Equazioni differenziali: l'algoritmo di Eulero con il foglio elettronico

Michele Impedovo, Università Bocconi Milano

Abstract

Solving a differential equation is perhaps the most important and natural application of Calculus. But there are a few equations that have symbolic solutions. The only way to solve a differential equation is often to adopt a numerical method. This paper illustrates the Euler and Runge-Kutta algorithms for solving differential equations and systems, the implementation by Excel and the classic predator-prey growth model.

1. Introduzione

In un certo senso la risoluzione di un'equazione differenziale è la naturale e più importante applicazione del calcolo differenziale.
Innanzitutto dal punto di vista storico, perché sulle soluzioni di equazioni differenziali è nata la cosmologia moderna, attraverso l'analisi newtoniana del moto dei pianeti soggetti alla forza centrale di attrazione gravitazionale del Sole.
Ma è importante anche dal punto di vista didattico, perché risalire dalla relazione tra una funzione e le sue derivate al comportamento della funzione stessa fa comprendere in modo solido e ricco la magica danza tra derivate e integrali.
Le equazioni differenziali che ammettono una soluzione simbolica sono assai poche: nella gran parte dei casi occorre un algoritmo per approssimare la soluzione.
Molti degli algoritmi nati per questo scopo hanno la stessa struttura: supponiamo che $x(t)$ sia la funzione incognita, nella variabile t, dell'equazione differenziale

$x'(t) = g(t, x(t))$

e che

$x(t_0) = x_0$

sia la condizione iniziale. Supponiamo inoltre di voler stimare $x(t)$ nell'intervallo $[t_0, T]$; si stabiliscono un *passo* $\Delta t = \dfrac{T - t_0}{n}$, la sequenza di tempi equispaziati

$t_0,\ t_1 = t_0+\Delta t,\ t_2 = t_1+\Delta t,\ \ldots,\ t_n = t_{n-1}+\Delta t = T$

e si cerca, in modo ricorsivo, di approssimare i valori

$x(t_0),\ x(t_1),\ \ldots,\ x(t_n)$

della funzione incognita.
In generale, tanto più Δt è piccolo, tanto migliore è l'approssimazione.
L'algoritmo di Eulero è il più semplice e intuitivo degli algoritmi di questa famiglia; l'algoritmo di Runge-Kutta ne è un naturale e potente raffinamento, utile in particolare per trattare i sistemi di equazioni differenziali.
In questo articolo parleremo di:

- algoritmo di Eulero;
- algoritmo di Runge-Kutta;
- applicazione dei due algoritmi a equazioni e sistemi di equazioni differenziali;
- implementazione dei due algoritmi in un foglio Excel;
- applicazione al classico modello bidimensionale *preda-predatore*.

2. Algoritmo di Eulero

Una delle grandi conquiste del calcolo differenziale (e una delle sue motivazioni più profonde) è la possibilità di approssimare i valori che una funzione assume vicino ad un punto in cui essa è nota: è sufficiente sommare, a tale valore noto, il *differenziale primo*.

Un esempio banale: per approssimare $\sqrt{10}$, consideriamo la funzione $f(x)=\sqrt{x}$, la cui derivata prima è $f'(x)=\dfrac{1}{2\sqrt{x}}$, e il punto $x_0=9$, in cui risulta $f(9)=3$, $f'(9)=1/6$.

Se x è vicino a 9:

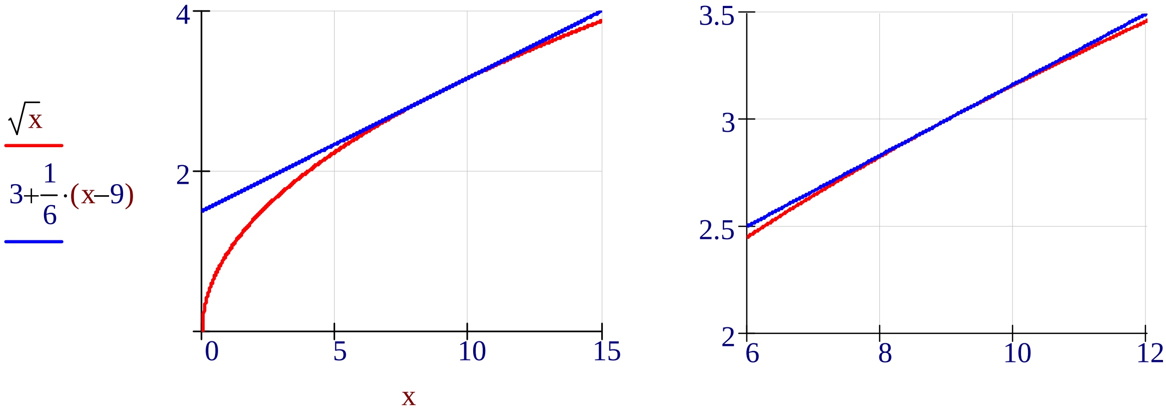
$$\sqrt{x} = f(x) \approx f(9)+f'(9)(x-9), \qquad f(x) \approx 3+\frac{1}{6}(x-9).$$

