# Analisi di Fourier

Marcello Colozzo - http://www.extrabyte.info

## Dichiarazione di funzioni periodiche

In[1]:=
```
SetOptions[
  {
   Plot,
   ListLinePlot, ParametricPlot, ListPlot,
   ListLinePlot
  },
  AxesStyle -> Directive[
    Hue[5 / 6],
    8
   ],
  FrameStyle -> Directive[
    Hue[5 / 6],
    8
   ]
 ];
```

Se non ci sono molti intervalli di periodicità, si può procedere per traslazione del grafico. Ad esempio, consideriamo uno smorzamento esponenziale con costante di tempo $\tau$: $y(t) = A\, e^{-t/\tau}$ assumendo che sia periodico con periodo assegnato $T$.

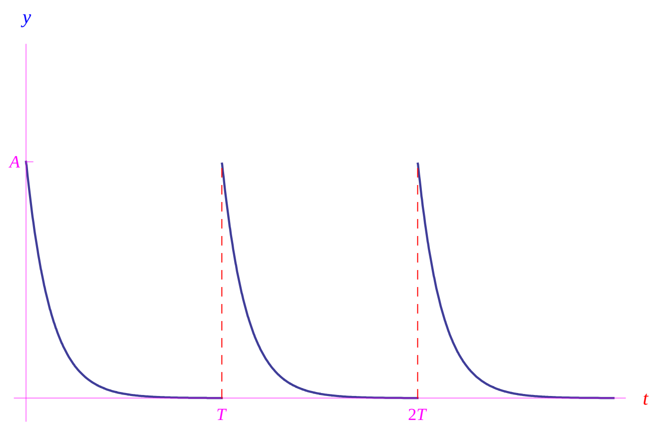
In[2]:=
```
A = 10.; τ = 1; T = 8;
```

In[3]:=
```
Clear[y]
```

In[4]:=
```
y[t_] := Which[
  t ≥ 0 && t ≤ T, A * e^(-t/τ),
  t ≥ T && t ≤ 2 T, A * e^(-(t-T)/τ),
  t ≥ 2 T && t ≤ 3 T, A * e^(-(t-2 T)/τ)
 ]
```

In[5]:=
```
ploty = Plot[
  y[t],
  {t, 0, 3 T},
  Exclusions → {t == T, t == 2 T},
  PlotStyle → Thickness[0.0035],
  PlotRange → {-1, 15},
  AxesLabel →
   {
    Style["t", Small, Red, Italic],
    Style["y", Small, Blue, Italic]
   },
  Epilog → {
    Dashed,
    Red,
    Line[{{T, 0}, {T, 10}}],
    Line[{{2 T, 0}, {2 T, 10}}]
   },
  Ticks →
   {
    {
     {T, "T"}, {2 T, "2T"}
    },
    {
     {A, "A"}
    }
   }
 ]
```

Out[5]=




### Serie di Fourier

*Mathematica* dispone di vari package per l'analisi di Fourier. Tuttavia per funzioni definite con l'assegnazione condizionata, è necessario prestare attenzione. Infatti, un utilizzazione cieca di tali strumenti, conduce a risultati disastrosi. Ad esempio:

In[6]:=
```
Needs["FourierSeries`"]
```

In[7]:=
```
yfourier[t_] = FourierTrigSeries[
  (*funzione di cui si cerca la serie di Fourier*)
  y[t],
  (*variabile indipendente*)
  t
 ]
```

Out[7]= FourierTrigSeries$\left[\text{Which}\left[t \geq 0 \,\&\&\, t \leq 8,\ A\, e^{-\frac{t}{\tau}},\ t \geq T \,\&\&\, t \leq 2\,T,\ A\, e^{-\frac{t-T}{\tau}},\ t \geq 2\,T \,\&\&\, t \leq 3\,T,\ A\, e^{-\frac{t-2\,T}{\tau}}\right],\ t\right]$

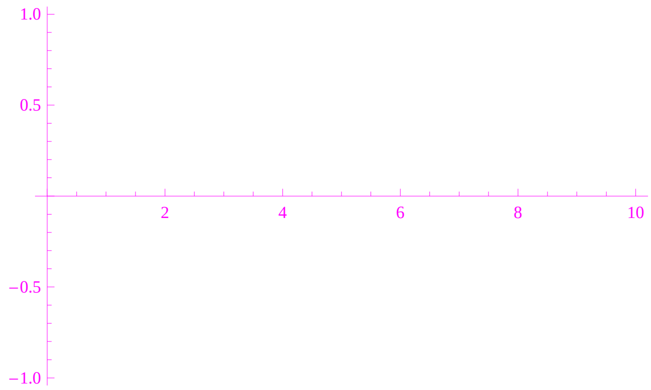
In[8]:= **yfourier[2.]**

Out[8]= FourierTrigSeries[1.35335, 2.]

