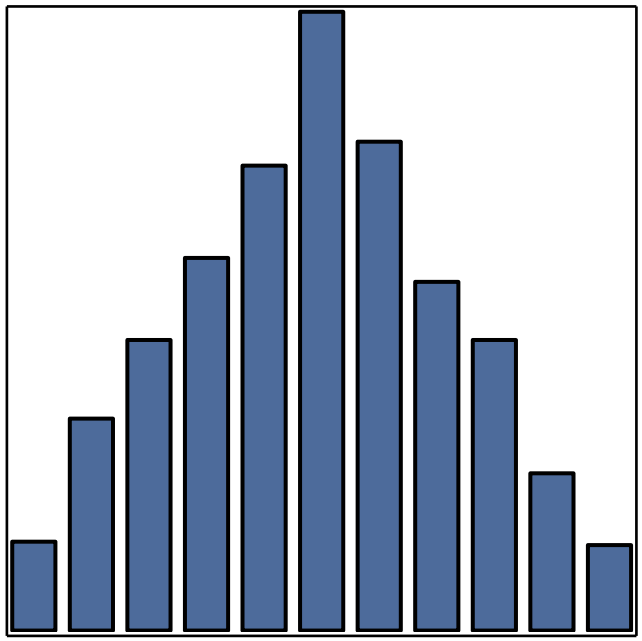
Dadi

```
dado1 := rand(1..6): # dado1 è una procedura che richiamata senza parametri produce un numero intero "casuale" tra 1 e 6
dado2 := rand(1..6): # dado2 lo è identico a dado1
[seq(dado1() + dado2(), n = 1..1000)]: # lista dei risultati di 1000 lanci di una coppia di dadi
map[2](numboccur, %, [$2..12]) # contiamo le frequenze assolute dei diversi risultati tra 2 e 12
```

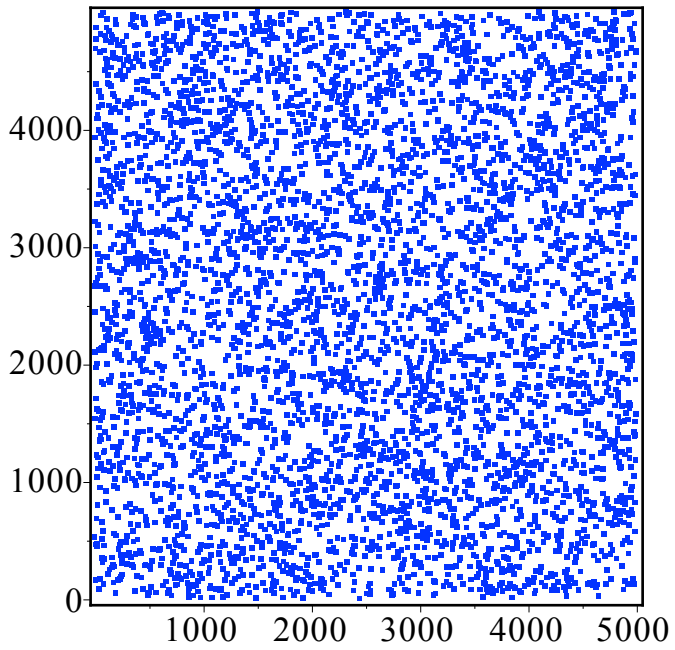
[26, 62, 85, 109, 136, 181, 143, 102, 85, 46, 25] **(1.1)**

```
Statistics[ColumnGraph](%, axes = boxed, tickmarks = [[], []]) # questa è la rappresentazione grafica
```

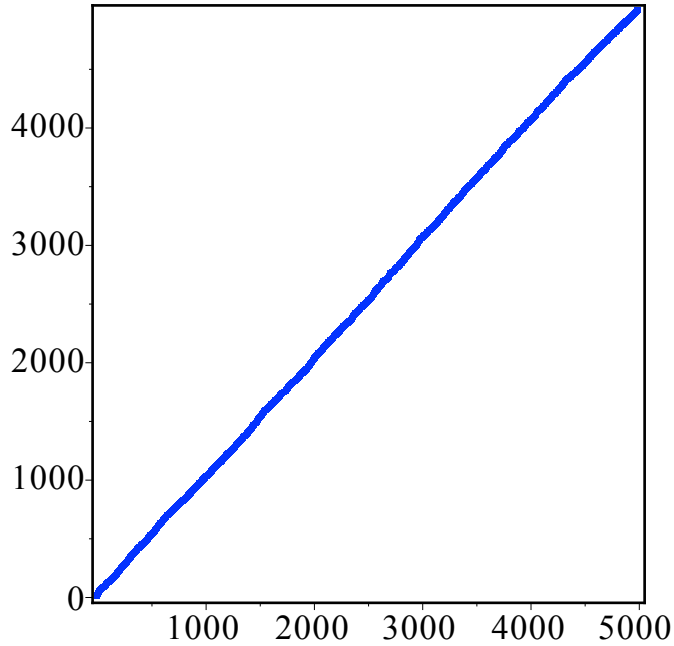


Punti

```
l := [seq(rand(5000)( ), i = 1 ..5000)] :  # lista di 5000 numeri interi "casuali" tra 1 e 5000
s := sort(l) :  # lista degli stessi numeri disposti in ordine crescente
plot([i $ i = 1 ..5000], l,    # l'uniformità della distribuzione è evidenziata dal grafico dei valori ...
      style = point, symbol = point, color = blue, axes = boxed)
```



```
plot( [i $ i = 1 ..5000], s,   # ... e ancor più da quello dei valori ordinati, che approssima la diagonale
      style = point, symbol = point, color = blue, axes = boxed)
```



$$punti := \left[seq\left(\left[\frac{rand(5000)(\,)}{5000.}, \frac{rand(5000)(\,)}{5000.} \right], i = 1 ..5000 \right) \right] :$$ # 5000 punti "casuali" uniformemente distribuiti nel quadrato $[0,1]^2$

$$sep := selectremove\left(p \rightarrow evalb\left(p[1]^2 + p[2]^2 \leq 1 \right), punti \right) :$$ # separiamo quelli interni alla circonferenza unitaria da quelli esterni

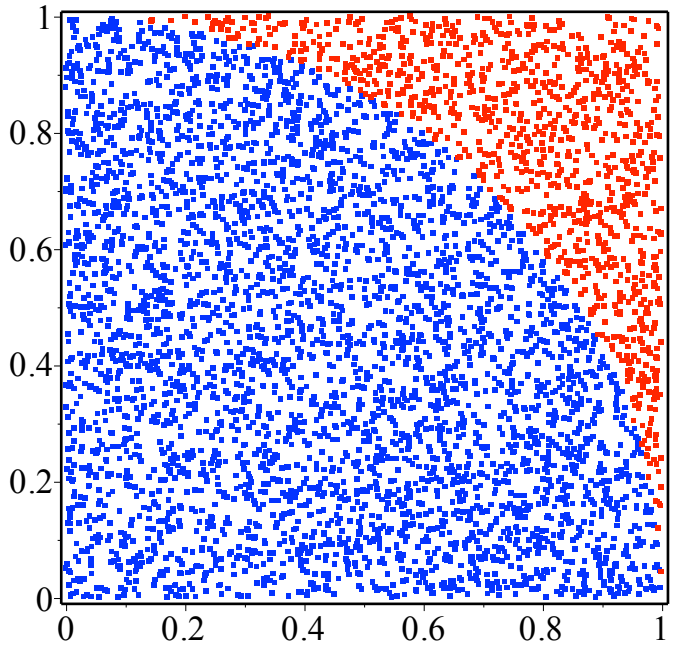
```
plot1 := plot(sep[1], style = point, symbol = point, color = blue, view = [0..1, 0..1])   # rappresentiamo quelli interni in blu ...
```