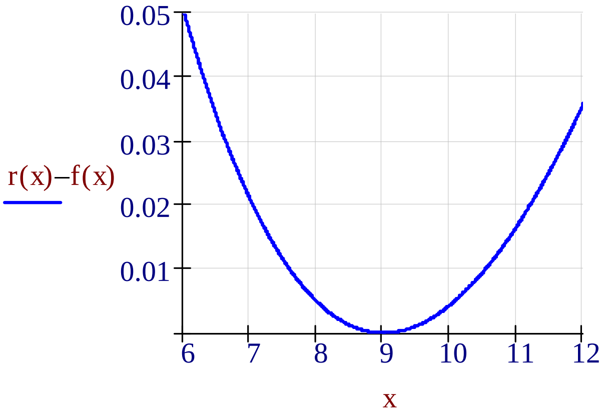
Dunque, per esempio:

$$\sqrt{10} = f(10) \approx 3+\frac{1}{6} \approx 3.167.$$

I grafici seguenti mostrano:

- $f(x)$ e la retta tangente $r(x) = 3+\dfrac{1}{6}(x-9)$ nell'intervallo $[0, 15]$;
- $f(x)$ e $r(x)$ nell'intervallo $[6, 12]$;
- la curva d'errore $r(x)-f(x)$ nell'intervallo $[6, 12]$.





Sull'ultimo grafico leggiamo che l'errore per la stima di $\sqrt{10}$ è circa 0.005, mentre per la stima di $\sqrt{12}$ è 0.035.

Torniamo all'equazione differenziale $x'(t)=g(t,x(t))$ e alla funzione incognita $x(t)$ che vogliamo approssimare nei punti $t_0, t_1, \ldots, t_n$.

Se t_k e t_{k+1} sono abbastanza vicini, possiamo ben approssimare $x(t_{k+1})$ mediante la somma tra $x(t_k)$ e il differenziale primo:

$x(t_{k+1}) \approx x(t_k) + x'(t_k)\Delta t$;

d'altra parte dall'equazione differenziale ricaviamo

$x'(t_k) = g(t_k, x(t_k))$

e dunque l'algoritmo di Eulero è sintetizzato dalla seguente relazione ricorsiva:

$$\boxed{x(t_{k+1}) \approx x(t_k) + g(t_k, x(t_k))\Delta t.}$$

Applichiamo l'algoritmo di Eulero alla classica equazione differenziale $x' = x$, con la condizione iniziale $x(0) = 1$, la cui soluzione è $x(t) = e^t$.

Cerchiamo la soluzione nell'intervallo $[0, 1]$ e poniamo $\Delta t = 1/n$. Risulta:

$x(0) = 1$

$$x\left(\frac{1}{n}\right) \approx x(0)+x'(0)\frac{1}{n} = x(0)+x(0)\,\frac{1}{n} = 1+\frac{1}{n}$$

$$x\left(\frac{2}{n}\right) \approx x\left(\frac{1}{n}\right)+x'\left(\frac{1}{n}\right)\frac{1}{n} = x\left(\frac{1}{n}\right)+x\left(\frac{1}{n}\right)\frac{1}{n} = \left(1+\frac{1}{n}\right)+\left(1+\frac{1}{n}\right)\frac{1}{n} = \left(1+\frac{1}{n}\right)\left(1+\frac{1}{n}\right) = \left(1+\frac{1}{n}\right)^2$$

$$x\left(\frac{3}{n}\right) \approx x\left(\frac{2}{n}\right)+x'\left(\frac{2}{n}\right)\frac{1}{n} = x\left(\frac{2}{n}\right)+x\left(\frac{2}{n}\right)\frac{1}{n} = \left(1+\frac{1}{n}\right)^2+\left(1+\frac{1}{n}\right)^2\frac{1}{n} = \left(1+\frac{1}{n}\right)^2\left(1+\frac{1}{n}\right) = \left(1+\frac{1}{n}\right)^3$$

…

$$x\left(\frac{n}{n}\right) \approx x(1) = \left(1+\frac{1}{n}\right)^n.$$

Si ritrova così un risultato ben noto, che ci fa attribuire ad Eulero la piena coscienza del ruolo del numero e nel calcolo infinitesimale: al tendere di n a infinito la successione

$$\left(1+\frac{1}{n}\right)^n$$

tende al numero *di Eulero*: e.

Svolgiamo ora un esempio significativo dell'algoritmo di Eulero. Supponiamo di voler risolvere l'equazione differenziale nella funzione incognita $x(t)$

$$x' = \frac{x-t}{x+t}$$

con la condizione iniziale

$x(0) = 1$

nell'intervallo $t \in [0, 4]$.