Se proviamo a plottare:

In[9]:=
```
Plot[
 yfourier[t],
 {t, 0, 10}
]
```

Out[9]=




Nel caso di assegnazione condizionata, conviene determinare i coefficienti di Fourier (consultare la dispensa http://www.extrabyte.info/serie_fourier000.pdf):

In[11]:=
$$\texttt{a[k\_] := }\frac{2}{T}\,\texttt{Integrate}\Big[\;\texttt{y[t] Cos}\Big[\frac{2*k*\pi*t}{T}\Big],\;\{t,0,T\},\;\texttt{Assumptions}\to k\in\texttt{Integers}\;\Big]$$

In[12]:=
$$\texttt{b[k\_] := }\frac{2}{T}\,\texttt{Integrate}\Big[\;\texttt{y[t] Sin}\Big[\frac{2*k*\pi*t}{T}\Big],\;\{t,0,T\},\;\texttt{Assumptions}\to k\in\texttt{Integers}\;\Big]$$

In[13]:=
$$\texttt{yfourier[t\_, n\_] := }\frac{a[0]}{2}+\texttt{Sum}\Big[a[k]\,\texttt{Cos}\Big[\frac{2*k*\pi*t}{T}\Big]+b[k]\,\texttt{Sin}\Big[\frac{2*k*\pi*t}{T}\Big],\;\{k,1,n\}\Big];$$

In[14]:= **yfourier[t, 4]**

Out[14]=
$$1.24958+1.5457\,\mathrm{Cos}\Big[\frac{\pi t}{4}\Big]+0.720759\,\mathrm{Cos}\Big[\frac{\pi t}{2}\Big]+0.381455\,\mathrm{Cos}\Big[\frac{3\pi t}{4}\Big]+0.229922\,\mathrm{Cos}[\pi t]+$$
$$1.21399\,\mathrm{Sin}\Big[\frac{\pi t}{4}\Big]+1.13217\,\mathrm{Sin}\Big[\frac{\pi t}{2}\Big]+0.898782\,\mathrm{Sin}\Big[\frac{3\pi t}{4}\Big]+0.722321\,\mathrm{Sin}[\pi t]$$

In[15]:= **yfourier[1, 4]**

Out[15]= 4.46902

Sembra tutto ok, ma se plottiamo:

In[16]:=
```
Plot[
 yfourier[t, 1],
 {t, 1, 20}
]
```

Integrate::ilim : Invalid integration variable or limit(s) in {1.00039, 0, 8}. ≫

Integrate::ilim : Invalid integration variable or limit(s) in {1.00039, 0, 8}. ≫

Integrate::ilim : Invalid integration variable or limit(s) in {1.00039, 0, 8}. ≫

General::stop : Further output of Integrate::ilim will be suppressed during this calculation. ≫

NIntegrate::itraw : Raw object 1.0003881428571428` cannot be used as an iterator. ≫

NIntegrate::itraw : Raw object 1.0003881428571428` cannot be used as an iterator. ≫

NIntegrate::itraw : Raw object 1.0003881428571428` cannot be used as an iterator. ≫

General::stop : Further output of NIntegrate::itraw will be suppressed during this calculation. ≫

Out[16]= `$Aborted`

Per evitare questo warning, è preferibile specificare l'ordine di approssimazione e senza utilizzare l'assegnazione ritardata.

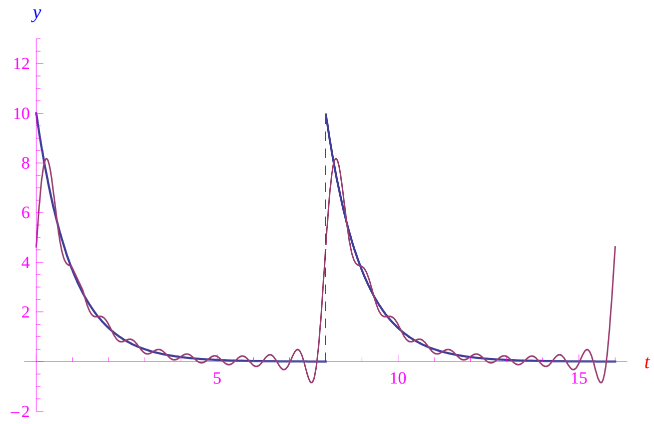
In[17]:=
```
Clear[yfourier]
```

In[18]:=
```
yfourier[t_] = a[0]/2 + Sum[a[k] Cos[(2*k*π*t)/T] + b[k] Sin[(2*k*π*t)/T], {k, 1, 10}];
```

$$\text{yfourier}[t\_] = \frac{a[0]}{2} + \text{Sum}\left[a[k]\,\text{Cos}\left[\frac{2*k*\pi*t}{T}\right] + b[k]\,\text{Sin}\left[\frac{2*k*\pi*t}{T}\right], \{k, 1, 10\}\right];$$