$$PLOT(\ldots) \tag{2.1}$$

```
plot2 := plot(sep[2], style = point, symbol = point, color = red, view = [0..1, 0..1])   # ... e quelli esterni in rosso
```

$$PLOT(\ldots) \tag{2.2}$$

```
plots[display](plot1, plot2, axes = boxed)   # visualizziamoli insieme
```




```
evalf(4 nops(sep[1])/5000)   # la frequenza relativa dei punti interni ci fornisce una stima di π
```

$$3.148800000 \tag{2.3}$$

Fitting

$dati := \left[seq\left(evalf\left(\left[x, x - \frac{10}{x} + \frac{rand(-1000..1000)()}{2000} \right] \right), x = 1..10, 0.05 \right) \right]$: # dati con un errore "casuale" rispetto a una legge teorica

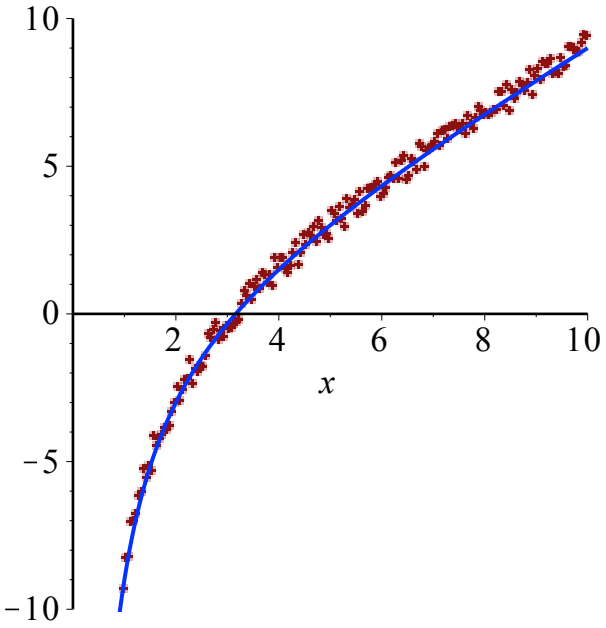
$dataplot := plot(dati, x = 0..10, view = [0..10, -10..10], style = point)$ # il plot dei dati ...

$$PLOT(...) \tag{3.1}$$

$funplot := plot\left(x - \frac{10}{x}, x = 0..10, view = [0..10, -10..10], color = blue \right)$ # ... e il plot della curva che rappresenta la legge teorica ...

$$PLOT(...) \tag{3.2}$$

$plots[display](dataplot, funplot)$ # ... visualizzati insieme



$CurveFitting[LeastSquares](dati, x, curve = a\,x + b)$ # retta di regressione

$$-6.02003321800036 + 1.62920969790966\,x \tag{3.3}$$

$CurveFitting[LeastSquares]\left(dati, x, curve = a\,x^2 + b\,x + c\right)$ # migliore approssimazione quadratica

$$-9.30065568743812 + 3.16973679134895\,x - 0.140047917585391\,x^2 \tag{3.4}$$

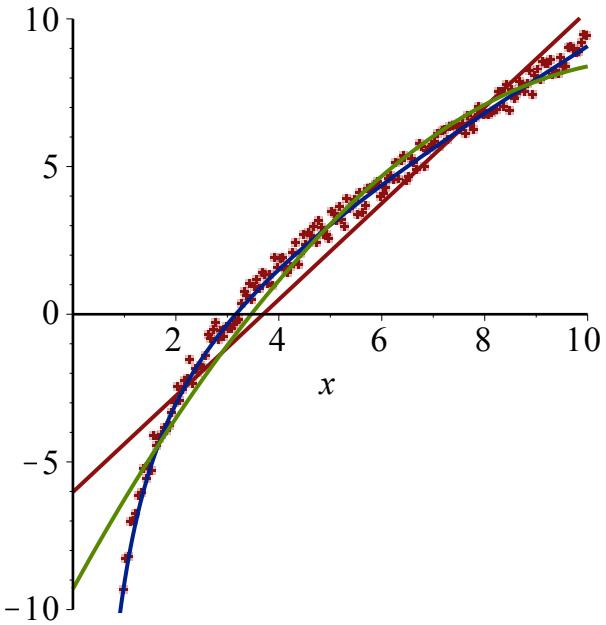
$CurveFitting[LeastSquares]\left(dati, x, curve = a\,x + \frac{b}{x}\right)$ # migliore stima dei parametri a e b nel giusto modello

$$1.00872104209131\,x - \frac{10.1055657847083}{x} \tag{3.5}$$

$fitplot := plot(\{\%, \%\%, \%\%\%\}, x = 0..10, view = [0..10, -10..10])$ # plot delle approssimazioni ottenute ...

$$PLOT(\ldots) \tag{3.6}$$

$plots[display](dataplot, fitplot)$ # ... visualizzati insieme ai dati



Medie

$Media(l) := \frac{apply(`+`, op(l))}{nops(l)}$ # media di una lista di dati numerici ...

$$l \to \frac{apply(`+`, op(l))}{nops(l)} \tag{4.1}$$

$Media([a, b, c])$ # ... e anche simbolici

$$\frac{1}{3} a + \frac{1}{3} b + \frac{1}{3} c \tag{4.2}$$

$MediaGen(k) := l \to \left(\frac{apply(`+`, op(map(`^`, l, k)))}{nops(l)} \right)^{\frac{1}{k}}$ # una definizione più generale (valida per dati a valori positivi)

$$k \to l \to (apply(`+`, op(map(`^`, l, k))) / nops(l))\,\hat{}\,(1/k) \tag{4.3}$$

$MediaAritmetica := MediaGen(1)$: # per k = 1 si ha la media aritmetica come sopra
$MediaAritmetica([a, b, c])$

$$\frac{1}{3} a + \frac{1}{3} b + \frac{1}{3} c \tag{4.4}$$

$MediaQuadratica := MediaGen(2)$: # per k = 2 si ha la media quadratica
$MediaQuadratica([a, b, c])$

$$\sqrt{\frac{1}{3} a^2 + \frac{1}{3} b^2 + \frac{1}{3} c^2} \tag{4.5}$$

$MediaArmonica := MediaGen(-1)$: # per k = -1 si ha la media armonica
$MediaArmonica([a, b, c])$