Si tratta di un'equazione differenziale che ammette un'unica soluzione, che però non si può scrivere esplicitamente in forma simbolica. Se chiediamo a MAPLE la soluzione simbolica, otteniamo la funzione incognita definita implicitamente da un'equazione in t, x.

```
dsolve({diff(x(t),t)=(x(t)-t)/(x(t)+t),x(0)=1},x(t));
```

$$x(t) = \mathrm{RootOf}\left(\ln\left(\frac{t^2+\_Z^2}{t^2}\right) + 2\arctan\left(\frac{\_Z}{t}\right) + 2\ln(t) - \pi\right)$$

Dunque non ci resta che percorrere la strada dell'approssimazione.

Fissiamo un passo di approssimazione, per esempio $\Delta t = 0.1$; in questo modo l'algoritmo di Eulero fornirà le approssimazioni dei valori
$x(0.1), x(0.2), \dots, x(4)$.
Conosciamo, della funzione incognita $x(t)$, la condizione iniziale $x(0) = 1$, e dunque il punto da cui parte: è il punto $(0, 1)$; ma l'equazione differenziale ci fornisce anche, in tale punto, la derivata $x'(0)$. Infatti sostituendo 0 a t e 1 a x nell'equazione differenziale, otteniamo
$$x'(0) = \frac{1-0}{1+0} = 1.$$
Mediante il differenziale primo (l'algoritmo di Eulero è tutto qui), possiamo approssimare $x(0.1)$:
$$x(t_1) \approx x(t_0)+x'(t_0)\Delta t.$$
Nel nostro esempio
$$x(0.1) \approx x(0) + x'(0)\cdot 0.1 = 1+1\cdot 0.1 = 1.1.$$
Con l'approssimazione di $x(0.1)$ possiamo approssimare $x'(0.1)$: sostituendo nell'equazione differenziale $t = 0.1$ e $x = 1.1$ otteniamo
$$x'(0.1) \approx \frac{1.1-0.1}{1.1+0.1} \approx 0.83.$$
Possiamo ora nello stesso modo approssimare $x(0.2)$:
$$x(0.2) \approx x(0.1) + x'(0.1)\cdot 0.1 \approx 1.1 + 0.83\cdot 0.1 \approx 1.183.$$
Proseguiamo così fino a $t_{40} = 4$.

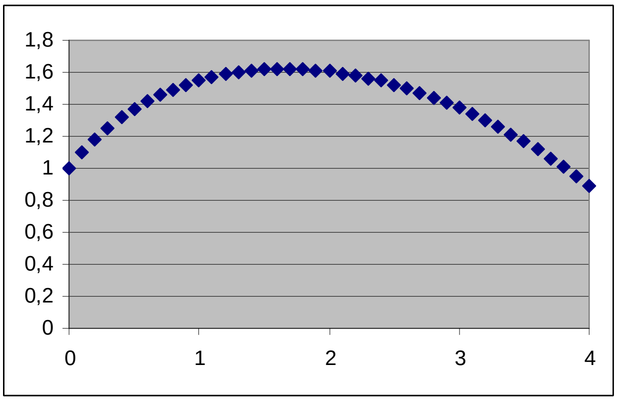
3. Algoritmo di Eulero e foglio elettronico

Vediamo ora come implementare questo algoritmo con Excel. Scriviamo:

- in A2 il passo dell'approssimazione;
- in B2 il valore iniziale t_0, cioè 0;
- in C2 la condizione iniziale $x(0)$, cioè 1
- in D2 scriviamo la formula =(C2–B2)/(C2+B2), che traduce l'equazione differenziale;
- in B3 la formula =B2+$A$2, che fa passare da t a $t+\Delta t$;
- in C3 il vero e proprio algoritmo di Eulero: =C2+D2*$A$2;
- copiamo D2 in D3.

Ora selezioniamo le celle B3:D3 e copiamo verso il basso, fino al valore di t desiderato. Nella figura seguente sono mostrate le prime e le ultime celle della tabella, e il grafico nell'intervallo $[0, 4]$.

	A	B	C	D
1	**dt**	**t**	**x(t)**	**x'(t)**
2	0.1	0	1	1
3		0.1	1.1	0.833333
4		0.2	1.183333	0.710843
5		0.3	1.254418	0.614003
6		0.4	1.315818	0.53375
7		0.5	1.369193	0.46501
8		0.6	1.415694	0.404672
9		0.7	1.456161	0.350698
10		0.8	1.491231	0.301685
11		0.9	1.521399	0.256628
12		1	1.547062	0.214782



