

Equazioni Differenziali ed Equazioni alle Differenze

Lo studio delle equazioni differenziali rappresenta uno dei settori più importanti e difficili della Matematica. Non è certo questa la sede per una trattazione adeguata dell'argomento, ma un linguaggio della potenza di Mathematica permette anche ai principianti di iniziare l'esplorazione dei problemi più difficili. Per la delicatezza degli argomenti trattati si consiglia comunque la lettura di ciò che segue ad un pubblico maturo (la maturità scientifica dovrebbe bastare).

(P.S. scusate l'humour ma lo faccio per alleggerire il mattone).

di Francesco Romani

Equazioni Differenziali Ordinarie

$$y'=y$$

Diamo per scontato che tutti sappiamo che cosa è una derivata. Il primo problema che affrontiamo è la ricerca di una funzione $y(x)$ che sia uguale alla propria derivata, ovvero che soddisfi l'equazione $y'=y$ (una equazione di questo tipo si dice *Equazione Differenziale Ordinaria*). Mathematica offre la routine **DSolve** che accetta nell'ordine l'equazione, la funzione incognita e la variabile indipendente e prova a trovare una soluzione. Il risultato, se va tutto bene, è una regola di riscrittura che sostituisce a $y[x]$ la soluzione generale della equazione data.

```
In[1]:=
DSolve[y'[x] == y[x], y[x], x]
Out[1]=
```

$$\{\{y[x] \rightarrow E^x C[1]\}\}$$

Per definire la funzione soluzione usiamo il trucco visto nella puntata scorsa.

```
In[2]:=
sol[x_]:=Evaluate[(y[x]/.%)][[1]]
```

```
In[3]:=
sol[t]
```

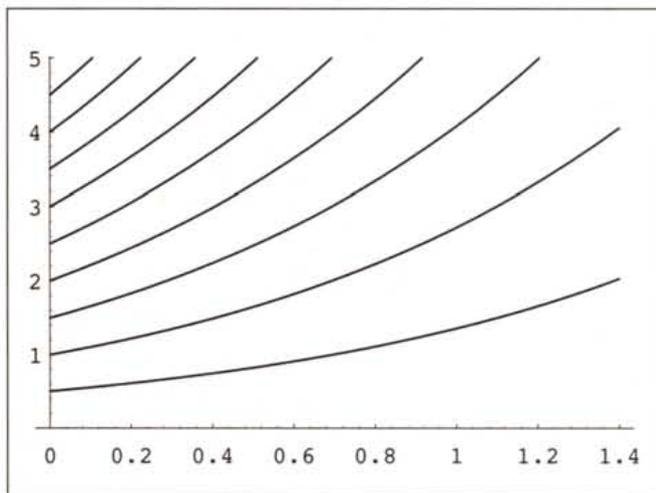
```
Out[3]=
t
E C[1]
```

Si scopre che la soluzione proposta è la funzione esponenziale e la verifica è così semplice che non c'è bisogno di farla con Mathematica. Si vede anche che la soluzione dipende da una costante indicata con $C[1]$ per ogni valore della quale si ottiene una specifica funzione che è ancora soluzione della equazione data. Abbiamo così scoperto che una equazione differenziale definisce non una funzione ma una famiglia di funzioni. Proviamo a disegnarne qualche elemento della famiglia.

```
In[4]:=
Table[sol[t]/.{C[1]->c}, {c,0,1,0.2}]
Out[4]=
```

```
{0, 0.2 E, 0.4 E, 0.6 E, 0.8 E, 1. E}
In[5]:=
Plot[Evaluate[Table[sol[t]/.
{C[1]->c}, {c,0.5,5,0.5}],
{t,0,1.4}], PlotRange->{0,5}];
```

Si ottiene una famiglia di esponenziali, (vedi Figura 1) e la costante $C[1]$ rappresenta il valore assunto dalla soluzione nel punto $x=0$.



$$y'=2y/x$$

Cambiamo equazione, stavolta cerchiamo le curve $y(x)$ tali che la derivata $y'(x)$ sia uguale a $2y(x)/x$