$$\frac{1}{\frac{1}{3\,a} + \frac{1}{3\,b} + \frac{1}{3\,c}} \tag{4.6}$$

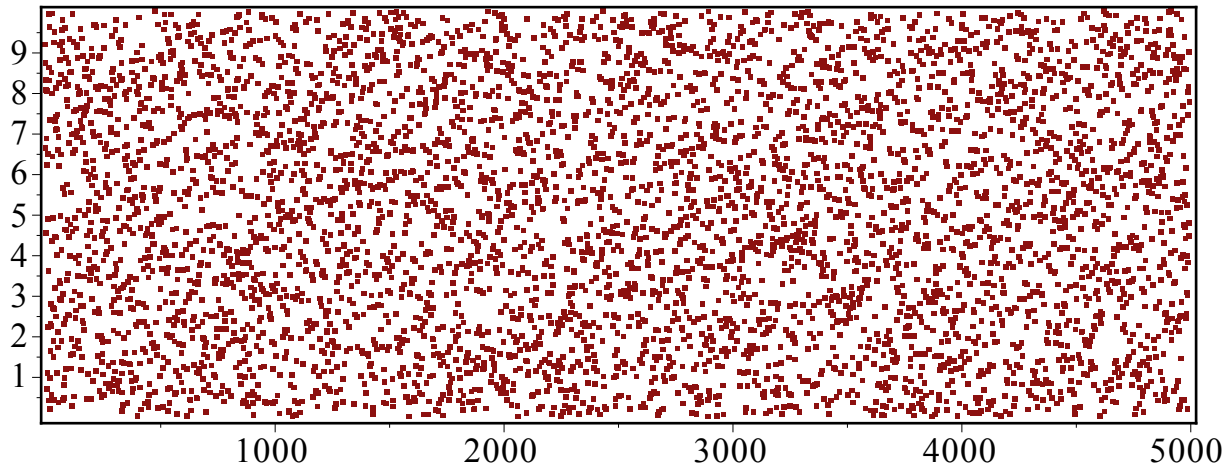
$MediaGeometrica(l) := apply(`\cdot`, op(l))^{\frac{1}{nops(l)}}$ # la media geometrica richiede un'altra definizione

$$l \rightarrow apply(`*`, op(l))^{\frac{1}{nops(l)}} \quad \textbf{(4.7)}$$

$MediaGeometrica([a, b, c])$

$$(a\ b\ c)^{1/3} \quad \textbf{(4.8)}$$

$dati := [seq(RandomTools[Generate](distribution(Uniform(0.0, 10.0))), n = 1..5000)]$: # dati uniformemente distribuiti nell'intervallo [0,10]
$plot([seq([n, dati[n]], n = 1..5000)],$ # l'uniformità è evidenziata dalla loro rappresentazione grafica ...
$style = point, symbol = point, axes = boxed)$



$numboccur(map(x \rightarrow evalb(x < 2.5), dati), true),$ # ... e anche dalle frequenze relative alla suddivisione in classi di uguale ampiezza
$numboccur(map(x \rightarrow evalb(2.5 \le x < 5), dati), true),$
$numboccur(map(x \rightarrow evalb(5 \le x < 7.5), dati), true),$
$numboccur(map(x \rightarrow evalb(7.5 \le x), dati), true)$

$$1248, 1243, 1241, 1268 \quad \textbf{(4.9)}$$

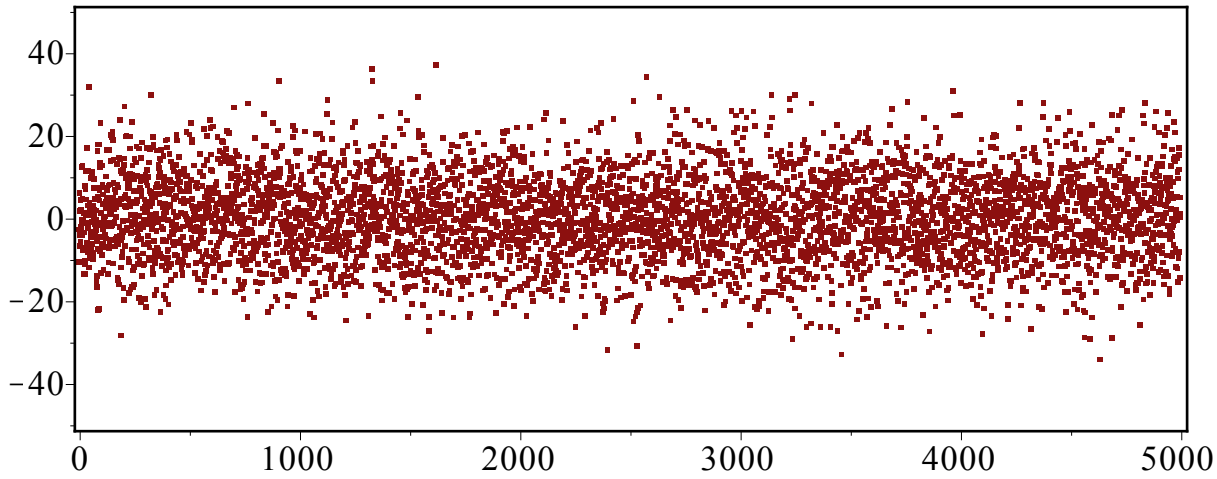
$$MediaAritmetica(dati)$$

$$5.01033732104856 \tag{4.10}$$

$$MediaQuadratica(dati)$$

$$5.77816678683359 \tag{4.11}$$

```
dati := [seq(RandomTools[Generate](distribution(Normal(0.0, 10.0))), n = 1..5000)]:  # dati con distribuzione normale intorno a 0
plot([seq([n, dati[n]], n = 1..5000)],  # ... come evidenziato dalla loro rappresentazione grafica
     style = point, symbol = point, axes = boxed, view = [0..5000, -50..50])
```



$$Media(dati) \quad \#\ la\ media\ è\ prossima\ a\ 0,\ come\ previsto$$

$$-0.128650707085032 \tag{4.12}$$

$$Mediana(l) := op\left(\text{floor}\left(\frac{nops(l)}{2}\right), sort(l)\right) \quad \#\ definizione\ della\ mediana$$

$$l \rightarrow op\left(\text{floor}\left(\frac{1}{2}\, nops(l)\right), sort(l)\right) \tag{4.13}$$

$$Mediana(dati) \quad \#\ anche\ la\ mediana\ è\ prossima\ a\ 0$$

$$-0.0539437847939151 \tag{4.14}$$

Crescita

```
CrescitaCostante := # questa procedura definisce una funzione p che esprime l'andamento della popolazione di anno in anno
 proc(i, c)         # assumendo la popolazione iniziale (anno 0) = i e un tasso di crescita annuale costante =c
  global p;         # p è una variabile globale, così per essere richiamabile fuori della procedura di definizione
  p := proc(t :: nonnegint)
   option remember;  # questa opzione serve a ricordare i valori di p già calcolati (anni precedenti), ciò è importante ...
   if t = 0 then i else (1 + c) p(t − 1) end if   # ... perché la definizione è ricorsiva e il calcolo riparte ogni volta da 0
  end proc;
  return p = eval(p)   # questo è il valore risultante della procedura: qui serve solo a visualizzare la definizione di p che è già stata data sopra
end proc :

CrescitaCostante(10, 0.1)   # modello con popolazione iniziale i = 10 e un tasso di crescita costante del 10 %
```

$$p = \mathbf{proc}(t::nonnegint)\ \mathbf{option}\ remember;\ \mathbf{if}\ t = 0\ \mathbf{then}\ 10\ \mathbf{else}\ (1 + 0.1) * p(t - 1)\ \mathbf{end\ if}\ \mathbf{end\ proc} \quad \textbf{(5.1)}$$

```
p(0)   # la popolazione iniziale ...
```

$$10 \quad \textbf{(5.2)}$$

```
p(1)   # ... quella alla fine del primo anno ...
```

