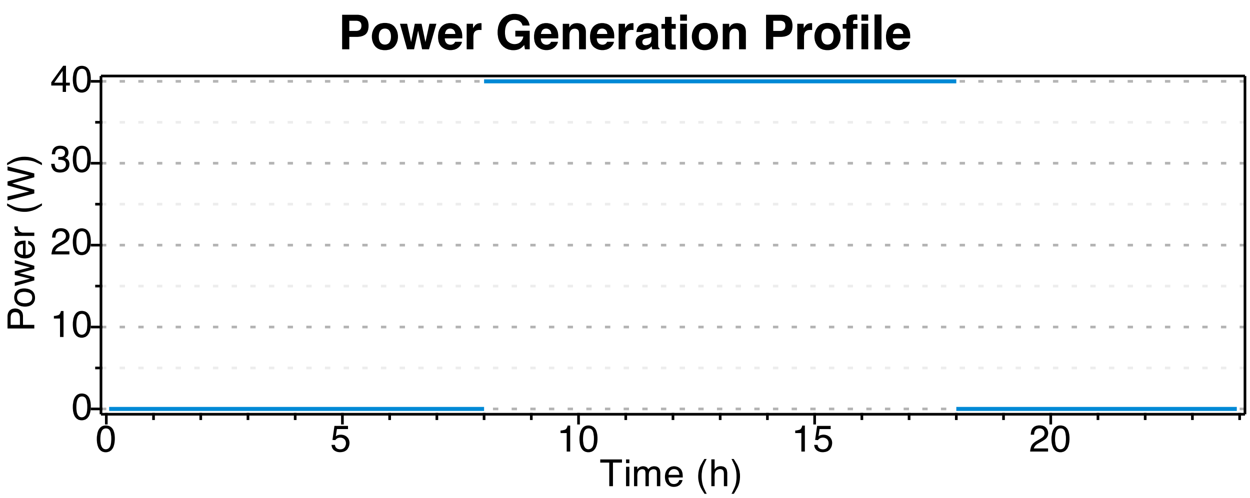
```
> maximize(I__r(t),t=1..24,location);
```

$$1.100000000, \{[\{t = 12.28318531\}, 1.100000000]\} \quad \textbf{(4.1.7)}$$

> Constant Power requested modelled as a piecewise function

```
Pow__cons := t -> piecewise(t < 8, 0, t< 18, 40, 0);

#Plot Pow over a day
plot(Pow__cons(t), t = 0 .. 24, title = "Power Generation
Profile", labels = ["Time (h)", "Power (W)"]);
```

$$Pow_{cons} := t \mapsto \begin{cases} 0 & t < 8 \\ 40 & t < 18 \\ 0 & otherwise \end{cases}$$



In this simulation, it is explored the behavior of a photovoltaic (PV) panel system connected to a constant load over a 24-hour period. The purpose of this study is to understand how different factors such as irradiance and load demand affect the performance of the PV panel system throughout the day.

Using the following code, the result from the previous iteration are considered at the iteration of fsolve.

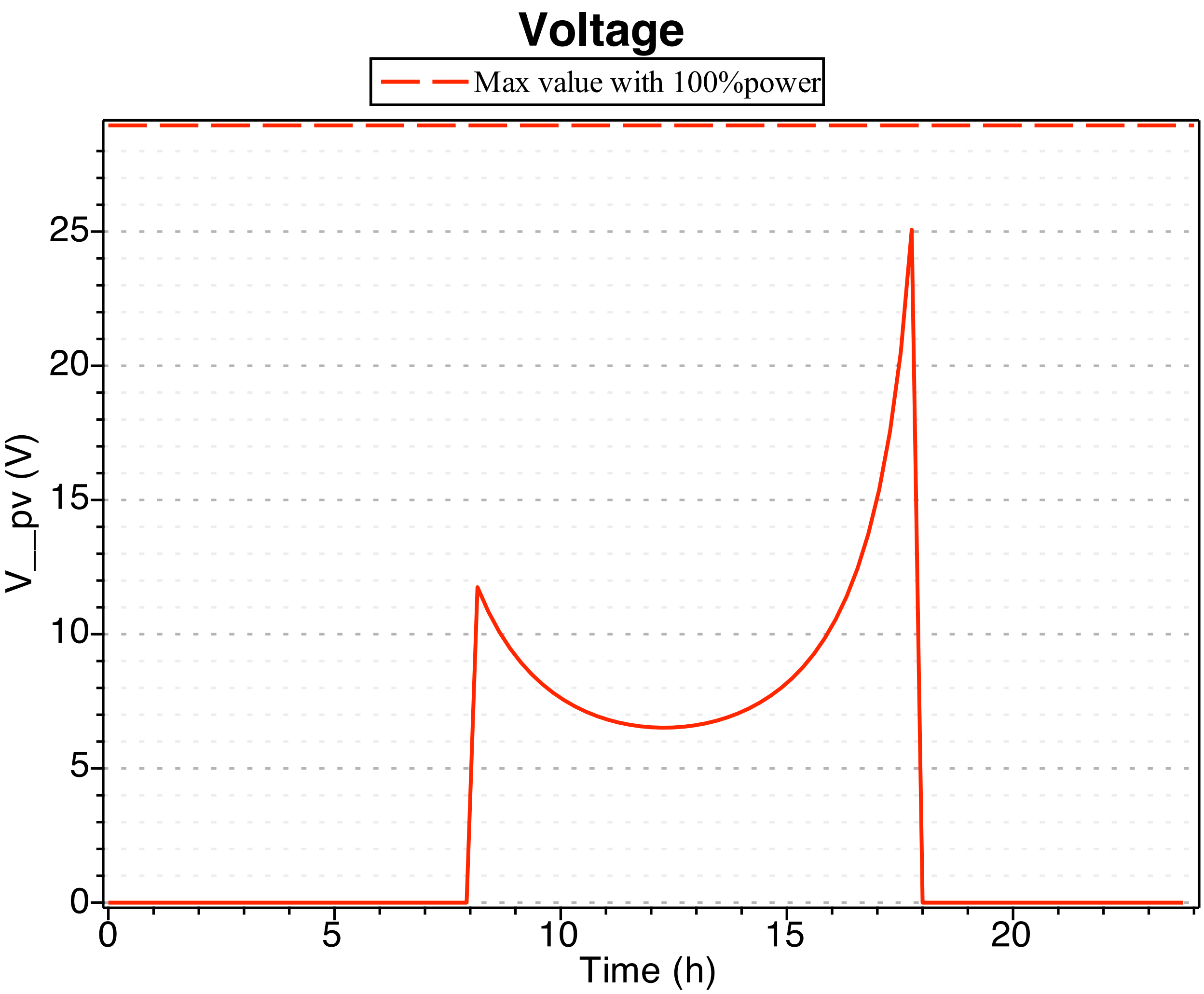
```
> Ns := 100:
  Tf := 24:

  T__VEC := [seq(Tf/Ns*i, i=0..Ns)]:

  equations := subs(V__pv(t) = V_pv, I__pv(t) = I_pv,eval({
                I__pv(t) = Iout(V__pv(t)),
                V__pv(t) = Pow/I__pv(t)
               })):


  # Define the variable set for the solver
  vars:= {V_pv,I_pv}:
```

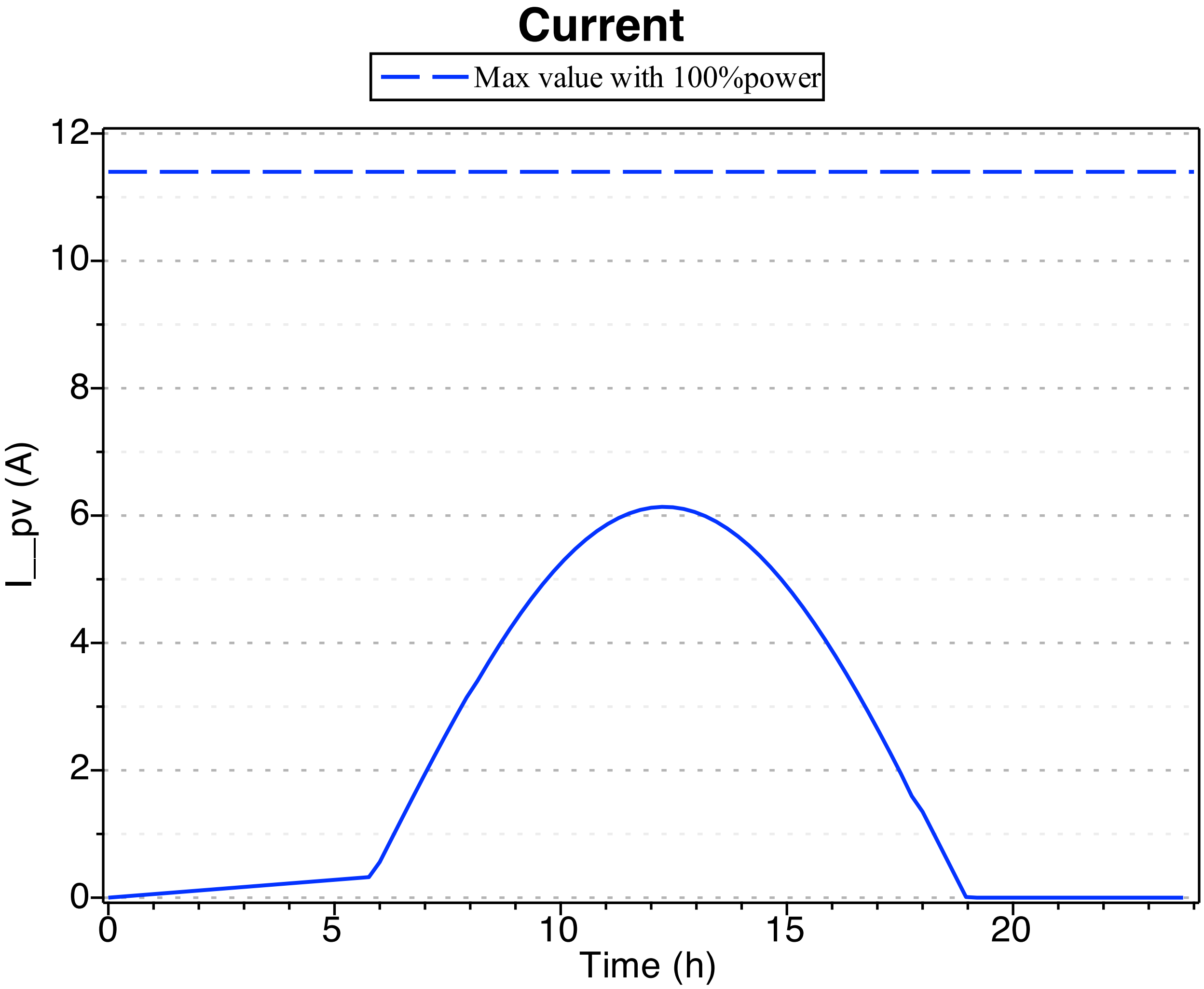
```
> tmp := fsolve(subs(data_Ns_full, t=T__VEC[1], I__ph = 7*800*
  I__r(0)/1000,Pow=Pow__cons(0),equations),vars):
  subs(data_Ns_full,I_ph = 9*800*I__r(0)/1000, t=T__VEC[1],eqns)
  :
```

```
sol := Matrix(1..Ns, 1..4, fill=0): # creates a matrix full of
zeroes
```

```
> for k from 1 to Ns do
;
tmp := fsolve(subs(data_Ns_full, t=T__VEC[k],I__ph = 7*800*
I__r(T__VEC[k])/1000,Pow=Pow__cons(T__VEC[k]),equations),tmp):
  sol[k,1..4] := (subs(t=T__VEC[k], tmp, <t,V_pv,I_pv,(V_pv*
I_pv)>)):

end:

sol:
pointp1 := pointplot(sol[1..-1,1],sol[1..-1,2],connect=true,
color=red, symbolsize=20,view=[default, 0..26],title =
"Voltage", labels = ["Time (h)", "V__pv (V)"]):
pointp2 := pointplot(sol[1..-1,1],sol[1..-1,3],connect=true,
color=blue, symbolsize=20,view=[default, 0..12],title =
"Current", labels = ["Time (h)", "I__pv (A)"]):
```