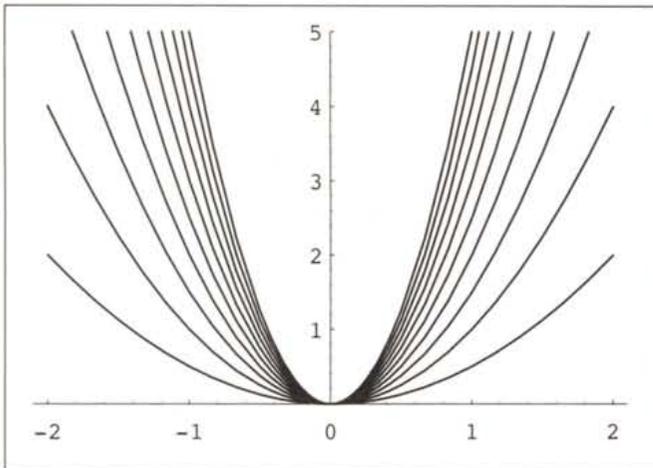
```
In[6]:=
DSolve[y'[x] == 2y[x]/x, y[x], x];
sol[x_]:=Evaluate[(y[x]/.%)][[1]]
sol[t]
```

```
Out[6]=
2
t C[1]
```

Guarda caso otteniamo una famiglia di parabole (Figura 2).

```
In[7]:=
Plot[Evaluate[Table[sol[t]/.
{C[1]->c}, {c,0.5,5,0.5}],
```

```
{t, -2, 2}, PlotRange->{0, 5}};
```



$y' = -xy$

Cambiamo ancora equazione, cerchiamo le curve $y(x)$ tali che la derivata $y'(x)$ sia uguale a $-x y(x)$

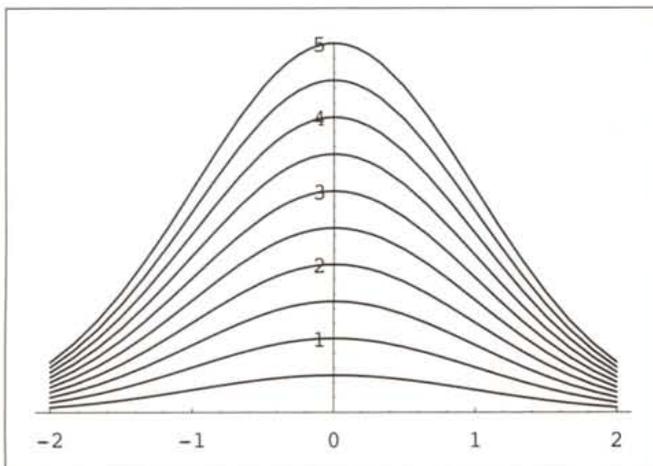
```
In[8]:= DSolve[y'[x] == -x y[x], y[x], x];
sol[x_] := Evaluate[(y[x]/.%)][[1]]
sol[t]
```

```
Out[8]= C[1]
```

```
t^2/2
E
```

```
In[9]:= Plot[Evaluate[Table[sol[t]/.
{C[1]->c}, {c, 0.5, 5, 0.5}]],
{t, -2, 2}, PlotRange->{0, 5}];
```

Otteniamo una famiglia di curve gaussiane (a campana) (Figura 3).



Secondo ordine a coefficienti costanti

Una equazione più complicata coinvolge la derivata seconda e la derivata prima: la soluzione dipende da due costanti.

```
In[10]:= DSolve[y''[x] + y[x] + 1 == 0, y[x], x]
Out[10]= {{y[x] -> -1 + C[2] Cos[x] - C[1] Sin[x]}}
```

Condizioni iniziali

Un primo modo per fissare una sola soluzione è quello di stabilire il valore iniziale (ovvero per $x=0$) della funzione e della sua derivata. **DSolve** permette di specificare i valori iniziali e automaticamente assegna i valori giusti alle costanti.

```
In[11]:= DSolve[{y''[x] + y[x] + 1 == 0,
y[0]==1,
y'[0]==0}, y[x], x]
```

```
Out[11]= {{y[x] -> -1 + 2 Cos[x]}}
```

```
In[12]:= u[x_] := Evaluate[Simplify[y[x]/.%][[1]]];
u[x]
```

```
Out[12]= -1 + 2 Cos[x]
```

Condizioni al contorno

Un secondo modo per fissare una soluzione è quello di stabilire il valore della funzione in due punti. In un certo senso questo è il problema centrale della Artiglieria: si sa dove è il cannone e dove si vuole colpire, si tratta di trovare la traiettoria giusta (*l'esempio è meno cretino di quanto sembra a prima vista perché la traiettoria di un proiettile è davvero la soluzione di una equazione differenziale*). **DSolve** permette di specificare i valori al contorno e automaticamente assegna i valori giusti alle costanti.

```
In[13]:= DSolve[{y''[x] + y[x] + 1 == 0,
y[0]==1,
y[Pi/2]==1}, y[x], x];
u[x_] := Evaluate[Simplify[y[x]/.%][[1]]];
u[x]
```

```
Out[13]= -1 + 2 Cos[x] + 2 Sin[x]
```