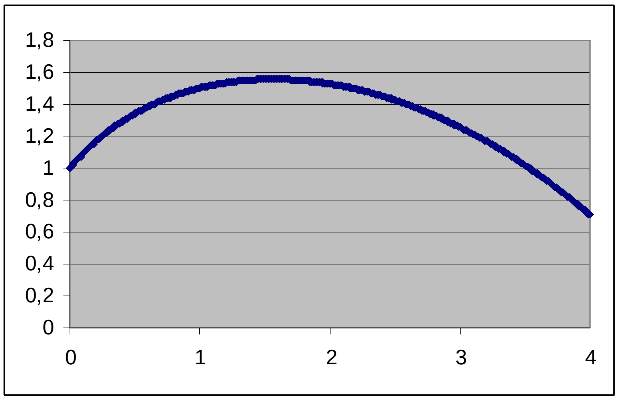
	A	B	C	D
39		3.7	1.063622	-0.55344
40		3.8	1.008278	-0.580608
41		3.9	0.950217	-0.608175
42		4	0.889399	-0.636193

Con $\Delta t = 0.1$ si ottengono le approssimazioni $x(1) = 1.547$ e $x(4) = 0.8894$. Un software professionale di matematica fornisce

$x(1) = 1.49828\ldots$
$x(4) = 0.686569\ldots$

L'errore relativo nella stima di $x(t)$ è circa il 3% in $t = 1$, e addirittura il 30% in $t = 4$.
Diminuendo Δt, per esempio $\Delta t = 0.01$, l'approssimazione migliora. È sufficiente, allo scopo, modificare il parametro nella cella A2 e copiare le celle B3:D3 fino ad ottenere 4 nella colonna B (quindi occorre copiare fino alla riga 402).

	A	B	C	D
1	**dt**	**t**	**x(t)**	**x'(t)**
2	0.01	0	1	1
3		0.01	1.01	0.980392
4		0.02	1.019804	0.961531
5		0.03	1.029419	0.943365
6		0.04	1.038853	0.925847



	A	B	C	D
101		0.99	1.501052	0.205155
102		1	1.503104	0.200992
103		1.01	1.505114	0.196855

	A	B	C	D
399		3.97	0.728314	-0.689968
400		3.98	0.721414	-0.693108
401		3.99	0.714483	-0.696254
402		4	0.70752	-0.699408

Risulta ora $x(1) = 1.503$ e $x(4) = 0.707$; l'approssimazione è migliorata circa di un fattore 10: l'errore relativo è 0.3% in $t = 1$ e 3% in $t = 4$.
Si dimostra che l'errore locale è proporzionale a Δt; ma a mano a mano che t si allontana da t_0 l'errore si accumula e l'approssimazione si deteriora.

Per comprendere meglio la propagazione dell'errore nell'algoritmo di Eulero, risolviamo la classica equazione differenziale del secondo ordine
$x'' = -x$
con le condizioni iniziali
$x(0) = 0$, $x'(0) = 1$,
la cui soluzione, come è noto, è
$x(t) = \sin(t)$.
Un'equazione differenziale del 2° ordine può essere trasformata in un sistema di 2 equazioni differenziali del 1° ordine. Se poniamo
$x' = y$
e di conseguenza
$x'' = y'$,
otteniamo il sistema

$$\begin{cases} x' = y \\ y' = -x \end{cases}$$

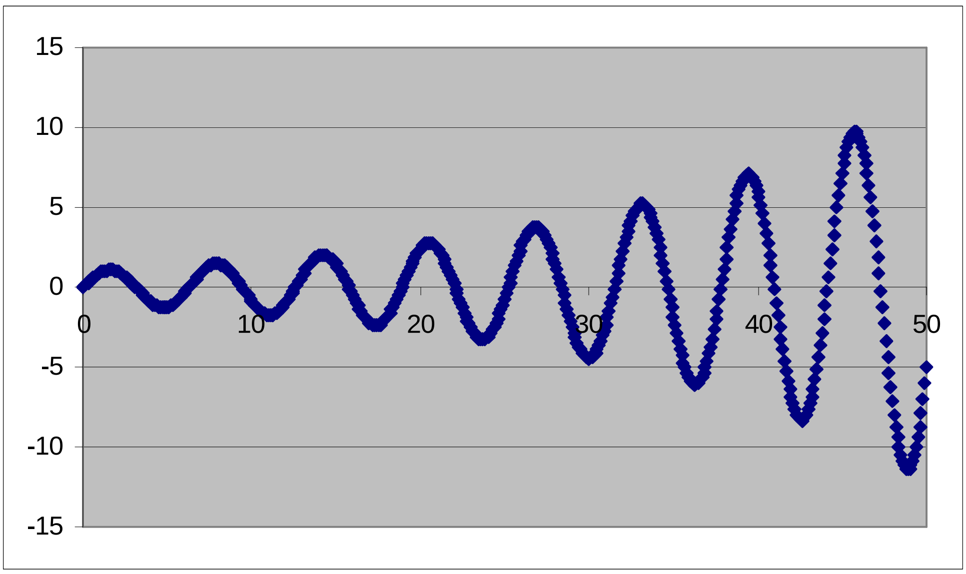
Fissiamo ancora $\Delta t = 0.1$. Scriviamo:

- in A2 il passo dell'approssimazione
- in B2 il valore iniziale t_0, cioè 0
- in C2 la condizione iniziale $x(0) = 0$
- in D2 la condizione iniziale $y(0) = 1$
- in E2 la formula =D2
- in F2 la formula =–C2
- in B3 la formula =B2+$A$2, che fa passare da t a $t+\Delta t$
- in C3 la formula =C2+E2*$A$2
- in D3 la formula =D2+F2*$A$2
- copiamo le celle D2:E2 in D3:E3.