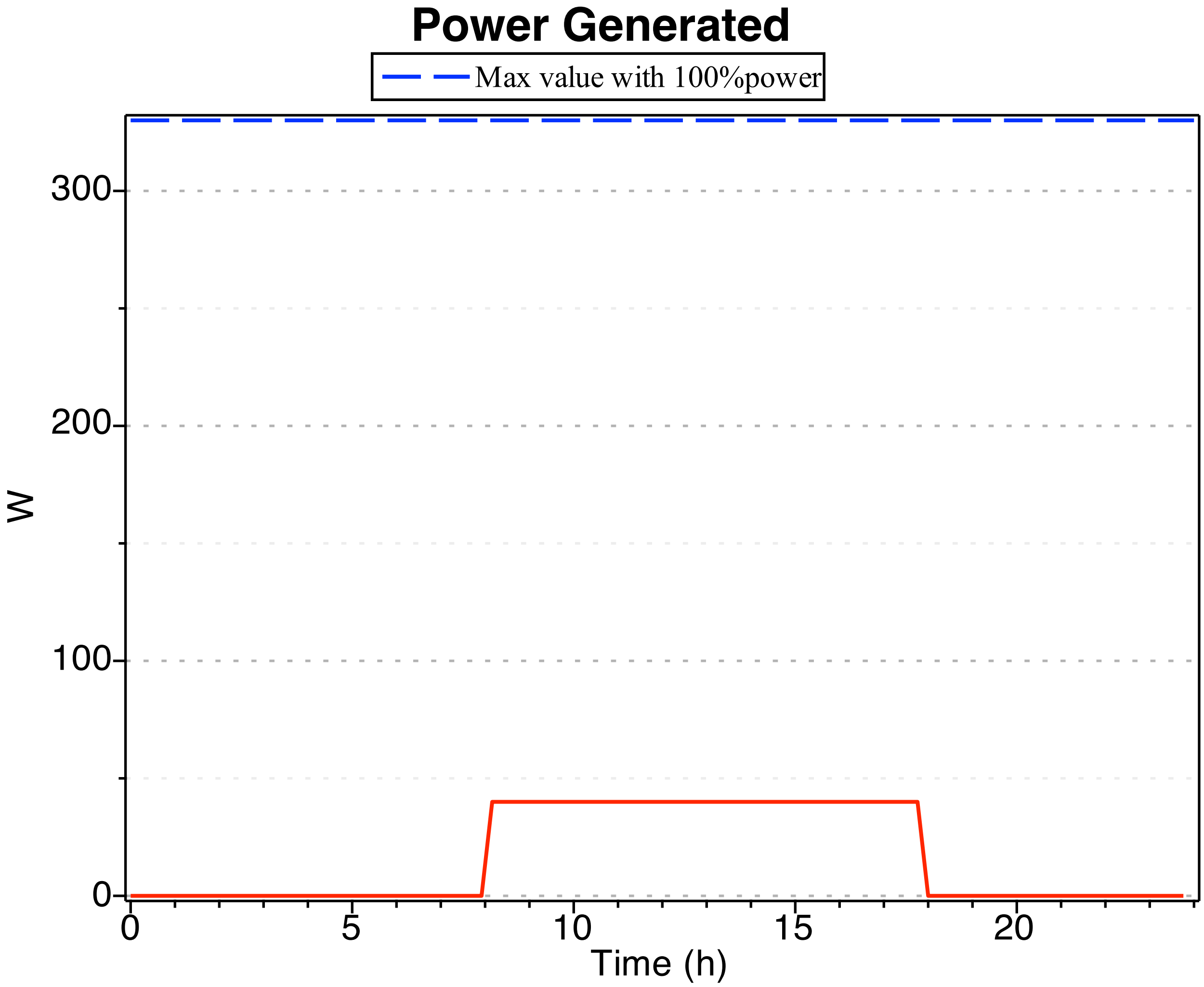
```
> line1 := plot(V__pv__max, x = 0..24, color = "red", linestyle
= dash, legend = "Max value with 100%power"):
line2 := plot(I__pv__max, x = 0..24, color = "blue", linestyle
= dash, legend = "Max value with 100%power"):

display(pointp1,line1,size = [600, 400]);
display(pointp2,line2,size = [600, 400]);
```


Voltage
Max value with 100%power
V__pv (V)
0
5
10
15
20
25
Time (h)
0
5
10
15
20



In the following plot it is display the Power generated by the solar panel (multiplying the voltage times the current) obtaining the same the Load requested defined in the piecewise function "Pow__cons"

```
> pointp3 := pointplot(sol[1..-1,1],sol[1..-1,4],connect=true,
  color=red, symbolsize=20,view=[default, 0..80],title = "Power
  Generated", labels = ["Time (h)", "W"]):

  line3 := plot(Pow_max, x = 0..24, color = "blue", linestyle =
  dash, legend = "Max value with 100%power"):

  display(pointp3,line3,size = [600, 400]);
```

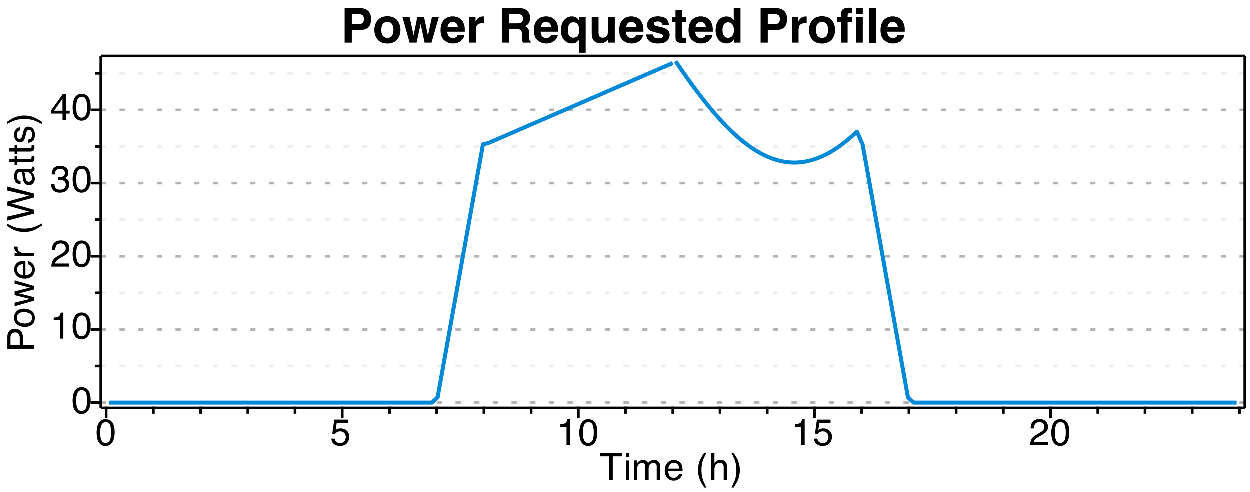


Simulation with various time law of the external load

Define the time varying Power modelled as a piecewise function

```
> Pow := t -> piecewise(0.4*(t < 7, 0,t < 8, 90*(t-7), t < 12,
  32 + 7 * t, t < 16, 132 - 50 * sin(t/2 - 12), t<17,90-90*
  (t-16), 0));

  #Plot Pow over a day
  plot(Pow(t), t = 0 .. 24, title = "Power Requested Profile",
  labels = ["Time (h)", "Power (Watts)"]);
```

$$Pow := t \mapsto \Big\{\ 0.4\cdot\left(t < 7, 0, t < 8, 90\cdot t - 630, t < 12, 32 + 7\cdot t, t < 16, 132 - 50\cdot\sin\left(\frac{t}{2} - 12\right), t <$$



The procedure to solve the system in the different hours during the day

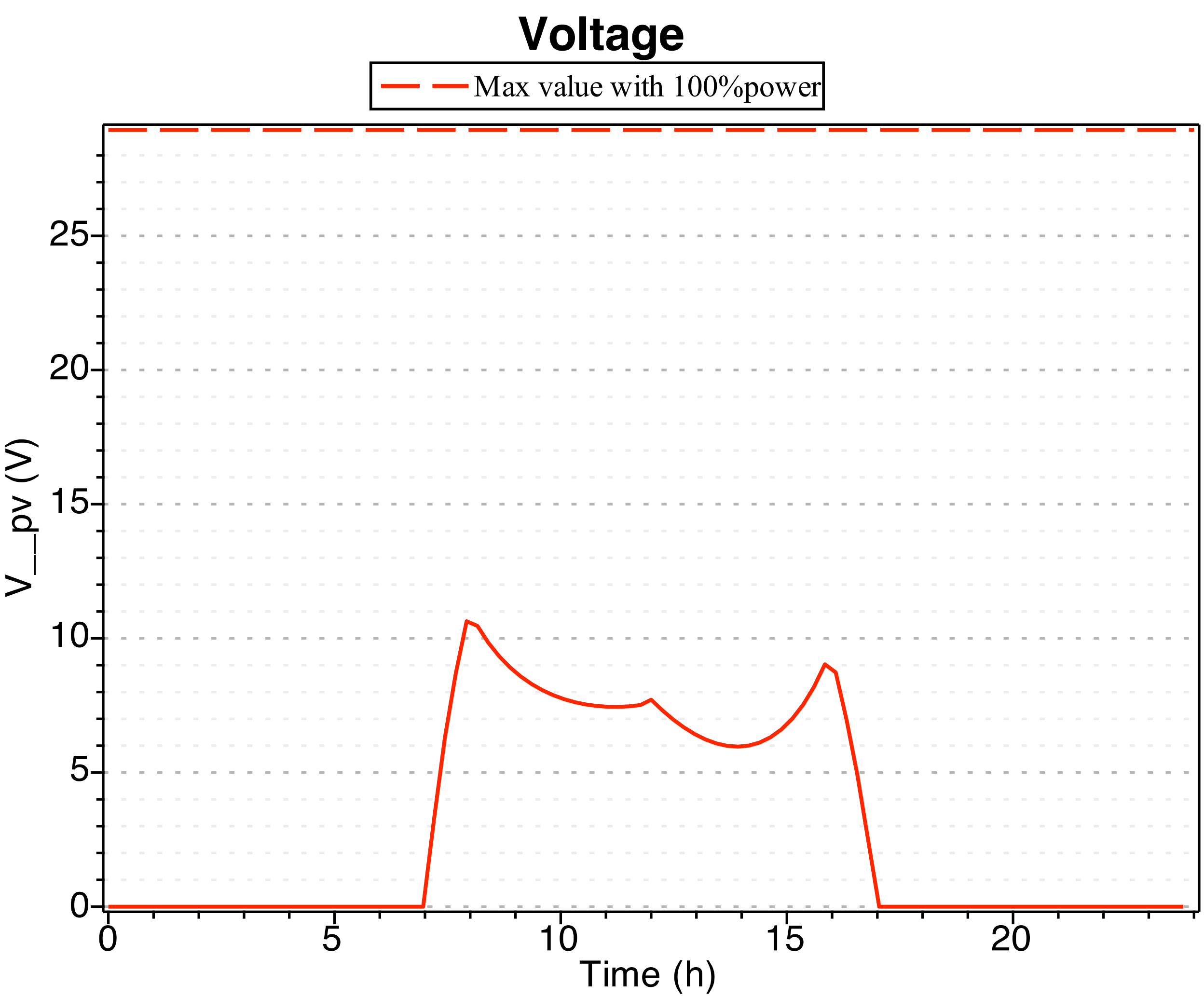
```
> Ns := 100:
  Tf := 24:

  T__VEC := [seq(Tf/Ns*i, i=0..Ns)]:

  equations := subs(V__pv(t) = V_pv, I__pv(t) = I_pv,eval({
                I__pv(t) = Iout(V__pv(t)),
                V__pv(t) = Pow/I__pv(t)
               })):


  # Define the variable set for the solver
  vars:= {V_pv,I_pv}:
```