Un caso più generale

Quella che segue è la più generale equazione differenziale lineare del secondo ordine a coefficienti costanti.

```
In[14]:= DSolve[y''[x]+a y'[x]+b y[x]+c == 0, y[x], x]
Out[14]= {{y[x] ->
```

$$\frac{c}{b} - (-) + E^{((-a - \sqrt{a^2 - 4b})x)/2} C[1] + E^{((-a + \sqrt{a^2 - 4b})x)/2} C[2]}$$

Si vede che la soluzione è data da due esponenziali che dipendono dalle soluzioni di un'equazione di secondo grado. Se scegliamo $a=0$ e $b=c=k^2$, abbiamo radici complesse coniugate e la soluzione diviene un bel pasticcio.

```
In[15]:= DSolve[{y''[x] + k^2 y[x] + k^2 == 0,
```

```

y[0]==0,
y[1]==1}, y[x], x];
u[x_]:=Evaluate[Simplify[y[x]/.%[[1]]]];
u[x]

```

$$\text{Out[15]=} \quad -1 + \frac{-I b(-1+x) E + I b(-2+E)}{2 I b(-1+E)} + \frac{I b x E + I b(-1+2 E)}{2 I b(-1+E)}$$

Per semplificare questa espressione bisognerebbe conoscere e sapere usare bene la relazioni che legano gli esponenziali complesse alle funzioni trigonometriche ($e^{i\theta} = \cos \theta + i \sin \theta$). Il package `Algebra`Trigonometry`` contiene la funzione `ComplexToTrig` che prova proprio a trasformare un'espressione complessa in una reale usando le funzioni trigonometriche.

```

In[16]:= <<"Algebra`Trigonometry`"
In[17]:= Simplify[ComplexToTrig[u[x]]]

```

`Out[17]= Csc[b] (-Sin[b] + 2 Sin[b x] + Sin[b - b x])`
 Per la cronaca `Csc[x]` indica la cosecante ovvero l'inverso del seno.

```

In[18]:= 1/Sin[x]
Out[18]= Csc[x]

```

Casi più complicati

Esistono ovviamente equazioni che `DSolve` non riesce a risolvere.

```

In[19]:= DSolve[y'[x] == 1-y[x]^2, y[x], x]
Solve::tdep:
The equations appear to involve
transcendental functions of the
variables in an essentially
non-algebraic way.
Out[19]= Solve[x + (Log[1-y[x]]/2) - (Log[1+y[x]]/2) ==

```

`C[1], y[x]]`
 Il risultato è una equazione trascendente che *Mathematica* non sa risolvere. Conviene allora provare con il package `Calculus`DSolve`` che contiene un insieme di algoritmi di rinforzo.

```

In[20]:= <<"Calculus`DSolve`"
In[21]:= DSolve[y'[x] == 1-y[x]^2, y[x], x]
u[x_]:=Evaluate[Simplify[y[x]/.%[[1]]]];

```

$$\text{Out[21]=} \quad \frac{2 x E - 2 C[1]}{2 x E + 2 C[1]}$$

Ovviamente ci sono equazioni che neanche il pacchetto rinforzato riesce a risolvere e ve ne sono molte per cui una soluzione in forma elementare non esiste e bisogna ricorrere a funzioni speciali, sviluppi in serie o alla soluzione numerica diretta, ma questa è un'altra storia ...

Equazioni alle Differenze

L'equivalente discreto delle equazioni differenziali sono le equazioni alle differenze, ovvero equazioni che legano i valori assunti da una sequenza discreta. Il caso più noto tra gli informatici è l'equazione ricorrente $F_0=0, F_1=1, F_n=F_{n-1}+F_{n-2}, n>1$ che genera i numeri di Fibonacci. Proviamola a risolvere con `RSolve` (l'equivalente discreto di `DSolve`).

```

In[1]:= <<"DiscreteMath`RSolve`"
In[2]:= sol=RSolve[{a[n]-a[n-1]==a[n-2];n>1,
a[0]==a[1]==1}, a[n], n];
In[3]:= f[n_]:=Evaluate[Simplify[a[n]/.sol[[1]]]];
f[n]
Out[3]= (1 - Sqrt[5] n) (-1 + Sqrt[5]) / (2 (-1 + Sqrt[5]) + 2 Sqrt[5]) + (1 + Sqrt[5] n) (1 + Sqrt[5]) / (2 (-1 + Sqrt[5]) + 2 Sqrt[5])