Ora selezioniamo le celle B3:F3 e copiamo verso il basso, fino $t = 50$ (circa 16π, in modo da osservare $x(t)$ sull'intervallo di tempo di 8 periodi). Nella figura seguente sono mostrate le prime righe della tabella.

	A	B	C	D	E	F
1	**dt**	**t**	**x(t)**	**y(t)**	**x'(t)**	**y'(t)**
2	0.1	0	0	1	1	0
3		0.1	0.1	1	1	-0.1
4		0.2	0.2	0.99	0.99	-0.2
5		0.3	0.299	0.97	0.97	-0.299
6		0.4	0.396	0.9401	0.9401	-0.396
7		0.5	0.49001	0.9005	0.9005	-0.49001
8		0.6	0.58006	0.851499	0.851499	-0.58006
9		0.7	0.66521	0.793493	0.793493	-0.66521
10		0.8	0.744559	0.726972	0.726972	-0.744559
11		0.9	0.817256	0.652516	0.652516	-0.817256
12		1	0.882508	0.57079	0.57079	-0.882508

La figura seguente mostra il grafico nell'intervallo $[0, 50]$. Come si vede, la periodicità è andata perduta: anziché oscillare tra -1 e 1 la soluzione di Eulero ha un'ampiezza crescente, addirittura maggiore di 10 dopo solo 8 periodi. Si può dimostrare che allontanandosi da t_0 l'errore è proporzionale a $e^{t_i - t_0}$.



La morale è che l'algoritmo di Eulero non è adatto se si vuole approssimare la soluzione in un intervallo troppo ampio. Occorre un algoritmo più sofisticato, che vedremo fra poco.

4. Algoritmo di Runge-Kutta

Con l'algoritmo di Eulero l'approssimazione di $x(t_{k+1})$ consiste nel sommare a $x(t_k)$ il differenziale primo $x'(t_k)\Delta t$:

$$x(t_{k+1}) \approx x(t_k) + x'(t_k)\Delta t$$

Il teorema di Lagrange (o teorema del valor medio) consente di trasformare questa approssimazione in un'<u>uguaglianza</u>: infatti, se $x(t)$ è derivabile in $[t_k, t_{k+1}]$, esiste un punto $c \in (t_k, t_{k+1})$ tale che

$$x(t_{k+1}) = x(t_k) + x'(c)\Delta t$$

Non sappiamo dove sia c, ma se Δt è *piccolo*, possiamo ipotizzare che scegliendo c come <u>punto medio</u> dell'intervallo $[t_k, t_{k+1}]$ l'approssimazione possa migliorare:

$$x(t_{k+1}) = x(t_k) + x'\left(t_k + \frac{\Delta t}{2}\right)\Delta t$$

dove

$$x'\left(t_k + \frac{\Delta t}{2}\right) = g\left(t_k + \frac{\Delta t}{2}, x\left(t_k + \frac{\Delta t}{2}\right)\right).$$

L'*algoritmo di Runge-Kutta* (Carle Runge, 1856-1927, Martin Kutta, 1867-1944) è sintetizzato dalla seguente relazione ricorsiva:

$$\boxed{x(t_{k+1}) = x(t_k) + g\left(t_k + \frac{\Delta t}{2}, x\left(t_k + \frac{\Delta t}{2}\right)\right)\Delta t.}$$

Riprendiamo l'esempio precedente $f(x) = \sqrt{x}$ e $x_0 = 9$; con il differenziale primo avevamo ottenuto l'approssimazione $\sqrt{10} \approx 3.167$, con un errore relativo pari a 0.1%. Utilizzando il teorema di Lagrange nel punto medio tra 9 e 10 otteniamo

$\sqrt{10} \approx f(3)+f'(9.5)\cdot 1 = 3.16222\ldots$

In realtà risulta

$\sqrt{10} = 3.16227\ldots$

e l'errore relativo è ora 0.002%.

Applichiamo l'algoritmo di Runge-Kutta di nuovo all'equazione differenziale

$$x' = \frac{x-t}{x+t}$$

con la condizione iniziale $x(0) = 1$. Poniamo ancora $\Delta t = 0.1$; approssimando $x(0.1)$ con l'algoritmo di Eulero avevamo ottenuto:

$$x(0.1) \approx x(0) + x'(0)\cdot 0.1 = 1 + 1\cdot 0.1 = 1.1.$$

Con l'algoritmo di Runge-Kutta risulta:

$$x(0.1) \approx x(0) + x'(0.05)\cdot 0.1$$

Il compito ulteriore, rispetto all'algoritmo di Eulero, è quello di stimare $x'(0.05)$; dall'equazione differenziale risulta

$$x'(0.05) = \frac{x(0.05) - 0.05}{x(0.05) + 0.05}.$$

Per approssimare $x'(0.05)$ occorre dunque approssimare $x(0.05)$; per fare questo ricorriamo ancora al differenziale primo dell'algoritmo di Eulero:

$x(0.05) \approx x(0) + x'(0) \cdot 0.05$ = 1+1·0.05 = 1.05.