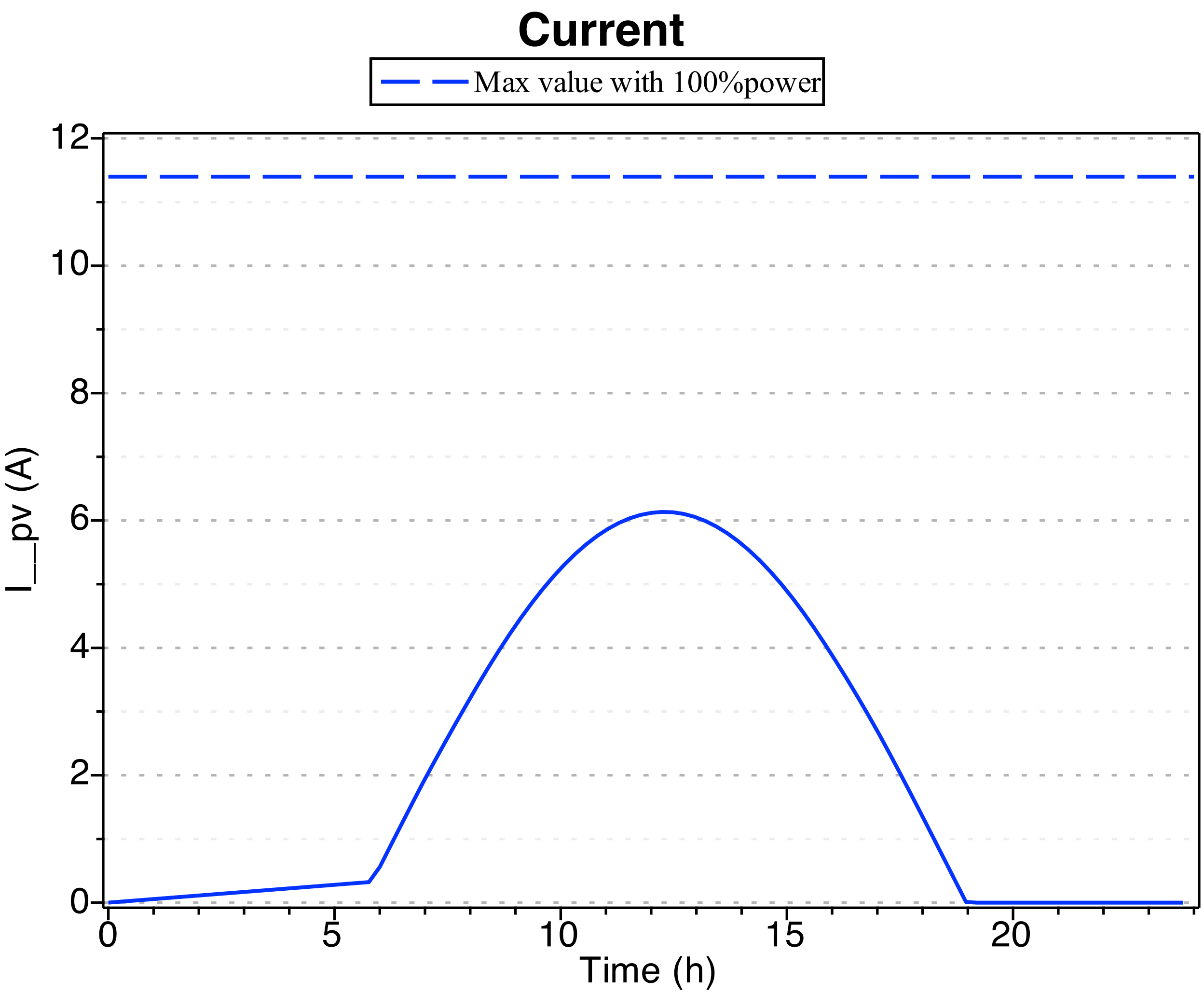
```
> tmp := fsolve(subs(data_Ns_full, t=T__VEC[1], I__ph = 7*800*
  I__r(0)/1000,Pow=Pow(0),equations),vars):
  subs(data_Ns_full,I_ph = 7*800*I__r(0)/1000, t=T__VEC[1],eqns)
  :

  sol := Matrix(1..Ns, 1..4, fill=0): # creates a matrix full of
  zeroes
```

```
> for k from 1 to Ns do
  ;
  tmp := fsolve(subs(data_Ns_full, t=T__VEC[k],I__ph = 7*800*
  I__r(T__VEC[k])/1000,Pow=Pow(T__VEC[k]),equations),tmp):
    sol[k,1..4] := (subs(t=T__VEC[k], tmp, <t,V_pv,I_pv,(V_pv*
  I_pv)>)):
```

```
end:

sol:
pointp1 := pointplot(sol[1..-1,1],sol[1..-1,2],connect=true,
color=red, symbolsize=20,view=[default, 0..16],title =
"Voltage", labels = ["Time (h)", "V__pv (V)"]):
pointp2 := pointplot(sol[1..-1,1],sol[1..-1,3],connect=true,
color=blue, symbolsize=20,view=[default, 0..12],title =
"Current", labels = ["Time (h)", "I__pv (A)"]):
```

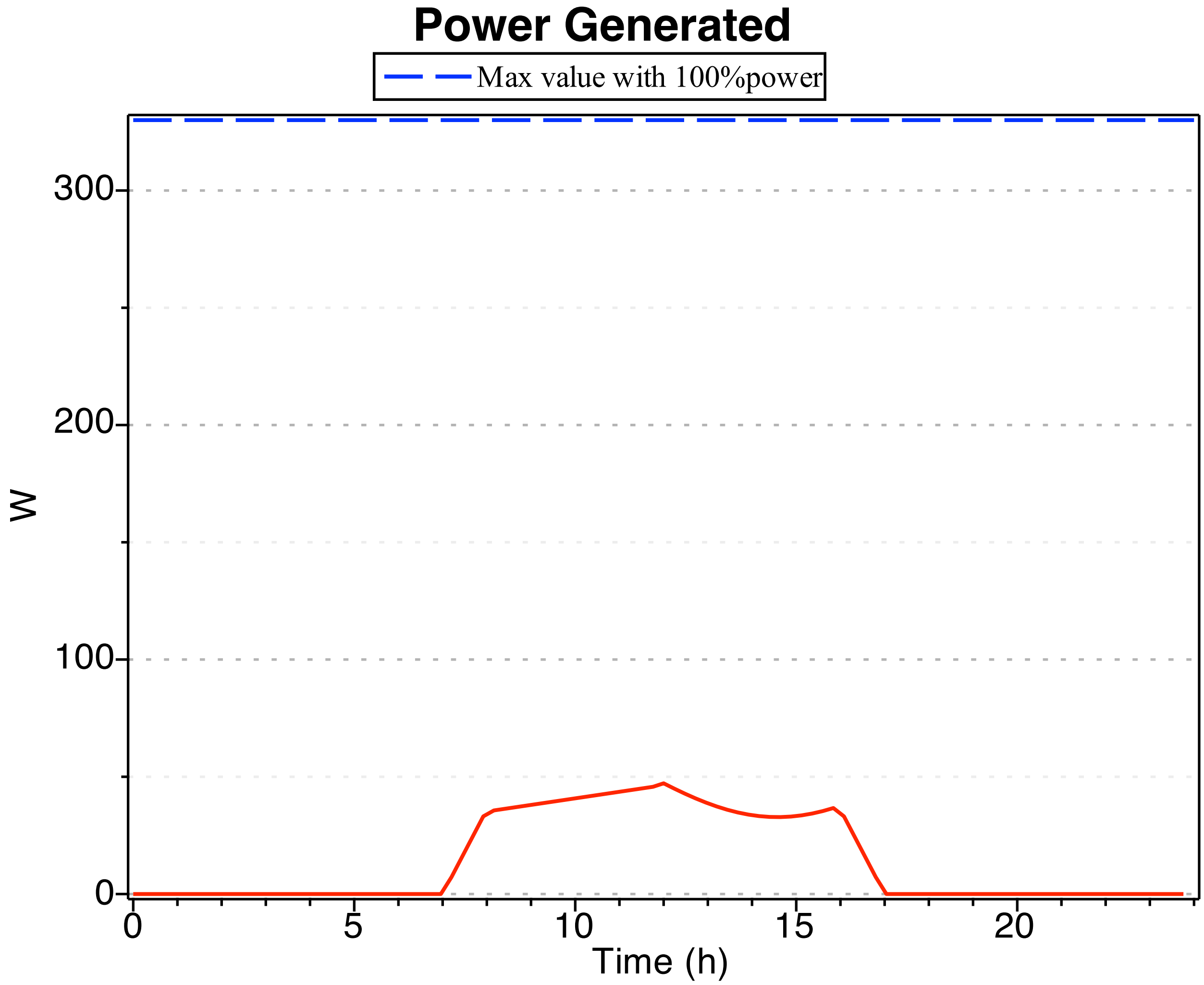
```
> line1 := plot(V__pv__max, x = 0..24, color = "red", linestyle
  = dash, legend = "Max value with 100%power"):
  line2 := plot(I__pv__max, x = 0..24, color = "blue", linestyle
  = dash, legend = "Max value with 100%power"):

  display(pointp1,line1,size = [600, 400]);
  display(pointp2,line2,size = [600, 400]);
```