$$11.0 \quad \textbf{(5.3)}$$

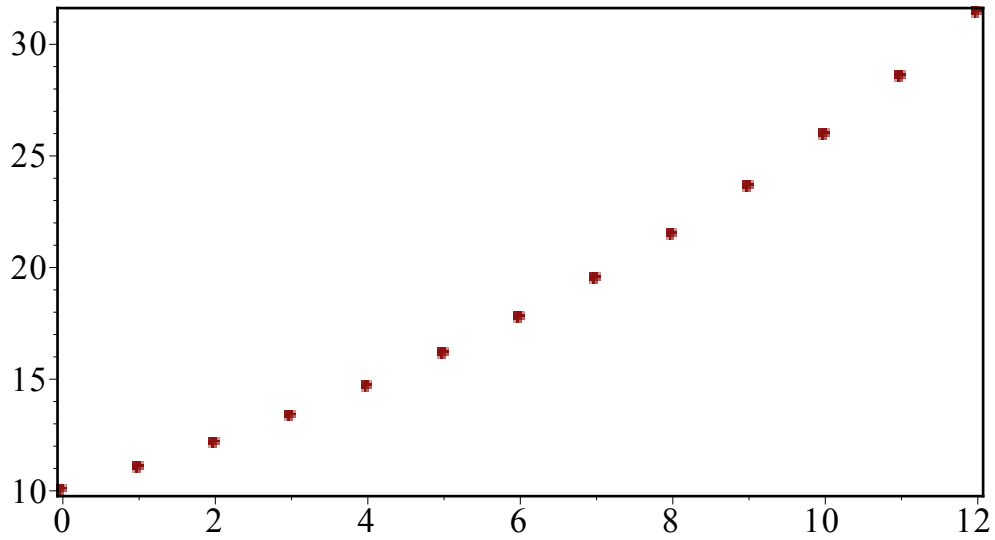
```
seq(p(t), t = 0 ..12)   # ... e così via, nei primi 12 anni
```

$$10, 11.0, 12.10, 13.310, 14.6410, 16.10510, 17.715610, 19.4871710, 21.43588810, 23.57947691, 25.93742460, 28.53116706, 31.38428377 \quad \textbf{(5.4)}$$

```
op(4, eval(p))   # questa è la tabella in cui sono registrati i valori già calcolati
```

$$table([0 = 10, 1 = 11.0, 2 = 12.10, 3 = 13.310, 4 = 14.6410, 5 = 16.10510, 6 = 17.715610, 7 = 19.4871710, 9 = 23.57947691, 8 = 21.43588810, 11 = 28.53116706, 10 = 25.93742460, 12 = 31.38428377]) \quad \textbf{(5.5)}$$

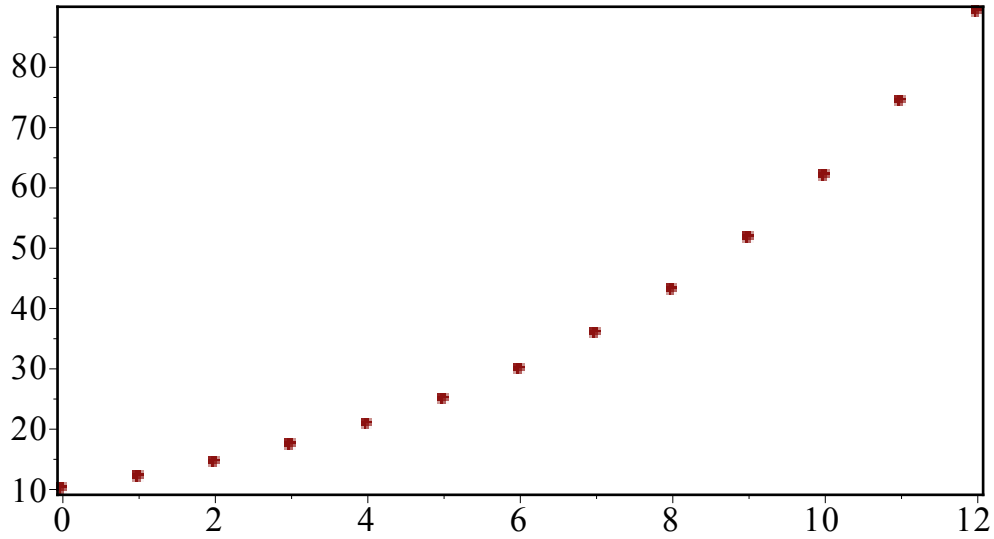
$plot([seq([t, p(t)], t = 0..12)], style = point, symbol = solidcircle, axes = boxed)$ # *la crescita è di tipo esponenziale*



CrescitaCostante(10, 0.2) *# con un tasso annuale del 20% ...*

p = **proc**(t::*nonnegint*) **option** *remember*; **if** $t = 0$ **then** 10 **else** $(1 + 0.2) * p(t - 1)$ **end if** **end proc** **(5.6)**

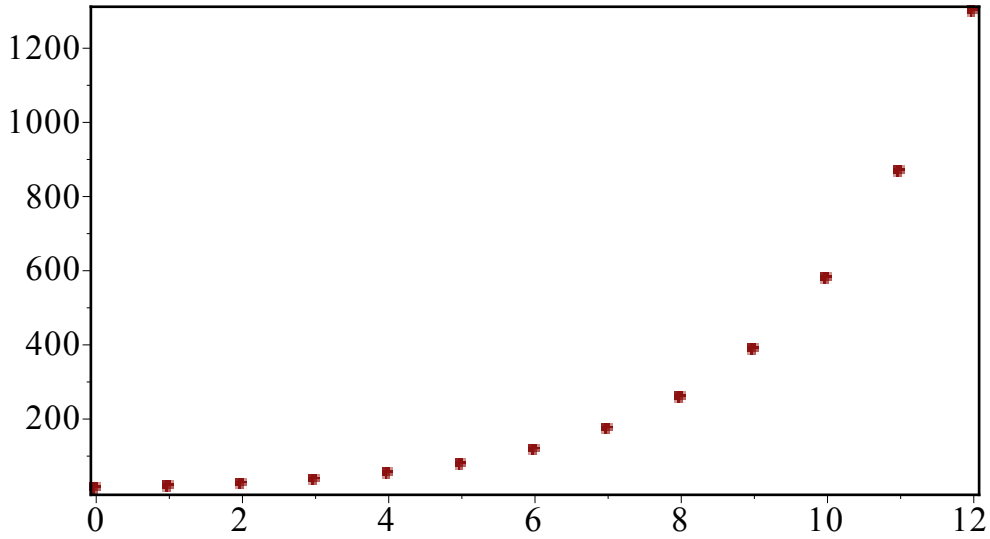
plot([*seq*([t, $p(t)$], $t = 0..12$)], *style* = *point*, *symbol* = *solidcircle*, *axes* = *boxed*) *# ... la crescita è più veloce*



CrescitaCostante(10, 0.5) # *aumentando il tasso fino al 50% ...*

$p = \mathbf{proc}(t::nonnegint)\ \mathbf{option}\ remember;\ \mathbf{if}\ t = 0\ \mathbf{then}\ 10\ \mathbf{else}\ (1 + 0.5) * p(t - 1)\ \mathbf{end\ if}\ \mathbf{end\ proc}$ **(5.7)**

plot([*seq*([*t*, *p*(*t*)], *t* = 0 ..12)], *style* = *point*, *symbol* = *solidcircle*, *axes* = *boxed*) # *... la crescita è ancora più veloce*