```

`In[4]:= N[f[n]]`
`0.276393 (-0.618034)^n + 0.723607 1.61803^n`
 Si ottiene una strana porcheria che consiste in due esponenziali uno dei quali tende a zero con n che va all'infinito mentre l'altro cresce indefinitamente. Tabuliamo i primi 10 valori della soluzione. Guarda caso i loro valori numerici approssimati sono i numeri di Fibonacci.

```

In[5]:= risultati=Table[f[n],{n,10}];
In[6]:= N[risultati]
Out[6]= {1., 2., 3., 5., 8., 13., 21., 34., 55., 89.}
Non c'è trucco! Non c'è inganno! Sono proprio i numeri di Fibonacci anche se invece del calcolo numerico usiamo la semplificazione algebrica.

```

Bibliografia

S. Wolfram, *Mathematica. A System for Doing Mathematics by Computer, II Edition* (Addison Wesley, 1991).
Guide to standard Mathematica Packages, version 2.2. Cap. 3 Calculus`DSolve`. *Mathematica* Technical Report, Wolfram Research, 1993.
Guide to standard Mathematica Packages, version 2.2. Cap. 4 DiscreteMathematics`RSolve`. *Mathematica* Technical Report, Wolfram Research, 1993.
 Un buon libro di Analisi Matematica permetterà al lettore interessato di approfondire l'argomento in modo corretto e ordinato (e di vedere in altra luce la carrellata di esempi di questo articolo).
 Per le equazioni ricorrenti e i numeri di Fibonacci si veda F.Luccio. **La struttura degli Algoritmi**. Boringhieri, Torino 1982.

```
In[7]:=
Simplify[risultati]
Out[7]=
{1, 2, 3, 5, 8, 13, 21, 34, 55, 89}
```

Vediamo un altro marchingegno che si può applicare ad una equazione alle differenze: la ricerca di una funzione continua di una variabile x la cui serie di Taylor abbia come coefficienti la sequenza che risolve l'equazione data. Una simile funzione si dice funzione generatrice della sequenza. *Mathematica* offre la funzione **GeneratingFunction** che oltre ai parametri di **RSolve** riceve la variabile in cui si vuole il risultato.

```
In[8]:=
GeneratingFunction[
{a[n] - a[n-1] == a[n-2] /; n>=2,
a[0] == a[1] == 1}, a[n], n, z]
Out[8]=
1
{{-----}}
1 - z - z^2
```

Una rapida verifica mostra che è tutto a posto.

```
In[9]:=
Series[%[[1,1]],{z,0,10}]
Out[9]=
1 + z + 2z^2 + 3z^3 + 5z^4 + 8z^5 + 13z^6 + 21z^7
+ 34z^8 + 55z^9 + 89z^10 + O[z^11]
```

Un altro modo di trovare la stessa funzione generatrice è cercare la somma simbolica della serie di Taylor che ha come coefficienti una sequenza data.

```
In[10]:=
Simplify[PowerSum[f[n], {z, n, 0}]]
Out[10]=
1
-----
1 - z - z^2
```

Tanto per esplorare ancora un poco vediamo alcune altre equazioni alle differenze con la soluzione generale, e la funzione generatrice. Prendiamo prima una equazione alle differenze con la soluzione ovvia $a_n = 1$.

```
In[11]:=
RSolve[{(a[n] - a[n-1] == 0)/; n>0,
a[0] == 1}, a[n], n]
Out[11]=
{{a[n] -> 1}}
```

```
In[12]:=
GeneratingFunction[
```

```
{(a[n] - a[n-1] == 0)/; n>0,
a[0] == 1}, a[n], n, z]
Out[12]=
1
{{-----}}
1 - z
In[13]:=
Series[%[[1,1]],{z,0,5}]
Out[13]=
1+z+z^2+z^3+z^4+z^5+O[z^6]
In[14]:=
PowerSum[1, {z, n, 0}]
Out[14]=
1
-----
1 - z
```

Vediamo ora una equazione alle differenze con la soluzione $a_n = n$.