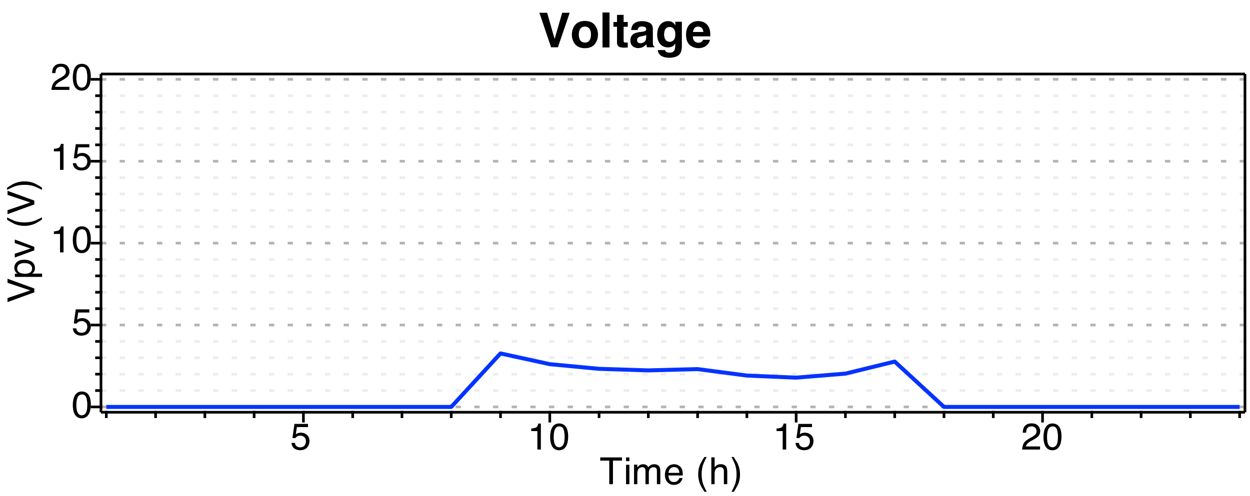
In the following plot it is display the Power generated by the solar panel (multiplying the voltage times the current) obtaining the same the Load requested defined in the piecewise function "Pow"

```
> pointp3 := pointplot(sol[1..-1,1],sol[1..-1,4],connect=true,
  color=red, symbolsize=20,view=[default, 0..80],title = "Power
  Generated", labels = ["Time (h)", "W"]):

  line3 := plot(Pow_max, x = 0..24, color = "blue", linestyle =
  dash, legend = "Max value with 100%power"):

  display(pointp3,line3,size = [600, 400]);
```

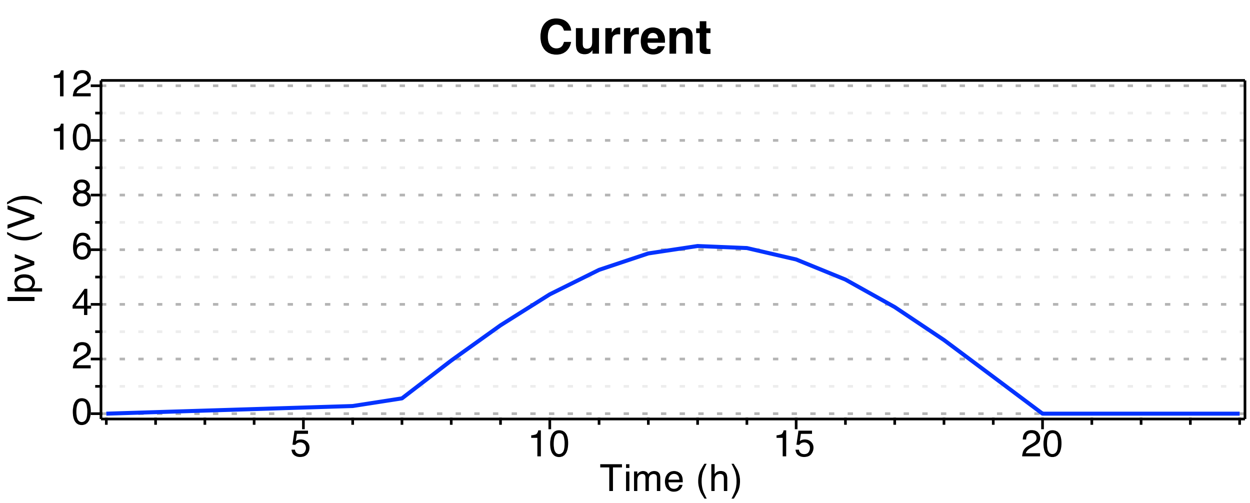


Another procedure to solve the system in the different hours during the day that do not considers the result from the previous iteration and it is less precise

```
> # Define the procedure
  solpow := proc(n):
      fsolve(convert(subs(data_Ns_full,I__ph = 7*800*I__r(n)
  /1000, Pow = 0.3*Pow(n),x2y, eqns),set),subs( x2y, {I__pv(t),
  V__pv(t)}));
  end proc:
> results := [seq([n,solpow(n)], n = 0..24)]:
> for i from 1 to 24 do
    results[i][2]:
    I__pv(i) := subs(%,I__pv):
    V__pv(i) := subs(%%,V__pv):
  end do:
> with(plots):
  pointplot([seq([r, V__pv(r)], r = 1..24)], style=line, color=
  blue, symbolsize=20,view=[default, 0..20],title = "Voltage",
  labels = ["Time (h)", "Vpv (V)"]);
```




```
> with(plots):
  pointplot([seq([r, I__pv(r)], r = 1..24)], style=line, color=
  blue, symbolsize=20,view=[default, 0..12],title = "Current",
  labels = ["Time (h)", "Ipv (V)"]);
```



2) Simulation with external load and battery

Simulation with external load and battery connected (current constant)

In the simplified battery model provided in the paper: "Hybrid vehicle optimisation: lead acid battery modelling" the current I0 is assumed to be constant for this reason in the next section the equations for the panel connected to the battery and a load are solved assuming the battery current constant, specifically I0 = 8

```
> Ns := 100:
```

```
Tf := 24:

T__VEC := [seq(Tf/Ns*i, i=0..Ns)]:

equations__prova := {

            V0__func(t,I__0)*(I__0+Iout(V0__func(t,I__0)))=
Pow__L

             };


# Define the variable set for the solver
vars2:= {Pow__L}:
```

$$equations_{prova} := \Bigg\{ \frac{1}{1000.000000 + 1000.000000\, \mathrm{e}^{3.72444321\, I_0}} \Bigg(\Big(12623. + I_0 \Big($$

$$-9718.07\, \mathrm{e}^{-0.00002247241511\, t + 3.72444321\, I_0} - 9718.07\, \mathrm{e}^{-0.00002247241511\, t}$$

$$+ 9718.07\, \mathrm{e}^{3.72444321\, I_0} + 9718.07 \Big) \ln\Big(1. - 0.001694266338\, I_0\, t \tag{5.1.1}$$

$$- 0.1437185100\, I_0\, t \max\left(0., I_0 \left(1. - 1.\, e^{-0.00002247241511\, t}\right)\right)^{0.64741}\Big) + \Big(12623.$$

$$- 0.05413451899\, I_0^2\, t + (-92.06069999 - 314.8790363\, t)\, I_0\Big)\, e^{3.72444321\, I_0}$$

$$- 0.05413451899\, I_0^2\, t + (-1984.39563 - 314.8790363\, t)\, I_0\Big) \Bigg(I_0$$

$$+ \frac{1}{(Rh + Rs)\, Rs} \Bigg(-(Rh$$

$$+ Rs)$$

$$\eta\, V_{th}\, \mathrm{LambertW}\Bigg(\frac{1}{(Rh + Rs)\, \eta\, V_{th}} \Bigg(I_{ds}\, Rs\, Rh$$

$$e^{\frac{1}{(Rh+Rs)\,\eta\, V_{th}} \left(\left(I_{ds}\, Rs + I_{ph}\, Rs + \frac{1}{1000.000000 + 1000.000000\, e^{3.72444321\, I_0}} \left(12623.\right.\right.\right.}$$

$$+ I_0 \big(-9718.07\, \mathrm{e}^{-0.00002247241511\, t + 3.72444321\, I_0} - 9718.07\, \mathrm{e}^{-0.00002247241511\, t} + 9718.07\, \mathrm{e}^{3.72444321\, I_0}$$

$$+ 9718.07 \big) \ln\big(1. - 0.001694266338\, I_0\, t - 0.1437185100\, I_0\, t \max\big(0., I_0\, (1. - 1.\, \mathrm{e}^{-0.00002247241511\, t})\big)^{0.64741} \big)$$

$$+ \big(12623. - 0.05413451899\, I_0^{\,2}\, t + (-92.06069999 - 314.8790363\, t)\, I_0 \big)\, \mathrm{e}^{3.72444321\, I_0} - 0.05413451899\, I_0^{\,2}\, t + ($$

$$-1984.39563 - 314.8790363\, t)\, I_0 \big) \big) Rh \big) \big) \big) + Rs \Big(I_{ds}\, Rh + I_{ph}\, Rh$$

$$- \frac{1}{1000.000000 + 1000.000000\, \mathrm{e}^{3.72444321\, I_0}} \Big(12623. + I_0 \Big($$

$$-9718.07\, \mathrm{e}^{-0.00002247241511\, t + 3.72444321\, I_0} - 9718.07\, \mathrm{e}^{-0.00002247241511\, t}$$

$$+ 9718.07\, \mathrm{e}^{3.72444321\, I_0} + 9718.07 \Big) \ln\Big(1. - 0.001694266338\, I_0\, t$$

$$- 0.1437185100\, I_0\, t \max\Big(0., I_0\, (1. - 1.\, \mathrm{e}^{-0.00002247241511\, t})\Big)^{0.64741} \Big) + \Big(12623.$$

$$- 0.05413451899\, I_0^{\,2}\, t + (-92.06069999 - 314.8790363\, t)\, I_0 \Big)\, \mathrm{e}^{3.72444321\, I_0}$$

$$- 0.05413451899\, I_0^{\,2}\, t + (-1984.39563 - 314.8790363\, t)\, I_0 \Big) \Big) \Big) \Big) \Big) = Pow_L \Big\}$$

```
> tmp := fsolve(subs(data_Ns_full, t=T__VEC[1], I__ph = 7*800*
  I__r(0)/1000,I__0=8,equations__prova),vars2):
  sol := Matrix(1..Ns, 1..6, fill=0): # creates a matrix full of
  zeroes
```