Sostituendo:

$$x'(0.05) \approx \frac{1.05 - 0.05}{1.05 + 0.05} \approx 0.909$$

e finalmente

$x(0.1) \approx x(0) + x'(0.05) \cdot 0.1$ = 1+0.909·0.1 = 1.0909.

Si prosegue poi nello stesso modo:

$x(0.2) \approx x(0.1) + x'(0.15) \cdot 0.1$

e così via, con passo Δt, fino a 4.

5. Algoritmo di Runge-Kutta e foglio elettronico

Vediamo ora l'implementazione dell'algoritmo di Runge-Kutta con Excel.
Le prime quattro colonne A, B, C, D sono le stesse dell'algoritmo di Eulero. Ci servono due nuove colonne E e F, in cui calcolare prima $x(t+\Delta t/2)$ con il differenziale primo, e poi $x'(t+\Delta t/2)$ mediante l'equazione differenziale.
In E2 immettiamo la formula

=C2+D2*$A$2/2

In F2 immettiamo la formula

=(E2-B2-$A$2/2)/(E2+B2+$A$2/2)

Passiamo alla riga 3: in B3 scriviamo la formula

=B2+$A$2

In C3 immettiamo la formula che rappresenta il cuore dell'algoritmo di Runge-Kutta:

=C2+F2*$A$2

Copiamo le celle D2:F2 in D3:F3, poi selezioniamo l'intervallo B2:F2 e copiamo verso il basso, fino alla riga 42 (cioè fino a $t = 4$).

	A	B	C	D	E	F
1	**dt**	**t**	**x(t)**	**x'(t)**	**x(t+dt/2)**	**x'(t+dt/2)**
2	0.1	0	1	1	1.05	0.909091
3		0.1	1.090909	0.832061	1.132512	0.766084
4		0.2	1.167517	0.707499	1.202892	0.655859
5		0.3	1.233103	0.608637	1.263535	0.56617
6		0.4	1.28972	0.526549	1.316048	0.490388
7		0.5	1.338759	0.456155	1.361567	0.424556
8		0.6	1.381215	0.394311	1.40093	0.366141
9		0.7	1.417829	0.338946	1.434776	0.313431
10		0.8	1.449172	0.288627	1.463603	0.265215
11		0.9	1.475693	0.242326	1.48781	0.220612
12		1	1.497755	0.199281	1.507719	0.178956

	A	B	C	D	E	F
39		3.7	0.883745	-0.6144	0.853025	-0.629363
40		3.8	0.820809	-0.644734	0.788572	-0.659994
41		3.9	0.754809	-0.675686	0.721025	-0.691278
42		4	0.685681	-0.707329	0.650315	-0.723289

Con $\Delta t = 0.1$ l'algoritmo di Runge-Kutta fornisce una stima di $x(t)$ con un errore relativo pari a 0.03% in $t = 1$ e 0.1% in $t = 4$.
Con $\Delta t = 0.01$ l'errore è praticamente nullo. Si dimostra che l'errore locale è proporzionale a Δt^2.

Proviamo a controllare se con l'algoritmo di Runge-Kutta riusciamo a migliorare l'approssimazione della soluzione dell'equazione differenziale $x'' = -x$, che trasformiamo nel sistema di equazioni differenziali

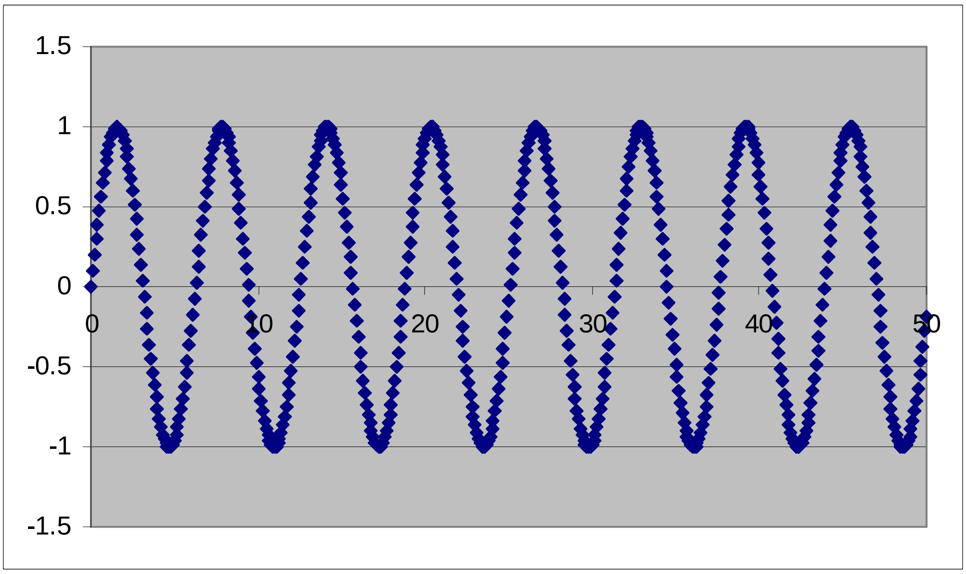
$$\begin{cases} x' = y \\ y' = -x \end{cases}$$