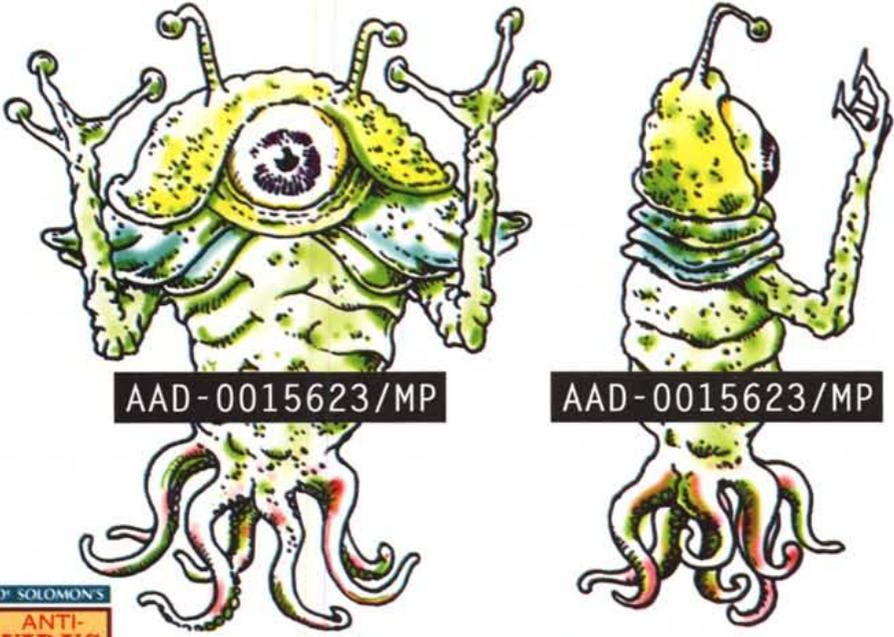
```
In[15]:=
RSolve[{(a[n] - a[n-1] == 1)/; n>0,
a[0] == 0}, a[n], n]
Out[15]=
{{a[n] -> n}}
In[16]:=
GeneratingFunction[
{(a[n] - a[n-1] == 1)/; n>0,
a[0] == 0}, a[n], n, z]
Out[16]=
z
{{-----}}
(-1 + z)^2
```

```
In[17]:=
Series[%[[1,1]],{z,0,5}]
Out[17]=
z+2z^2+3z^3+4z^4+5z^5+O[z^6]
In[18]:=
PowerSum[n, {z, n, 0}]
Out[18]=
z
-----
(-1 + z)^2
```

```
In[17]:=
Series[%[[1,1]],{z,0,5}]
Out[17]=
z+2z^2+3z^3+4z^4+5z^5+O[z^6]
In[18]:=
PowerSum[n, {z, n, 0}]
Out[18]=
z
-----
(-1 + z)^2
```

I (felici) possessori di una copia di *Mathematica* possono continuare a piacere con scherzetti del genere.

Francesco Romani è raggiungibile tramite Internet all'indirizzo romani@di.unipi.it



AAD-0015623/MP

AAD-0015623/MP

DR. SOLOMON'S
ANTI-VIRUS TOOLKIT
TELESOFTWARE

NOME IN CODICE: PIOVRA 01	MANIFESTAZIONE: Dalle 15,30 alle 16,30
INCREMENTO: 1,2 k	lascia gigantesche impronte "digitali"
EFFETTI: Scrittura della F.A.T.	sul video, simili a macchie di sangue.

IDENTIFICATO SEI GIORNI FA: SECURITY KIT® LO CONOSCE GIÀ. E LO TIENE A DISTANZA.

ECCO SECURITY KIT®, il primo Antivirus per Pc che si «auto-aggiorna» via etere mentre tu continui a lavorare.

Il problema fondamentale di un normale antivirus è di nascere già datato: riconosce i virus esistenti 'da prima' ma è quasi impotente contro i cento (o chissà quanti) nuovi e



polimorfici diffusi ogni settimana. SecurityKit® salta l'ostacolo rinnovandosi ogni giorno via etere: l'Antivirus Toolkit® Dr.Solomon, cuore difensivo del sistema, «impara» via Telesoftware® Rai tutte le informazioni per prevenire le più attuali situazioni di pericolo per il tuo Pc. Agendo senza che tu te ne accorga!



Il servizio di aggiornamento avviene costantemente mediante l'utilizzo del Telesoftware Rai®. Il decodificatore hardware è programmato per ricevere gli aggiornamenti anche in background, ovvero mentre si lavora su altre applicazioni. Il software di gestione permette inoltre di ricevere sul proprio pc il Televideo e tutti gli altri servizi open offerti dalla Rai.

HOT-LINE RIVENDITORI
1670-14130

È un prodotto:

RAI

RADIO
TELEVISIONE
ITALIANA

SIOSISTEMI
University of Networking®



**TOTOBIT
INFORMATICA®**

COMPATIBILE CON TUTTE
LE SCHEDE TELESOFTWARE

SecurityKit®

E PUOI PENSARE AD ALTRO.