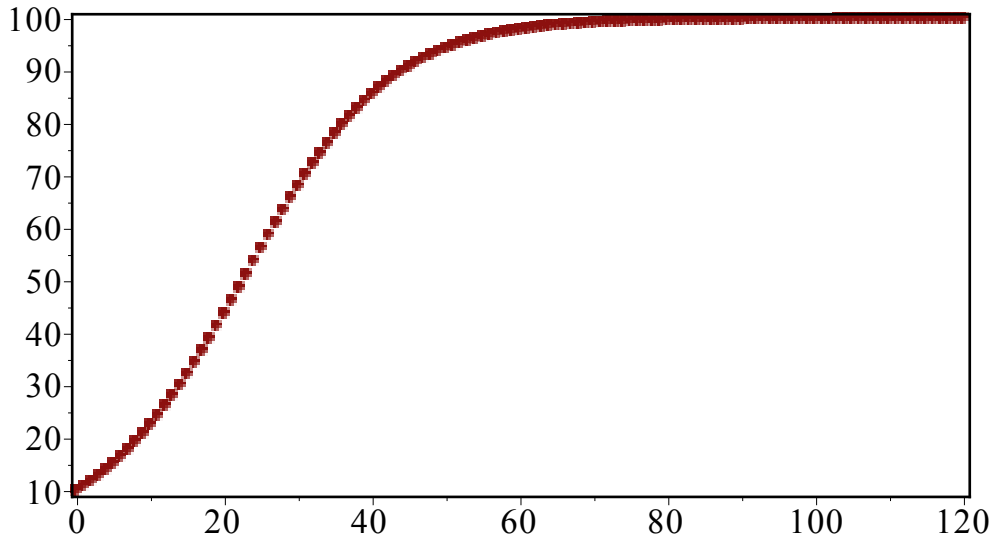


```
CrescitaLimitata := # questa procedura definisce una funzione p che esprime l'andamento della popolazione di anno in anno
 proc(i, m, c)      # con popolazione iniziale i e tasso di crescita teorica c, assumendo un valore massimo ottimale m per la popolazione
  global p;
  p := proc(t :: nonnegint)
   option remember;
   if t = 0 then i else (1 + c (1 - p(t - 1)/m)) p(t - 1) end if   # qui il fattore c è smorzato quando la popolazione tende a m
  end proc;
  return p = eval(p)
end proc :

CrescitaLimitata(10, 100, 0.1)   # modello con popolazione iniziale 10 e ottimale 100, e con tasso di crescita teorica del 10%
```

$$p = \mathbf{proc}(t::nonnegint)\ \mathbf{option}\ remember;\ \mathbf{if}\ t = 0\ \mathbf{then}\ 10\ \mathbf{else}\ (1 + 0.1 * (1 - p(t-1)/100)) * p(t-1)\ \mathbf{end\ if}\ \mathbf{end\ proc} \qquad \textbf{(5.8)}$$

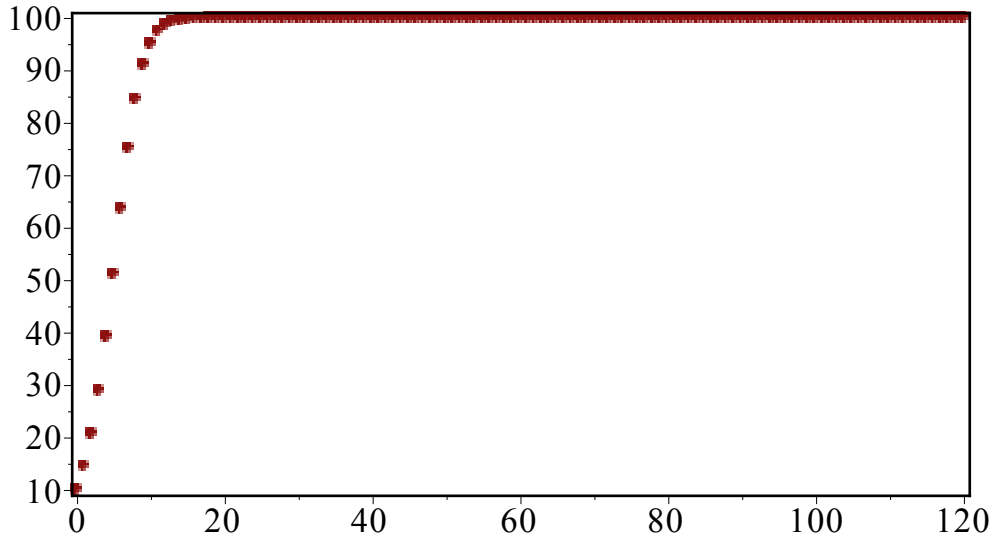
```
plot([seq([t, p(t)], t = 0 ..120)], style = point, symbol = solidcircle, axes = boxed)   # la popolazione tende gradualmente al valore ottimale
```



```
CrescitaLimitata(10, 100, 0.5)  # aumentando il tasso c al 50% ...
```

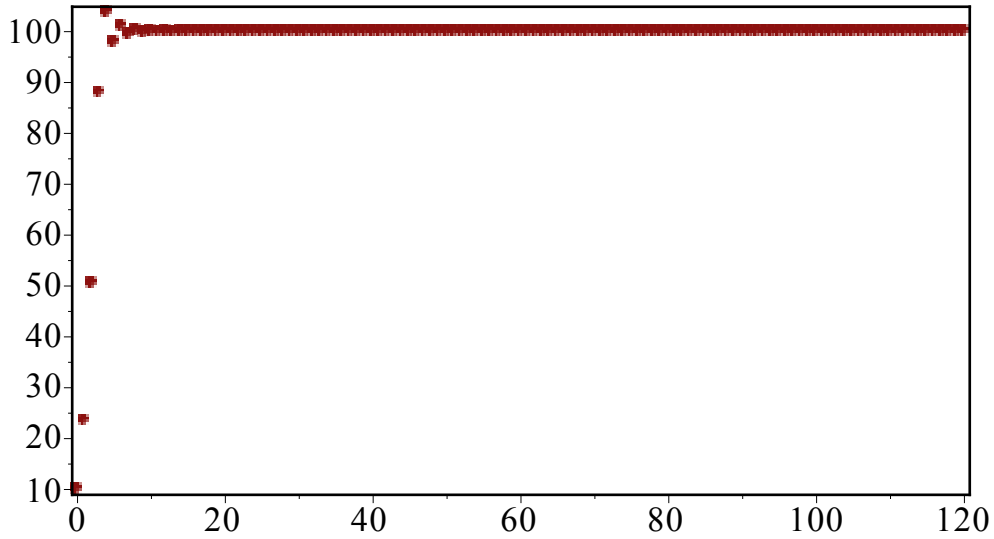
$$p := \mathbf{proc}(t::nonnegint)\ \mathbf{option}\ remember;\ \mathbf{if}\ t = 0\ \mathbf{then}\ 10\ \mathbf{else}\ (1 + 0.5 * (1 - p(t-1)/100)) * p(t-1)\ \mathbf{end\ if}\ \mathbf{end\ proc} \qquad \textbf{(5.9)}$$

```
plot([seq([t, p(t)], t = 0 ..120)], style = point, symbol = solidcircle, axes = boxed)  # ... la popolazione si tende al valore ottimale più rapidamente
```