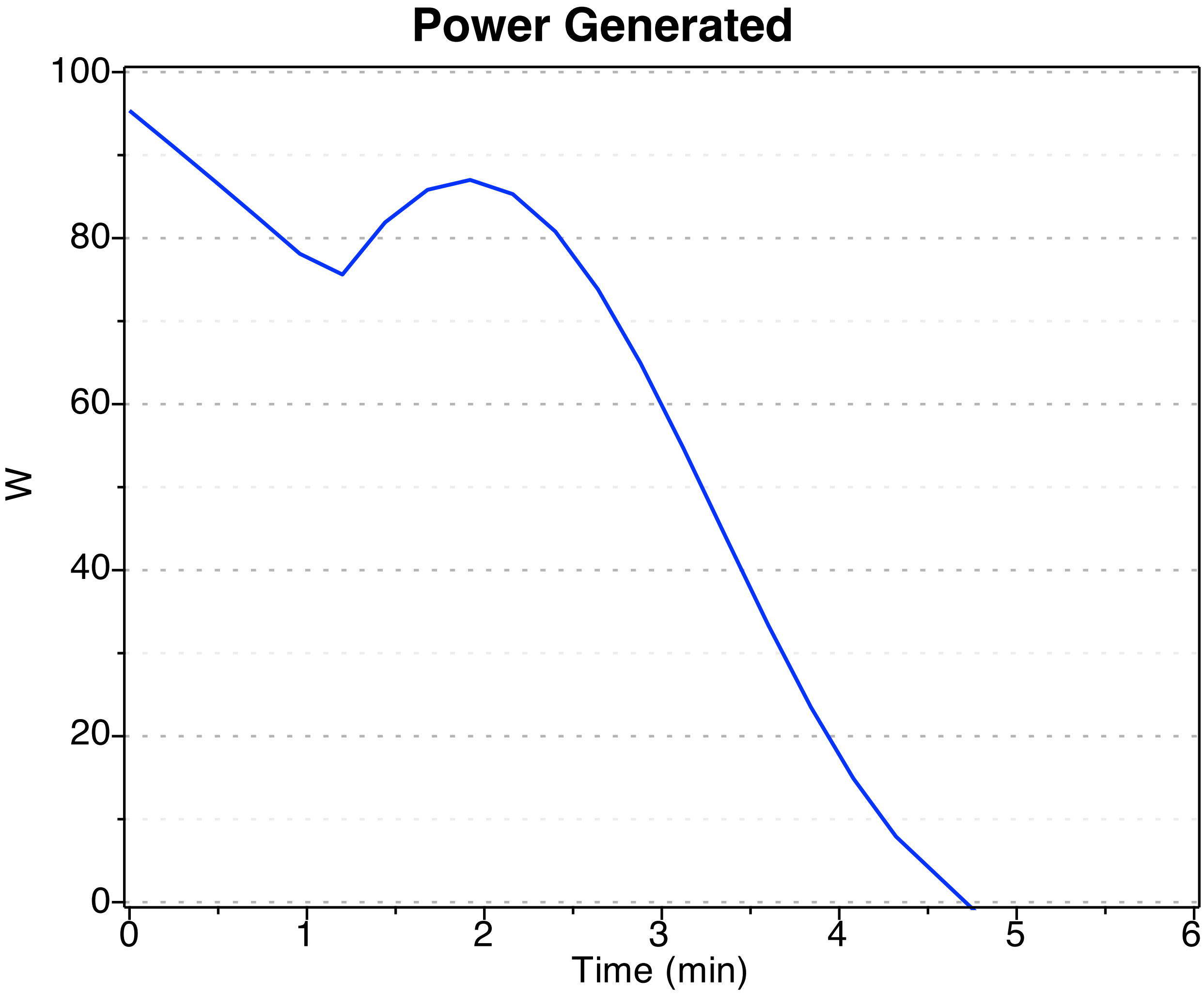
```
> for k from 1 to Ns do

  tmp := fsolve(subs(data_Ns_full, t=T__VEC[k],I__ph = 7*800*
  I__r(k)/1000,I__0=8,equations__prova),tmp):
    sol[k,1..6] := (subs(t=T__VEC[k], tmp, <t,Pow__L,V0__func
  (T__VEC[k],8),subs(data_Ns_full, t=T__VEC[k], I__ph = 7*800*
  I__r(k)/1000,I__0=8,Iout(V0__func(t,I__0)))>)):
  end:
  #print(sol):
```

```
> pointp1 := pointplot(sol[1..-9,1],sol[1..-9,2],connect=true,
```

```
color=blue, symbolsize=20,view=[0..6,0..100 ],title = "Power
Generated", labels = ["Time (min)","W"]):
pointp2 := pointplot(sol[1..-1,1],sol[1..-1,4],connect=true,
color=blue, symbolsize=20,view=[0..6,0..10 ],title = "Current
(Output Solar Panel)", labels = ["Time (min)","A"]):
pointp3 := semilogplot(sol[1..30,1],sol[1..30,3],color=blue,
symbolsize=20,view=[default, default],title = "Voltage
(Battery)", labels = ["Time-log (min)","V"]):
```

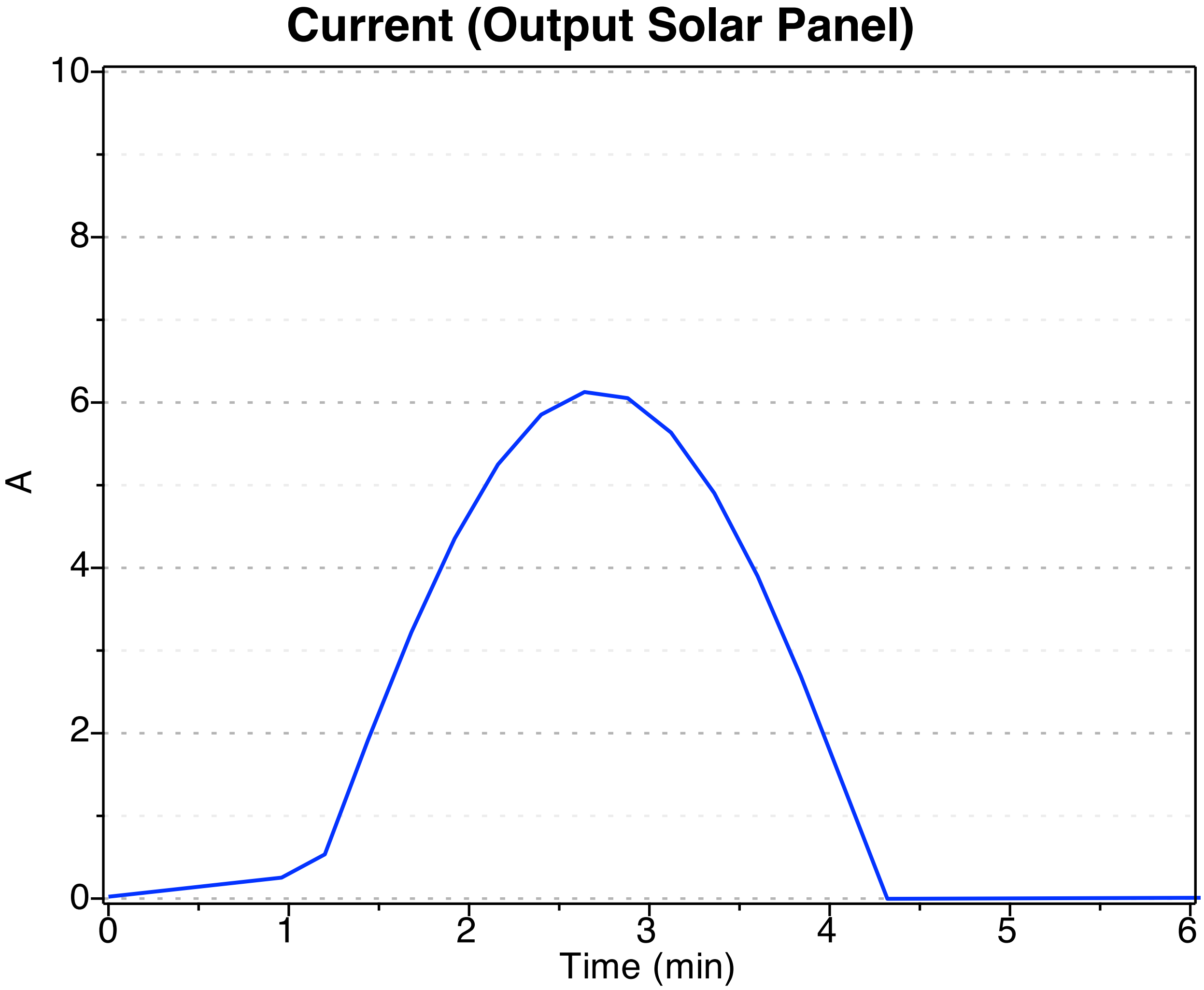
This graph shows the power generated over time in a system where a battery is connected to an external load. It is evident that the power consistently decreases as the battery discharges. Additionally, there are localized increases in power when the current generated by the solar panel rises, reflecting the effects of increased solar irradiance. In this part of the simulation, time is modeled in minutes instead of hours to clearly demonstrate the impact of the solar panel on the battery, which discharges within minutes.

```
> display(pointp1,size = [600, 400]);
```



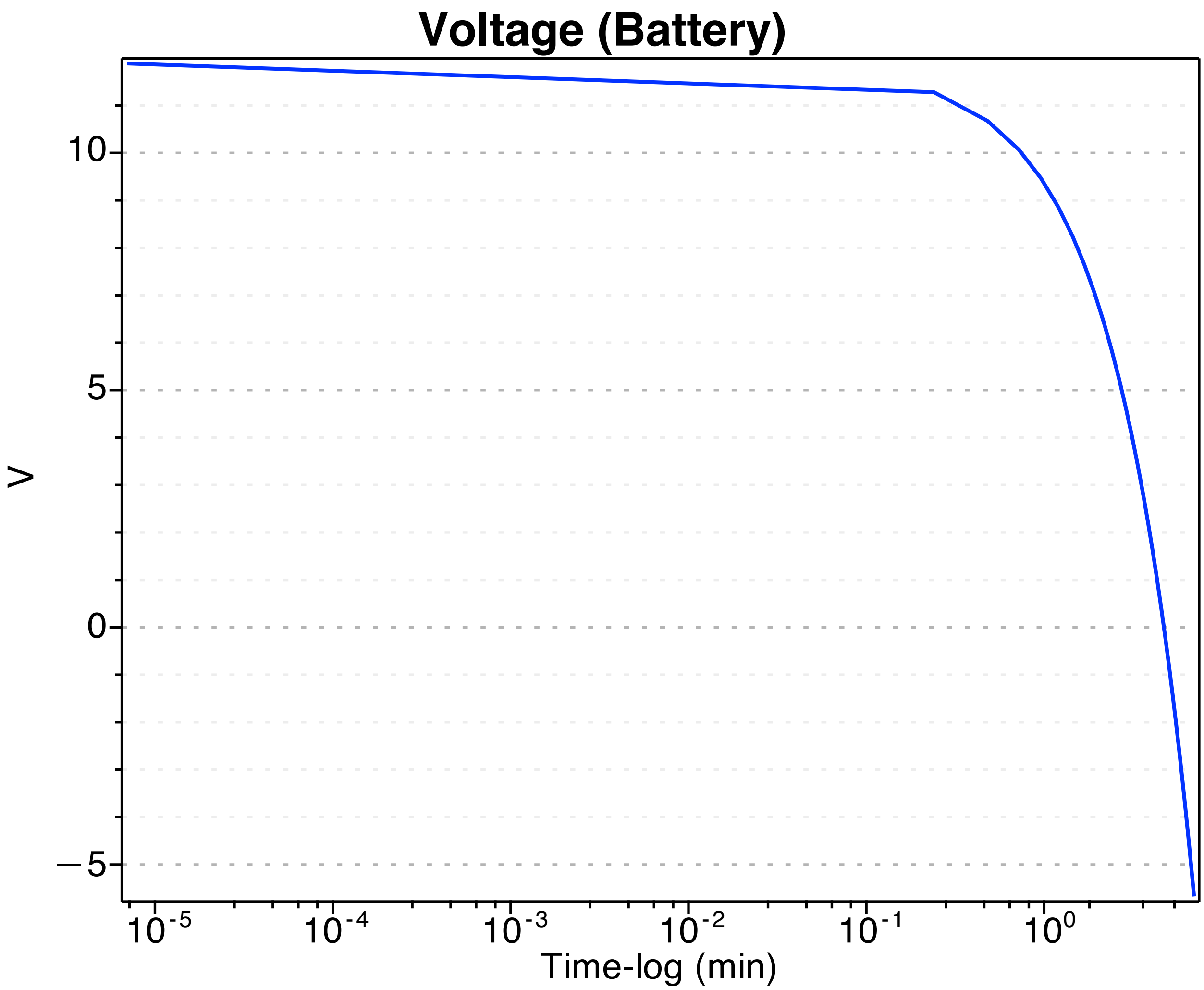
Output current of the solar panel considering the irradiance modelled in minutes instead of hours.
This is done to illustrate the effect of the effect of the solar panel and the Ceraolo battery at the same time connected togheter

```
> display(pointp2,size = [600, 400]);
```