In[19]:=
```
Plot[
 {
  y[t],
  yfourier[t]
 },
 {t, 0, 16},
 Exclusions → {t == 8},
 Epilog → {
   Red,
   Dashed,
   Line[{{8, 0}, {8, 10}}]
  },
 PlotRange → {-2, 13},
 PlotStyle → {
   {
    Thickness[0.0035]
   },
   {
    Thickness[0.0025]
   }
  },
 AxesLabel →
  {
   Style["t", Small, Red, Italic],
   Style["y", Small, Blue, Italic]
  }
]
```

Out[19]=




Ritorniamo al plotting di funzioni periodiche.

In[20]:=
```
Clear[y, ploty]
```

Per graficare in automatico, dobbiamo creare una lista di traslazioni, tramite l'istruzione **Table**. Siccome l'istruzione **Which** non ha l'attributo **Listable** (e non c'è verso di assegnarlo, probabilmente perché è un comando protetto), utilizziamo l'istruzione **Piecewis**e che ignora un qualunque numero di parentesi annidate. Generiamo la lista:

In[21]:=
```
lista[m_] := Table[
  {A * e^(-(t-n*T)/τ), t ≥ n * T && t ≤ (n + 1) * T},
  {n, 0, m}
 ]
```

Proviamo:

In[22]:=
```
lista[4]
```

Out[22]= $\{\{10.\ e^{-t},\ t \geq 0\ \&\&\ t \leq 8\},\ \{10.\ e^{8-t},\ t \geq 8\ \&\&\ t \leq 16\},\ \{10.\ e^{16-t},\ t \geq 16\ \&\&\ t \leq 24\},\ \{10.\ e^{24-t},\ t \geq 24\ \&\&\ t \leq 32\},\ \{10.\ e^{32-t},\ t \geq 32\ \&\&\ t \leq 40\}\}$

In[23]:=
```
y[t_, m_] := Piecewise[lista[m]]
```

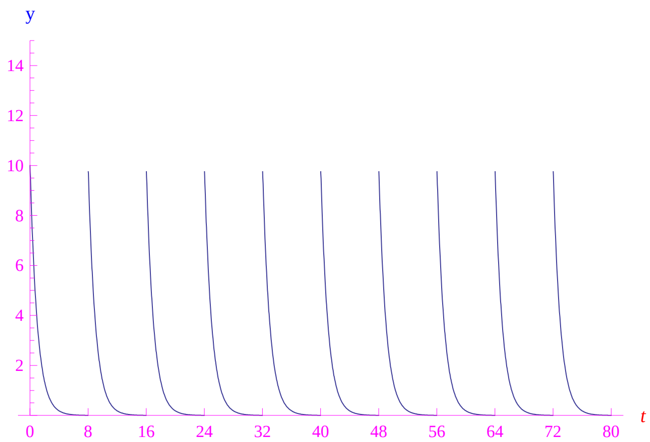
Proviamo:

In[24]:=
```
y[t, 4]
```

Out[24]=
$$\begin{cases} 10.\ e^{-t} & t \geq 0\ \&\&\ t \leq 8 \\ 10.\ e^{8-t} & t \geq 8\ \&\&\ t \leq 16 \\ 10.\ e^{16-t} & t \geq 16\ \&\&\ t \leq 24 \\ 10.\ e^{24-t} & t \geq 24\ \&\&\ t \leq 32 \\ 10.\ e^{32-t} & t \geq 32\ \&\&\ t \leq 40 \end{cases}$$

In[25]:=
```
Plot[
 y[t, 10],
 {t, 0, 80},
 PlotRange → {0, 15},
 Ticks →
  {
   Table[k, {k, 0, 80, 8}]
  },
 AxesLabel →
  {
   Style["t", Small, Red, Italic],
   Style["y", Small, Red, Blue]
  }
]
```

Out[25]=



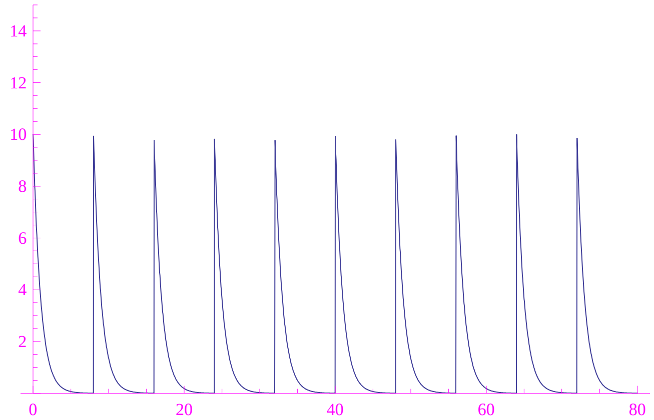

In[27]:=
```
Clear[y, ploty]
```

Conviene usare un'unica istruzione

In[28]:=
```
ploty[m_] := Plot[
  Piecewise[
   Table[
    {
     A * e^(-(t-n*T)/τ), t ≥ n * T && t ≤ (n + 1) * T
    },
    {n, 0, m}
   ]
  ],
  {t, 0, 80},
  PlotRange → {0, 15}
 ]
```