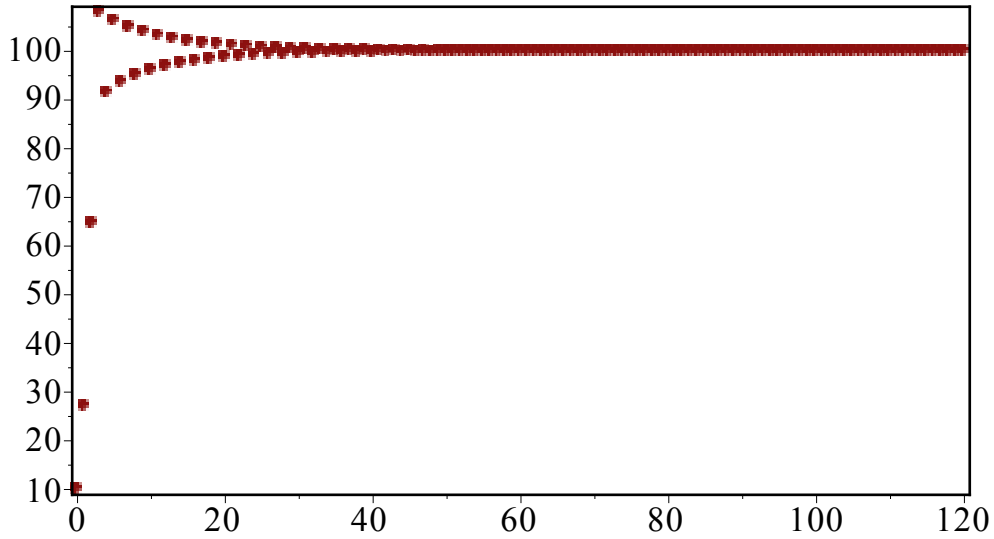
```
CrescitaLimitata(10, 100, 1.5)  # con un tasso del 150% ...
    p = proc(t::nonnegint) option remember; if t = 0 then 10 else (1 + 1.5 * (1 - p(t - 1) / 100)) * p(t - 1) end if end proc
plot([seq([t, p(t)], t = 0 ..120)], style = point, symbol = solidcircle, axes = boxed)  # ... si ha un'oscillazione smorzata intorno al valore ottimale
```

(5.10)



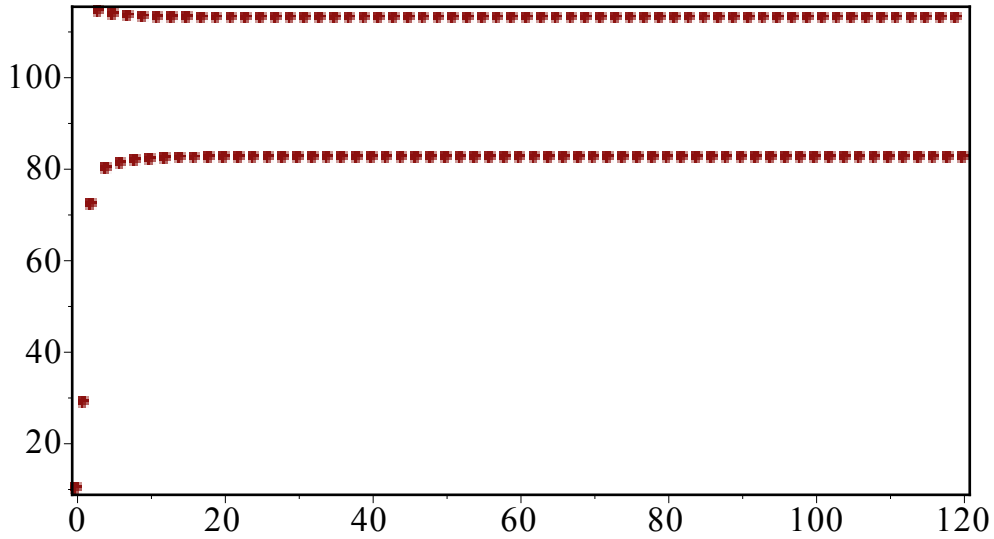
```
CrescitaLimitata(10, 100, 1.9)  # al 190% ...
    p = proc(t::nonnegint) option remember; if t = 0 then 10 else (1 + 1.9 * (1 - p(t - 1) / 100)) * p(t - 1) end if end proc
plot([seq([t, p(t)], t = 0 ..120)], style = point, symbol = solidcircle, axes = boxed)  # ... l'oscillazione diventa più ampia
```

(5.11)



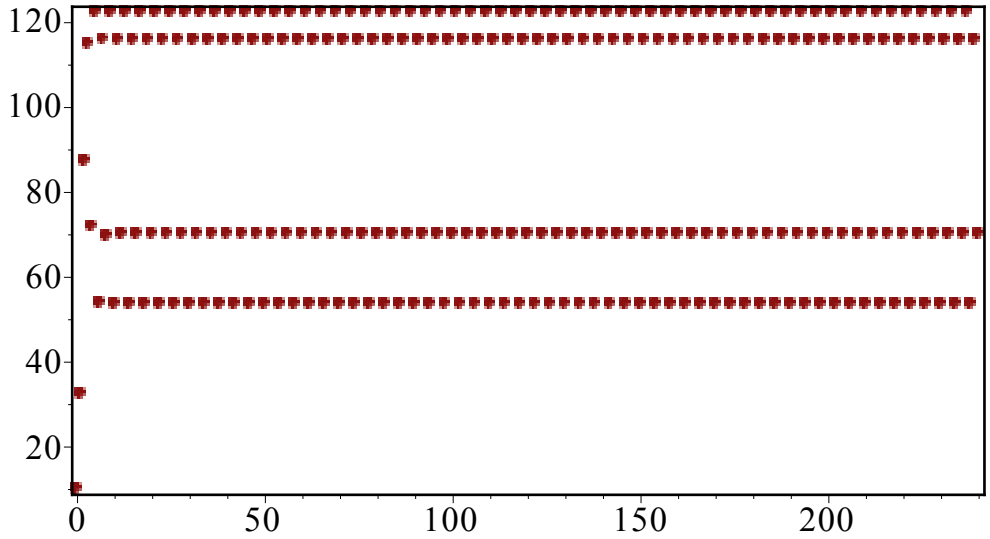
```
CrescitaLimitata(10, 100, 2.1)  # al 210% ...
    p = proc(t::nonnegint) option remember; if t = 0 then 10 else (1 + 2.1 * (1 - p(t - 1) / 100)) * p(t - 1) end if end proc
plot([seq([t, p(t)], t = 0 ..120)], style = point, symbol = solidcircle, axes = boxed)  # l'oscillazione si stabilizza tra due valori
```

(5.12)