This graph shows the Ceraolo battery model discharge voltage over time, with time plotted on a logarithmic scale

```
> display(pointp3,size = [600, 400]);
```



Simulation with constant external load and battery connected

Equations of pannel connected to a constant load with the procedure to solve the system in the different hours during the day.

In this simulation, it is explored the behaviour of a photovoltaic (PV) panel system connected to a constant load over a 24-hour period. The purpose of this study is to understand how different factors such as irradiance and load demand affect the performance of the PV panel and the connected battery system throughout the day.

```
> Ns := 100:
  Tf := 24:

  T__VEC := [seq(Tf/Ns*i, i=0..Ns)]:
```

```
equations__2 := subs(I__pv(t) = I_pv,eval({
               V0__func(t*60,I__0)-V__L,  # times 60 to
transform the battery voltage low in hours
             Vout(I__pv(t))-V__L,
             V__L*(I__0+I__pv(t))=Pow__L

              }));


# Define the variable set for the solver
vars2:= {I__0,I_pv,V__L}:
```

$$equations_2 := \left\{ \frac{1}{1000.000000 + 1000.000000\, e^{3.72444321\, I_0}} \Big(12623. + I_0 \Big(\tag{5.2.1}$$

$$-9718.07\, e^{-0.001348344907\, t + 3.72444321\, I_0} - 9718.07\, e^{-0.001348344907\, t}$$

$$+ 9718.07\, e^{3.72444321\, I_0} + 9718.07 \Big) \ln\Big(1. - 0.1016559803\, I_0\, t$$

$$- 8.623110600\, I_0\, t \max\left(0., I_0 \left(1. - 1.\, e^{-0.001348344907\, t}\right)\right)^{0.64741}\Big) + \Big(12623.$$

$$- 3.248071139\, I_0^2\, t + \left(-92.06069999 - 18892.74218\, t\right) I_0\Big)\, e^{3.72444321\, I_0}$$

$$- 3.248071139\, I_0^2\, t + \left(-1984.39563 - 18892.74218\, t\right) I_0\Big) - V_L,$$

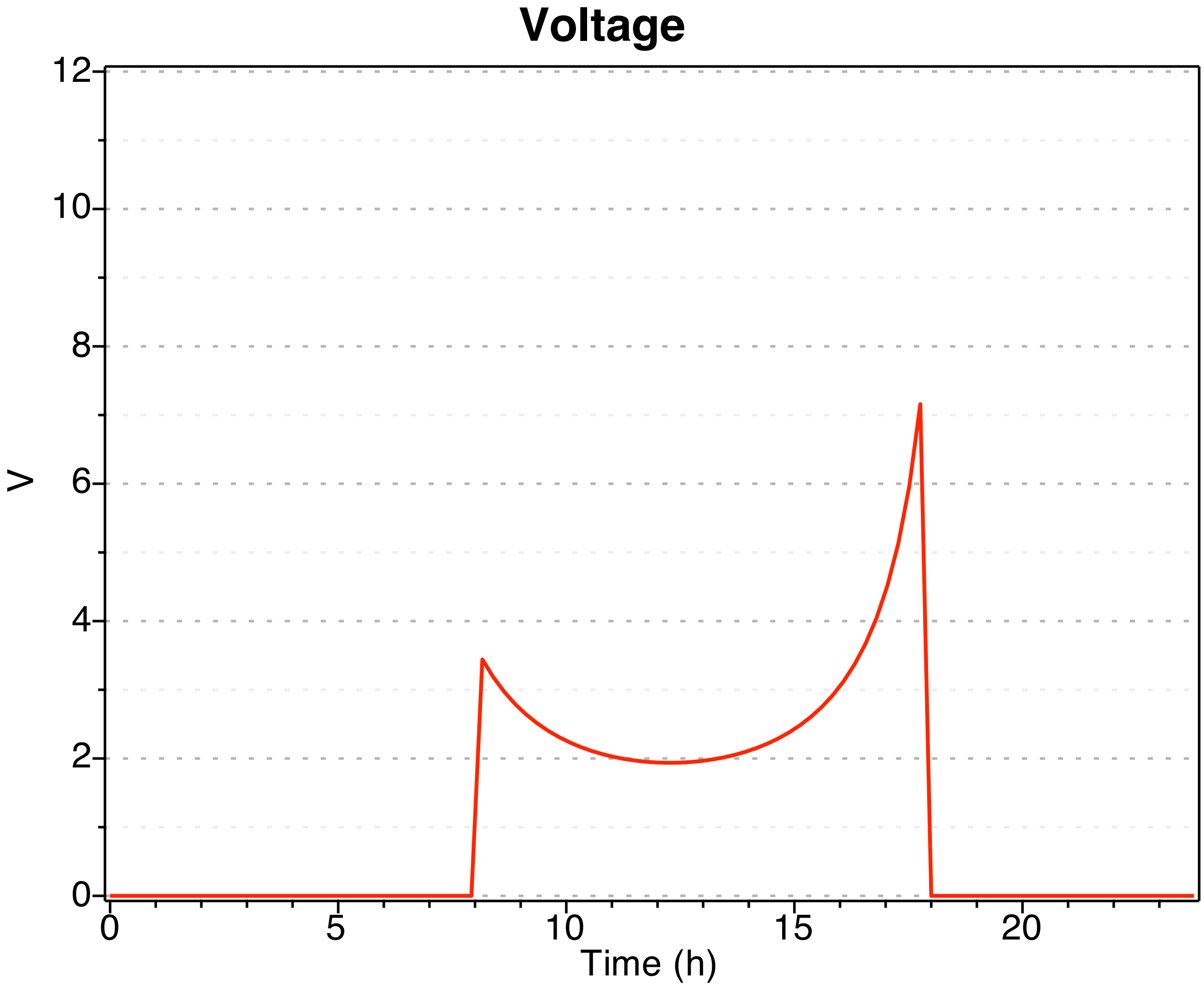
$$-\mathrm{LambertW}\left(\frac{I_{ds}\, Rh\, e^{\frac{Rh\left(I_{ds} - I\_pv + I_{ph}\right)}{\eta\, V_{th}}}}{\eta\, V_{th}} \right) \eta\, V_{th} + \left(-Rh - Rs\right) I\_pv + \Big(I_{ds}$$

$$+ I_{ph}) Rh - V_L, V_L (I_0 + I\_pv) = Pow_L \}$$

```
> tmp := fsolve(subs(data_Ns_full,t=T__VEC[1],I__ph = 7*800*I__r
(0)/1000,Pow__L=0.3*Pow__cons(0),equations__2),vars2);
```

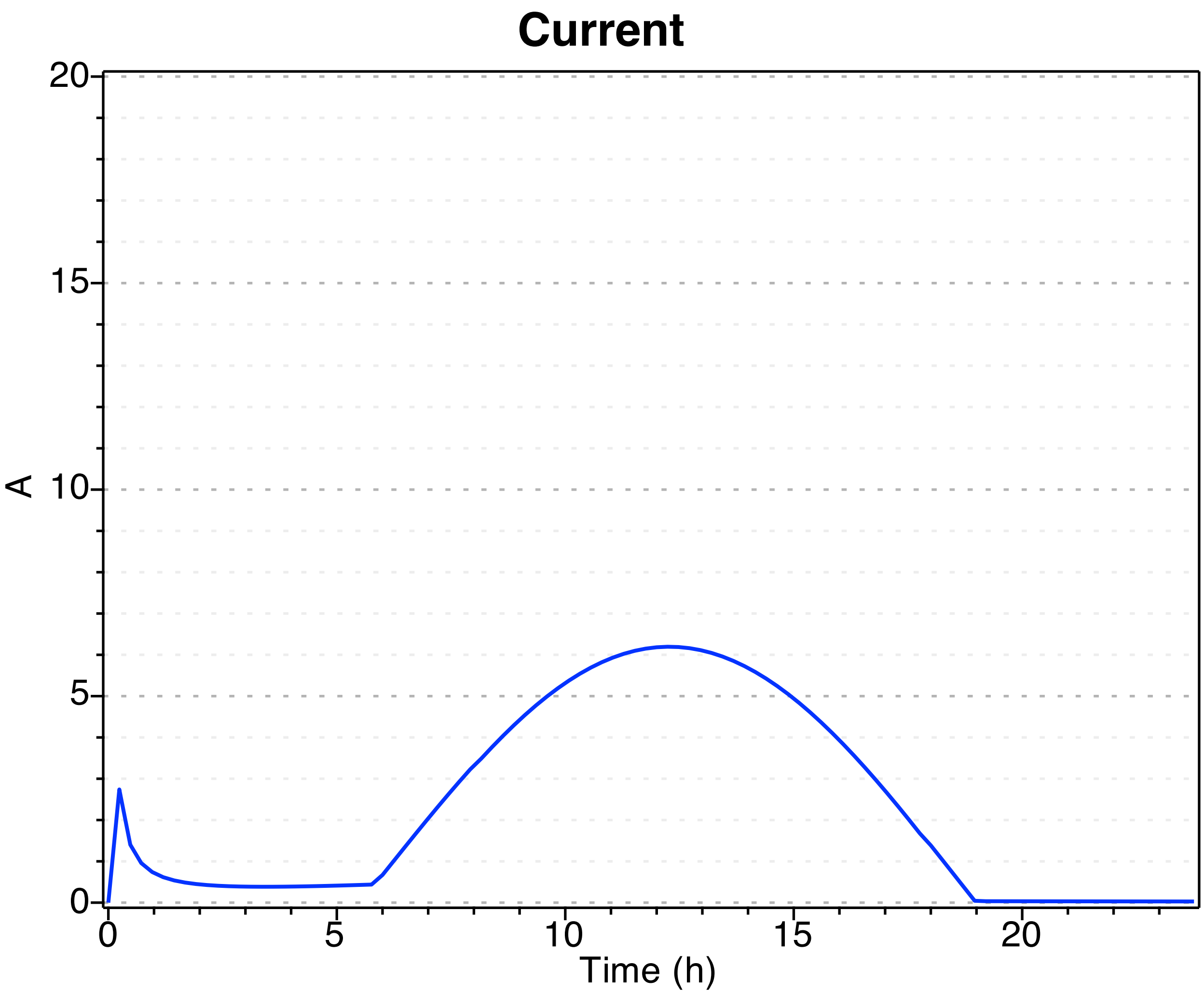
$$tmp := \{I_0 = 137.1160550, I\_pv = 1.663567043 \times 10^{-19}, V_L = 0.\} \quad \textbf{(5.2.2)}$$

```
> subs(data_Ns_full,I_ph = 7*800*I__r(0)/1000, t=T__VEC[1],
equations__2):

sol := Matrix(1..Ns, 1..5, fill=0): # creates a matrix full of
zeroes

for k from 2 to Ns do

tmp := fsolve(subs(data_Ns_full, t=T__VEC[k],I__ph = 7*800*
I__r(T__VEC[k])/1000,Pow__L=0.3*Pow__cons(T__VEC[k]),
equations__2),tmp):
  sol[k,1..5] := (subs(t=T__VEC[k], tmp, <t,I__0+I_pv,V__L,
(I_pv+I__0)*V__L>)):

end:
> sol:
pointp1 := pointplot(sol[1..-1,1],sol[1..-1,3],connect=true,
color=red, symbolsize=20,view=[default, 0..12],title =
"Voltage", labels = ["Time (h)", "V"]):
pointp2 := pointplot(sol[1..-1,1],sol[1..-1,2],connect=true,
color=blue, symbolsize=20,view=[default, 0..20],title =
"Current", labels = ["Time (h)","A"]):


display(pointp1,size = [600, 400]);
```