con le condizioni iniziali $x(0) = 0$, $x'(0) = 1$. Le prime righe della tabella sono le seguenti.

	A	B	C	D	E	F	G	H	I	J
1	**dt**	**t**	**x(t)**	**y(t)**	**x'(t)**	**y'(t)**	**x(t+dt/2)**	**y(t+dt/2)**	**x'(t+dt/2)**	**y'(t+dt/2)**
2	0.1	0	0	1	1	0	0.05	1	1	-0.05
3		0.1	0.1	0.995	0.995	-0.1	0.14975	0.99	0.99	-0.14975
4		0.2	0.199	0.980025	0.980025	-0.199	0.248001	0.970075	0.970075	-0.248001
5		0.3	0.296008	0.955225	0.955225	-0.296008	0.343769	0.940425	0.940425	-0.343769
6		0.4	0.39005	0.920848	0.920848	-0.39005	0.436092	0.901346	0.901346	-0.436092
7		0.5	0.480185	0.877239	0.877239	-0.480185	0.524046	0.85323	0.85323	-0.524046
8		0.6	0.565507	0.824834	0.824834	-0.565507	0.606749	0.796559	0.796559	-0.606749
9		0.7	0.645163	0.764159	0.764159	-0.645163	0.683371	0.731901	0.731901	-0.683371
10		0.8	0.718353	0.695822	0.695822	-0.718353	0.753145	0.659904	0.659904	-0.753145
11		0.9	0.784344	0.620508	0.620508	-0.784344	0.815369	0.58129	0.58129	-0.815369
12		1	0.842473	0.538971	0.538971	-0.842473	0.869421	0.496847	0.496847	-0.869421

La tabella è stata così costruita:
- in C2: 0
- in D2: 1 (le condizioni iniziali)
- in E2: =D2
- in F2: = –C2 (le due equazioni differenziali.
- in G2: =C2+E2*$A$2/2
- in H2: =D2+F2*$A$2/2 (approssimazione di $x(0.05)$ e $y(0.05)$ mediante il differenziale primo)
- in I2: =H2
- in J2: = –G2 (approssimazione di $x'(0.05)$ e $y'(0.05)$
- in C3: =C2+I2*$A$2
- in D3: =D2+J2*$A$2 (il vero e proprio algoritmo di Runge-Kutta)

Copiamo le celle E2:J2 in E3:J3, poi le celle B3:J3 verso il basso, fino a $t = 50$ (riga 502).
Ecco il grafico.



La periodicità è mantenuta, anche a notevole distanza da t_0: l'algoritmo di Runge-Kutta riesce a controllare l'errore globale.

6. Il modello preda-predatore di Lotka-Volterra

Possiamo facilmente adattare l'algoritmo di Runge-Kutta per approssimare la soluzione di un sistema di equazioni differenziali.
Supponiamo di voler descrivere la soluzione del seguente sistema di equazioni differenziali nelle funzioni incognite $x(t)$, $y(t)$.

$$\begin{cases} x' = x - 0.2xy \\ y' = -y + 0.2xy \end{cases}$$

con le condizioni iniziali $x(0) = 10$, $y(0) = 2$. Vogliamo analizzare le funzioni incognite nell'intervallo di tempo $[0, 8]$.
Si tratta del celebre modello *preda-predatore* formulato da Vito Volterra e Alfred Lotka. In un ambiente circoscritto convivono due specie, una delle quali (i predatori) si ciba dell'altra (le prede). Le funzioni incognite rappresentano:

- $x(t)$: il numero di prede al tempo t
- $y(t)$: il numero di predatori al tempo t

In assenza di predatori le prede crescerebbero con tasso di crescita costante a: $x' = ax$. La presenza dei predatori fa sì che il tasso di crescita a non sia costante, ma decrescente (per esempio linearmente) con il numero y di predatori:

$$x' = (a - by)x$$

I predatori, in assenza di prede, si estinguerebbero con tasso di mortalità costante c: $y' = -cy$. La presenza delle prede fa sì che il tasso di decrescita $-c$ dei predatori cresca (per esempio linearmente) con il numero delle prede:

$$y' = (-c + dx)y$$

Nell'esempio proposto i parametri sono così scelti: $a = c = 1$, $b = d = 0.2$; inizialmente le prede sono 5 volte più numerose dei predatori.
Scegliamo il passo $\Delta t = 0.1$. Per approssimare $x(0.1)$, $y(0.1)$ calcoliamo

$$x(0.1) \approx x(0) + x'(0.05) \cdot 0.1$$
$$y(0.1) \approx y(0) + y'(0.05) \cdot 0.1$$