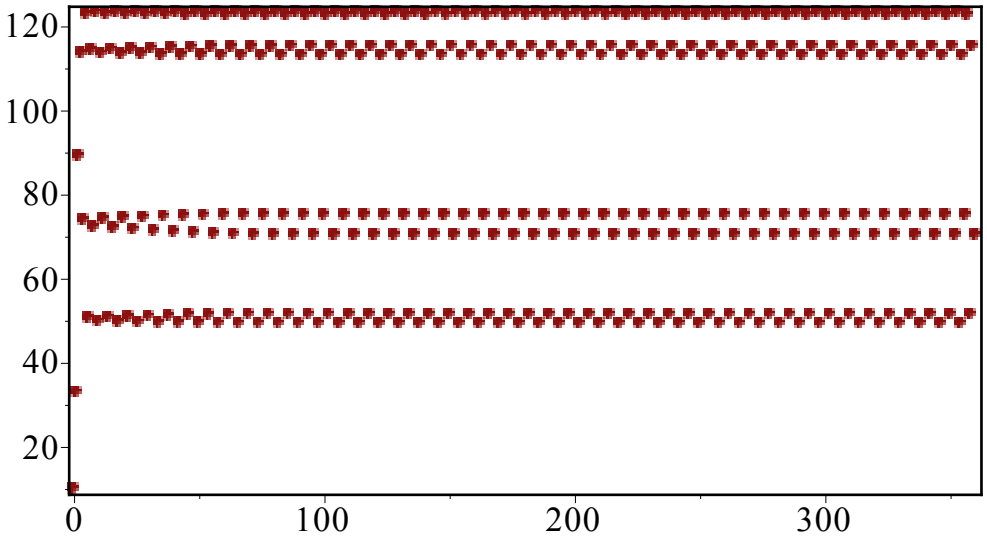
```
CrescitaLimitata(10, 100, 2.5)   # al 250% ...
    p = proc(t::nonnegint) option remember; if t = 0 then 10 else (1 + 2.5 * (1 − p(t − 1) / 100)) * p(t − 1) end if end proc
plot([seq([t, p(t)], t = 0 ..240)], style = point, symbol = solidcircle, axes = boxed)   # ... i valori limite diventano quattro
```

(5.13)



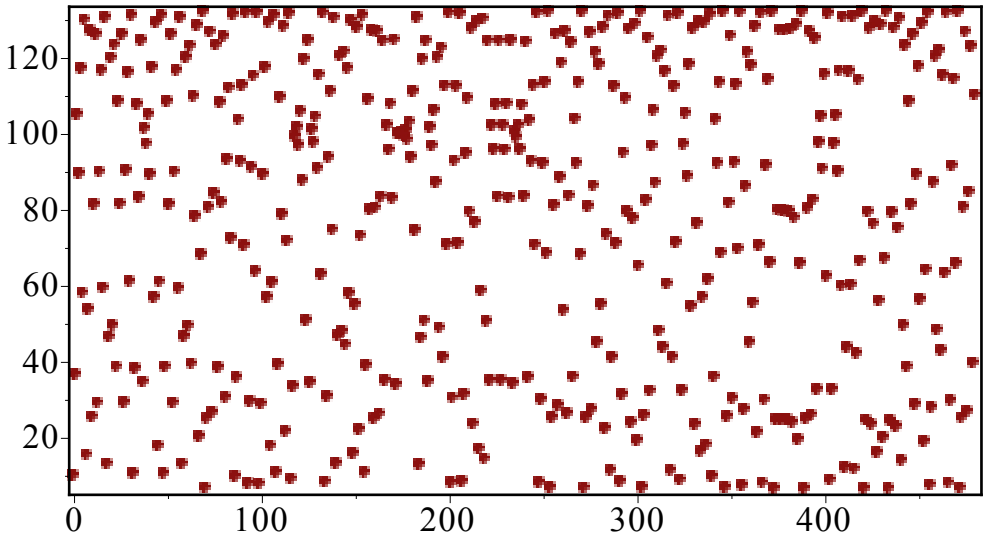
```
CrescitaLimitata(10, 100, 2.55)   # al 256% ...
    p = proc(t::nonnegint) option remember; if t = 0 then 10 else (1 + 2.55 * (1 - p(t - 1) / 100)) * p(t - 1) end if end proc
plot([seq([t, p(t)], t = 0 ..360)], style = point, symbol = solidcircle, axes = boxed)   # ... diventano otto
```

(5.14)



```
CrescitaLimitata(10, 100, 2.95)   # al 295% ...
    p = proc(t::nonnegint) option remember; if t = 0 then 10 else (1 + 2.95 * (1 - p(t - 1) / 100)) * p(t - 1) end if end proc
plot([seq([t, p(t)], t = 0 ..480)], style = point, symbol = solidcircle, axes = boxed)   # ... la situazione appare caotica
```

(5.15)

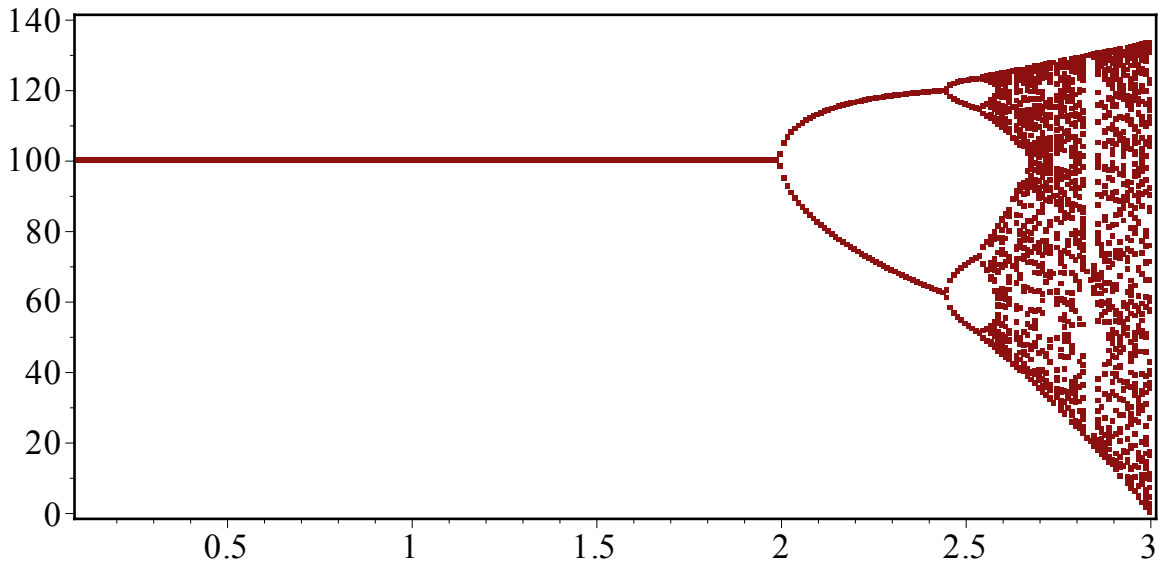


```
ValoriLimite :=
proc(i, m, c, n, T)   # procedura per stimare i valori limite
  CrescitaLimitata(i, m, c);
  return op(-n..-1, [seq([c, p(t)], t = 0..T)])   # si calcola la popolazione nei primi T anni e si considetano solo gli ultimi n valori
end proc
```

```
proc(i, m, c, n, T) CrescitaLimitata(i, m, c); return op(-n..-1, [seq([c, p(t)], t = 0..T)]) end proc
```

(5.16)

```
plot([seq(ValoriLimite(10, 100, c, 50, 500), c = 0..3, 0.01)],
     style = point, symbol = point, view = [0.1..3, 0..140], axes = boxed)   # ecco un grafico dei valori limite in funzione del tasso
```