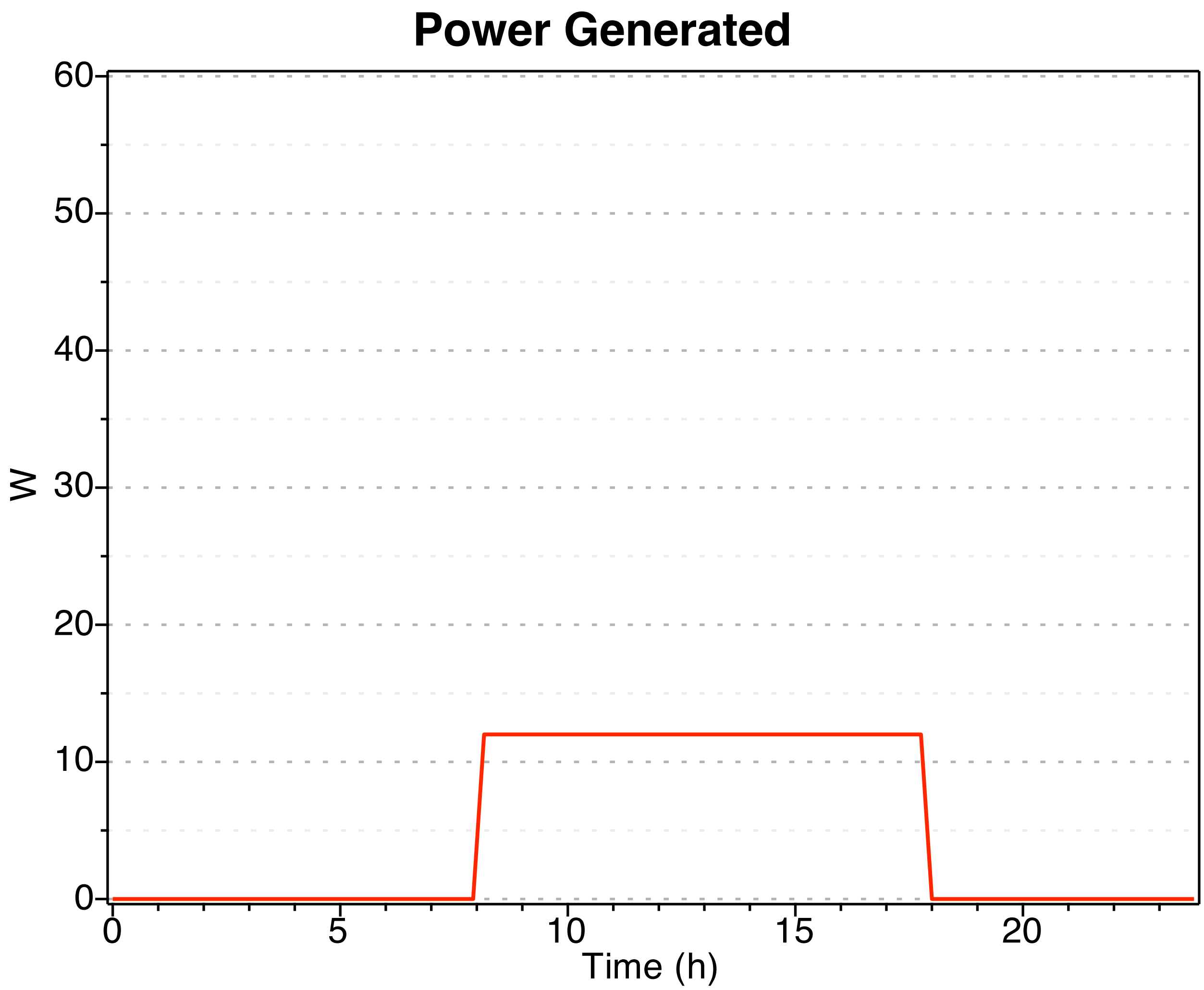
In the following plot it is possible to see a spike in the current due to the battery connected to model

```
> display(pointp2,size = [600, 400]);
```



In the following plot it is display the Power generated by the solar panel (multiplying the voltage times the current) obtaining the same the Load requested defined in the piecewise function "Pow__cons"

```
> pointp3 := pointplot(sol[1..-1,1],sol[1..-1,4],connect=true,
  color=red, symbolsize=20,view=[default, 0..60],title = "Power
  Generated", labels = ["Time (h)", "W"]):

  #line3 := plot(Pow_max, x = 0..24, color = "blue", linestyle =
  dash, legend = "Max value with 100%power"):

  display(pointp3,size = [600, 400]);
```



Simulation with various time law of the external load and battery connected

Equations of pannel connected to a time varying load with the procedure to solve the system in the different hours during the day

In this simulation, it is explored the behaviour of a photovoltaic (PV) panel system connected to a time-varying load over a 24-hour period. The purpose of this study is to understand how different factors such as irradiance and load demand affect the performance of the PV panel and the connected battery system throughout the day.

```
> Ns := 100:
  Tf := 24:

  T__VEC := [seq(Tf/Ns*i, i=0..Ns)]:

  equations__2 := subs(I__pv(t) = I_pv,eval({
```

```
            V0__func(t*60,I__0)-V__L,
           Vout(I__pv(t))-V__L,
           V__L*(I__0+I__pv(t))=Pow__L

            }));


# Define the variable set for the solver
vars2:= {I__0,I_pv,V__L}:
```

$$equations_2 := \Bigg\{ \frac{1}{1000.000000 + 1000.000000\, e^{3.72444321\, I_0}} \Big(12623. + I_0 \Big(\tag{5.3.1}$$

$$-9718.07\, e^{-0.001348344907\, t + 3.72444321\, I_0} - 9718.07\, e^{-0.001348344907\, t}$$

$$+ 9718.07\, e^{3.72444321\, I_0} + 9718.07 \Big) \ln\Big(1. - 0.1016559803\, I_0\, t$$

$$- 8.623110600\, I_0\, t \max\Big(0., I_0 \left(1. - 1.\, e^{-0.001348344907\, t} \right) \Big)^{0.64741} \Big) + \Big(12623.$$

$$- 3.248071139\, I_0^2\, t + (-92.06069999 - 18892.74218\, t)\, I_0 \Big)\, e^{3.72444321\, I_0}$$

$$- 3.248071139\, I_0^2\, t + (-1984.39563 - 18892.74218\, t)\, I_0 \Big) - V_L,$$

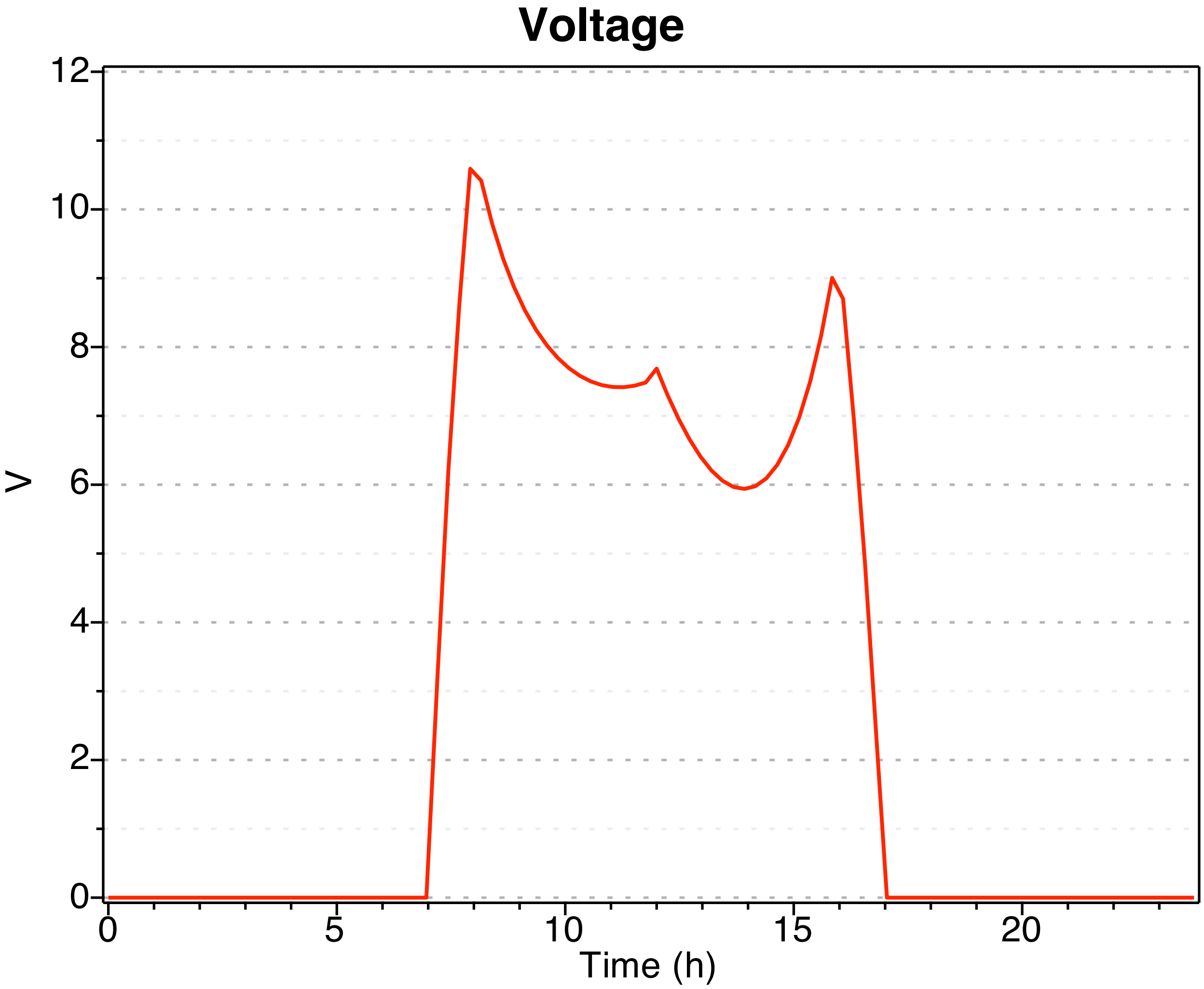
$$-\mathrm{LambertW}\left(\frac{I_{ds}\, Rh\, e^{\frac{Rh\, \left(I_{ds} - I\_pv + I_{ph} \right)}{\eta\, V_{th}}}}{\eta\, V_{th}} \right) \eta\, V_{th} + (-Rh - Rs)\, I\_pv + \Big(I_{ds}$$

$$+ I_{ph}) Rh - V_L, V_L (I_0 + I\_pv) = Pow_L \}$$

```
> tmp := fsolve(subs(data_Ns_full, t=T__VEC[1], I__ph = 7*800*
  I__r(0)/1000,Pow__L=Pow(0),equations__2),vars2);
```