In[29]:=
```
ploty[10]
```

Out[29]=




## Onda quadra troncata

In[30]:=
```
Clear[y, ploty, lista, A, T]
```

Il tracciamento automatico di un'onda quadra (periodica) è più complicato perché la funzione da plottare ha due espressioni analitiche nello stesso intervallo di periodicità. Per fissare le idee, consideriamo $y(t) = A$, se $0 \leqslant t \leqslant \frac{T}{2}$, e $y(t) = 0$, se $\frac{T}{2} < t \leqslant T$, dove $A$ è l'ampiezza dell'onda e $T$ il periodo.

In[31]:=
```
lista[m_, A_, T_] := Flatten[
  Table[
   {{A, t ≥ n * T && t ≤ (2 n + 1) * T/2},
    {0, t ≥ (n + 1) * T/2 && t ≤ (n + 1) * T}
   },
   {n, 0, m}
  ],
  1]
```

Proviamo:

In[32]:=
```
lista[2, 1, 2]
```

Out[32]=
```
{{1, t ≥ 0 && t ≤ 1}, {0, t ≥ 1 && t ≤ 2}, {1, t ≥ 2 && t ≤ 3},
 {0, t ≥ 2 && t ≤ 4}, {1, t ≥ 4 && t ≤ 5}, {0, t ≥ 3 && t ≤ 6}}
```

In[33]:=
```
y[t_, m_, A_, T_] := Piecewise[lista[m, A, T]]
```

Proviamo

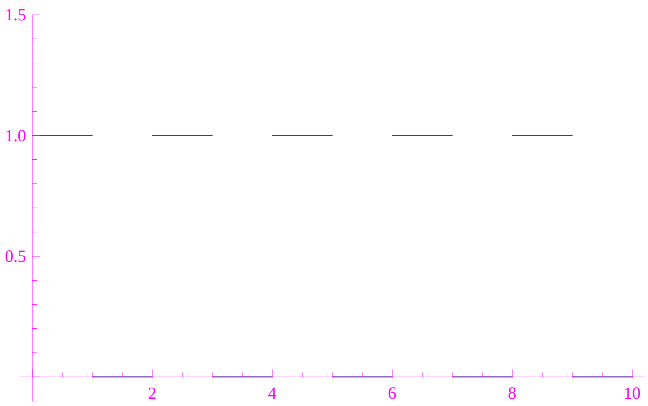
In[34]:=
```
y[t, 4, 1, 2]
```

Out[34]=

$$\begin{cases} 1 & t \geq 0 \,\&\&\, t \leq 1 \\ 0 & t \geq 1 \,\&\&\, t \leq 2 \\ 1 & t \geq 2 \,\&\&\, t \leq 3 \\ 0 & t \geq 2 \,\&\&\, t \leq 4 \\ 1 & t \geq 4 \,\&\&\, t \leq 5 \\ 0 & t \geq 3 \,\&\&\, t \leq 6 \\ 1 & t \geq 6 \,\&\&\, t \leq 7 \\ 0 & t \geq 4 \,\&\&\, t \leq 8 \\ 1 & t \geq 8 \,\&\&\, t \leq 9 \end{cases}$$

In[35]:=
```
Plot[
 y[t, 10, 1, 2],
 {t, 0, 10},
 PlotRange → {-0.1, 1.5}
]
```

Out[35]=




In[36]:=
```
Clear[y, ploty, A, T]
```

Inglobiamo in un'unica istruzione (considerando che *m* è il numero di intervalli di periodicità per un assegnato periodo *T*):

In[37]:=
```
ploty[A_, T_, m_] := Plot[
  Piecewise[
   Flatten[
    Table[
     {{A, t ≥ n*T && t ≤ (2 n + 1) * T/2},
      {0, t ≥ (n + 1) * T/2 && t ≤ (n + 1) * T}
     },
     {n, 0, m}
    ],
    1]
  ],
  {t, 0, m*T},
  PlotRange → {-0.1, 1.5*A},
  Ticks →
   {
    None,
    {
     {A, "A"}
    }
   },
  AxesLabel →
   {
    Style["t", Small, Red, Italic],
    Style["y", Small, Red, Italic]
   }
 ]
```

In[38]:=
```
ploty[1, 8, 20]
```

Out[38]=