Dunque ci serve calcolare $x(0.05)$ e $y(0.05)$, che approssimiamo mediante il differenziale primo:

$$x(0.05) \approx x(0)+x'(0)\cdot 0.05$$
$$y(0.05) \approx y(0)+y'(0)\cdot 0.05$$

Dopodiché si prosegue nello stesso modo per $t = 0.2, 0.3, \ldots, 8$.
Vediamo ora l'implementazione in Excel.
Scriviamo:

- in A2 il passo utilizzato, per esempio 0.1;
- in B2 il valore iniziale di t, cioè 0;
- in C2 il valore iniziale $x(0)$, cioè 10;
- in D2 il valore iniziale $y(0)$, cioè 2;
- in E2 il valore di $x'(0)$, cioè la formula =C2–0.2*C2*D2;
- in F2 il valore di $y'(0)$, cioè la formula = –D2+0.2*C2*D2;
- in G2 l'approssimazione di $x(0.05)$, cioè la formula =C2+E2*$A$2/2;
- in H2 l'approssimazione di $y(0.05)$, cioè la formula =D2+F2*$A$2/2;
- in I2 l'approssimazione di $x'(0.05)$, cioè la formula =G2–0,2*G2*H2;
- in J2 l'approssimazione di $y'(0.05)$, cioè la formula = –H2+0,2*G2*H2;
- in B3 aggiorniamo la variabile tempo incrementandola di 0.1;

Finalmente in C3 e D3 il vero e proprio algoritmo:

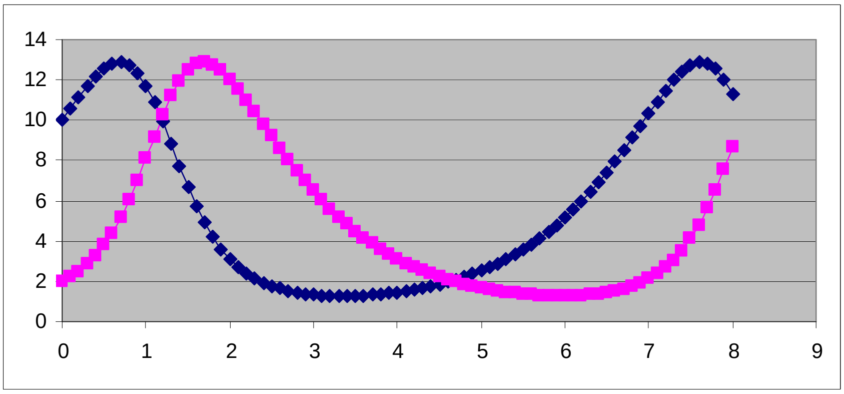
=C2+I2*$A$2
=D2+J2*$A$2

Si selezionano ora le celle E2:J2 e si copiano in E3:J3. Poi si seleziona tutta la riga B3:J3 e si copia verso il basso, fino alla riga 82.
La figura seguente mostra le prime e le ultime righe.

	A	B	C	D	E	F	G	H	I	J
1	**dt**	**t**	**x(t)**	**y(t)**	**x'(t)**	**y'(t)**	**x(t+dt/2)**	**y(t+dt/2)**	**x'(t+dt/2)**	**y'(t+dt/2)**
2	0,1	0	10	2	6	2	10,3	2,1	5,974	2,226
3		0,1	10,60	2,22	5,89	2,49	10,89	2,35	5,78	2,77
4		0,2	11,18	2,50	5,59	3,09	11,45	2,65	5,38	3,43
5		0,3	11,71	2,84	5,06	3,82	11,97	3,03	4,71	4,22
6		0,4	12,18	3,26	4,23	4,69	12,40	3,50	3,72	5,17
7		0,5	12,56	3,78	3,06	5,71	12,71	4,07	2,37	6,27

	A	B	C	D	E	F	G	H	I	J
78		7,6	12,85	4,80	0,52	7,53	12,88	5,17	-0,45	8,15
79		7,7	12,81	5,61	-1,57	8,76	12,73	6,05	-2,67	9,35
80		7,8	12,54	6,55	-3,88	9,87	12,35	7,04	-5,04	10,34
81		7,9	12,04	7,58	-6,21	10,67	11,72	8,11	-7,30	10,91
82		8	11,31	8,67	-8,30	10,94	10,89	9,22	-9,19	10,86

I grafici di $x(t)$ e $y(t)$ mostrano l'evoluzione periodica delle due specie.


14
12
10
8
6
4
2
0
0
1
2
3
4
5
6
7
8
9