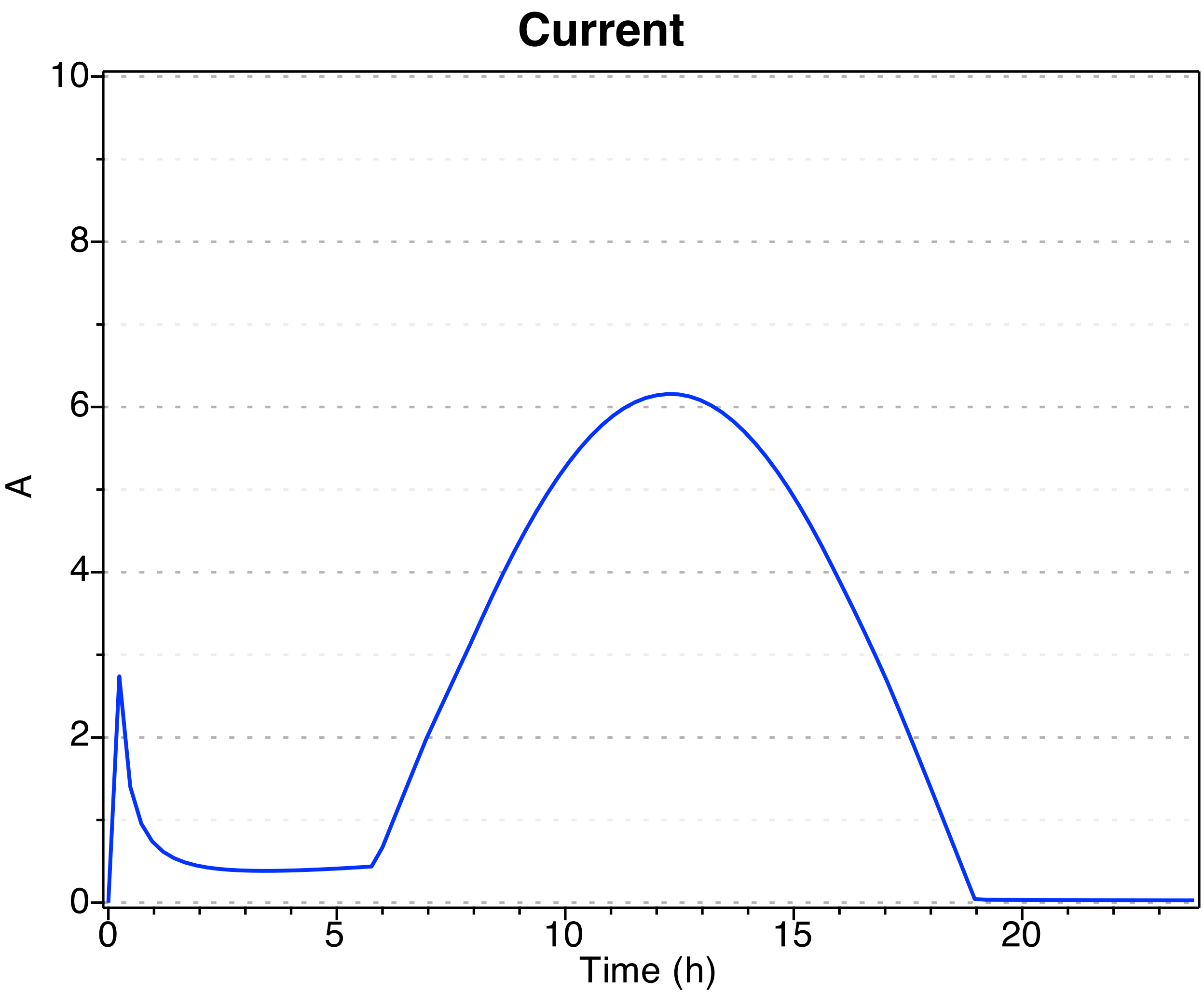
$$tmp := \{I_0 = 137.1160550, I\_pv = 1.663567043 \times 10^{-19}, V_L = 0.\} \tag{5.3.2}$$

```
> subs(data_Ns_full,I_ph = 7*800*I__r(0)/1000, t=T__VEC[1],
  equations__2):

  sol := Matrix(1..Ns, 1..5, fill=0): # creates a matrix full of
  zeroes

  for k from 2 to Ns do

  tmp := fsolve(subs(data_Ns_full, t=T__VEC[k],I__ph = 7*800*
  I__r(T__VEC[k])/1000,Pow__L=Pow(T__VEC[k]),equations__2),tmp):
    sol[k,1..5] := (subs(t=T__VEC[k], tmp, <t,I__0+I_pv,V__L,
  (I_pv+I__0)*V__L>)):

  end:
> sol:
> pointp1 := pointplot(sol[1..-1,1],sol[1..-1,3],connect=true,
  color=red, symbolsize=20,view=[default, 0..12],title =
  "Voltage", labels = ["Time (h)", "V"]):
  pointp2 := pointplot(sol[1..-1,1],sol[1..-1,2],connect=true,
  color=blue, symbolsize=20,view=[default, 0..10],title =
  "Current", labels = ["Time (h)","A"]):


  display(pointp1,size = [600, 400]);
```



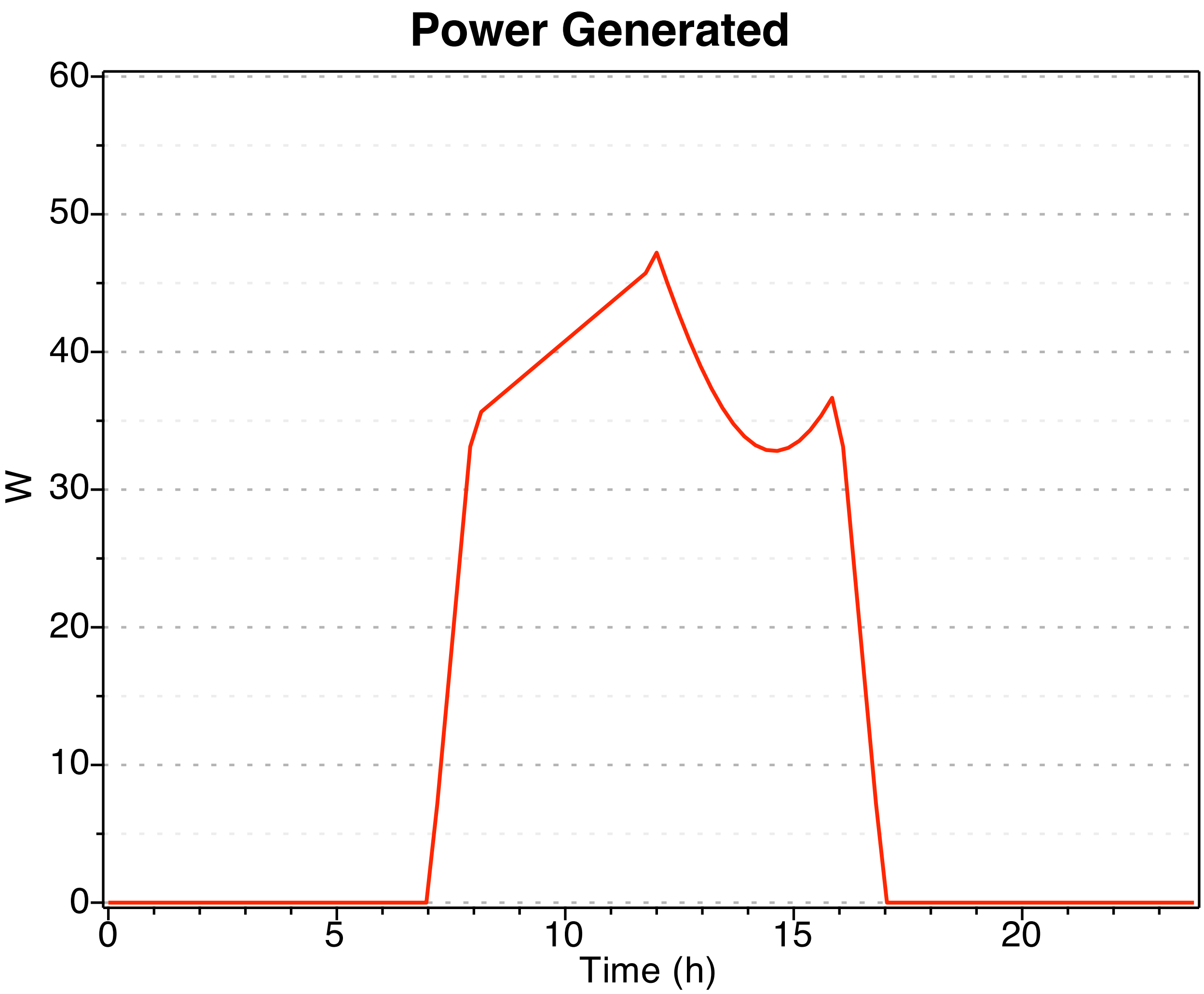
In the following plot it is possible to see a spike in the current due to the battery connected to model

```
> display(pointp2,size = [600, 400]);
```



In the following plot it is display the Power generated by the solar panel (multiplying the voltage times the current) obtaining the same the Load requested defined in the piecewise function "Pow"

```
> pointp3 := pointplot(sol[1..-1,1],sol[1..-1,4],connect=true,
  color=red, symbolsize=20,view=[default, 0..60],title = "Power
  Generated", labels = ["Time (h)", "W"]):

  #line3 := plot(Pow_max, x = 0..24, color = "blue", linestyle =
  dash, legend = "Max value with 100%power"):

  display(pointp3,size = [600, 400]);
```


Power Generated
W
Time (h)
60
50
40
30
20
10
0
0
5
10
15
20