Lanci

$x''(t) = a$ # *equazione del moto uniformemente accelerato*

$$\mathrm{D}^{(2)}(x)(t) = a \tag{6.1}$$

$x'(0) = v0, x(0) = x0$ # *valori iniziali*

$$\mathrm{D}(x)(0) = v0, x(0) = x0 \tag{6.2}$$

dsolve$(\{\%\%, \%\}, x(t))$ # *otteniamo la legge (formale) del moto ...*

$$x(t) = \frac{1}{2} a t^2 + v0\, t + x0 \tag{6.3}$$

subs$(a = \langle 0, -g\rangle, x0 = \langle 0, 0\rangle, v0 = v\,\langle \cos(a), \sin(a)\rangle, \%)$ # *... e la legge del modo relativa al lancio di un oggetto in termini vettoriali*

$$x(t) = \frac{1}{2}\begin{bmatrix} 0 \\ -g \end{bmatrix} t^2 + \begin{bmatrix} v\cos(a) \\ v\sin(a) \end{bmatrix} t + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{6.4}$$

expand$(\%)$

$$x(t) = \begin{bmatrix} t\,v\cos(a) \\ -\frac{1}{2} t^2 g + t\,v\sin(a) \end{bmatrix} \tag{6.5}$$

define$(x, subs(x(t) = x(t :: anything), \%))$ # *trasformiamo l'equazione sopra in una definizione*

$x(t)$

$$\begin{bmatrix} t\,v\cos(a) \\ -\frac{1}{2} t^2 g + t\,v\sin(a) \end{bmatrix} \tag{6.6}$$

solve$(x(t)[2] = 0, t)$ # *per quali tempi di ha quota zero?*

$$0, \frac{2\,v\sin(a)}{g} \tag{6.7}$$

$T := \%[2]$ # *ovviamente ci interessa solo il secondo*

$$\frac{2\,v\sin(a)}{g} \tag{6.8}$$

$x(T)[1]$ # ecco la distanza raggiunta in funzione dell'angolo a e della velocità iniziale v

$$\frac{2\,v^2\sin(a)\cos(a)}{g} \tag{6.9}$$

$diff(\%, a)$ # ... e la sua derivata rispetto ad a,

$$\frac{2\,v^2\cos(a)^2}{g} - \frac{2\,v^2\sin(a)^2}{g} \tag{6.10}$$

$solve(\% = 0, a)$ # ... che, indipendentemente dalla velocità iniziale, si annulla per:

$$\frac{1}{4}\pi,\ \frac{3}{4}\pi \tag{6.11}$$

$subs\left(a = \frac{\pi}{4}, x(T)\right)$ # questa è la distanza massima raggiunta in funzione della velocità iniziale v

$$\begin{bmatrix} \frac{v^2}{g} \\ 0 \end{bmatrix} \tag{6.12}$$

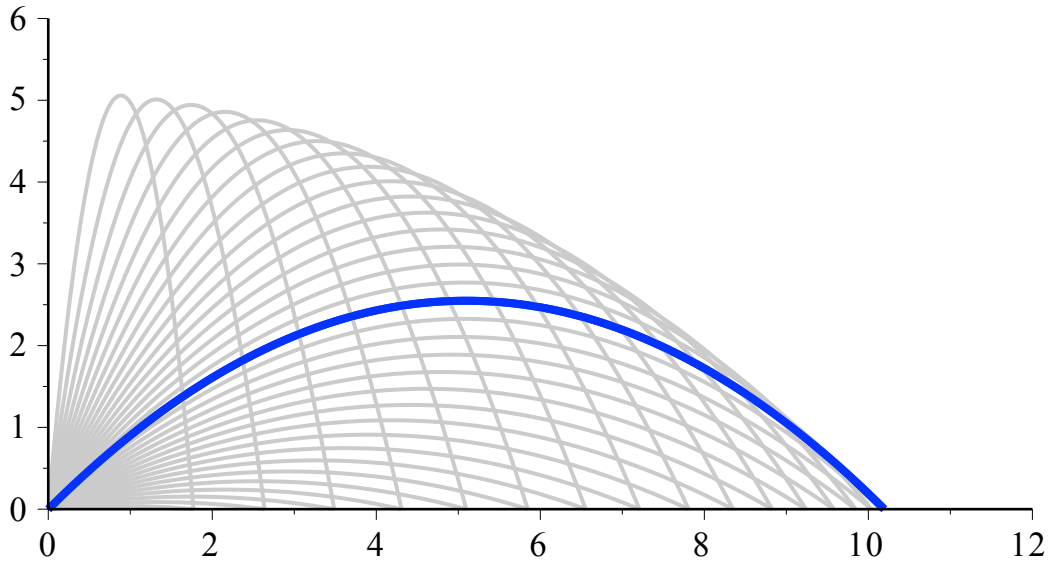
$subs(v = 10.0, g = 9.81, [op(convert(x(t), list)), t = 0 .. T])$ # fissiamo i valori di m, v e g

$$[10.0\,t\cos(a),\ -4.905000000\,t^2 + 10.0\,t\sin(a),\ t = 0..2.038735984\sin(a)]$$

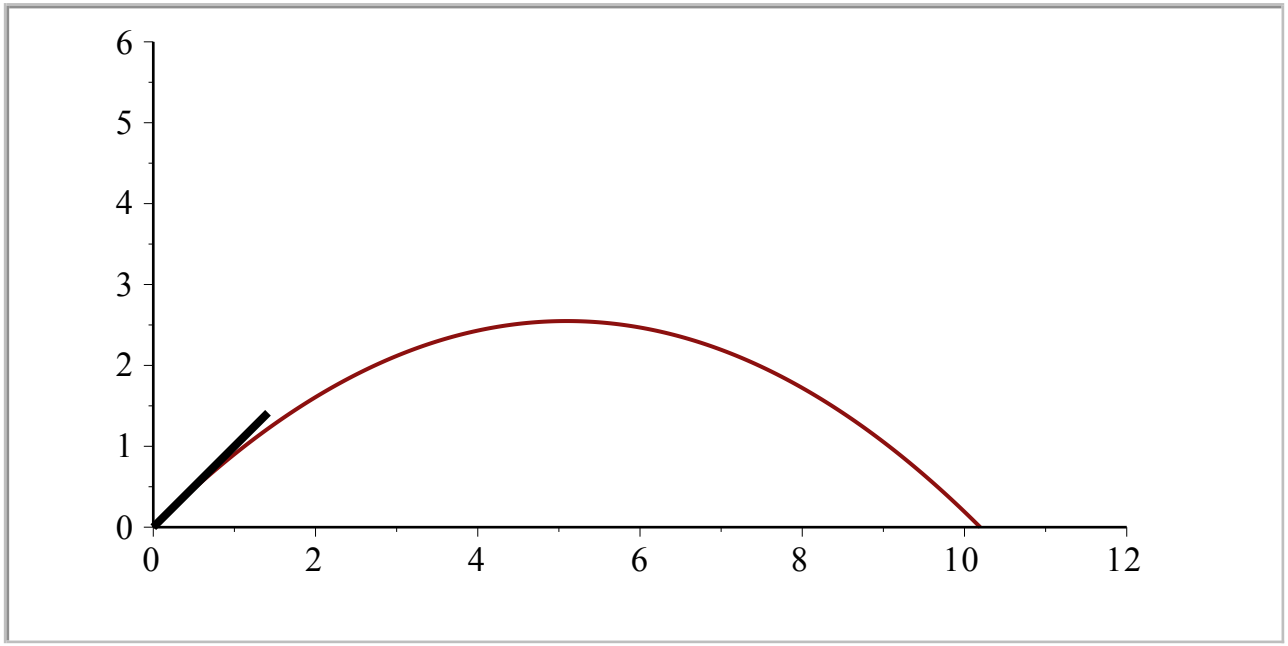
$subs\left(a = d\,\frac{evalf(\pi)}{180}, \%\right)$ # esprimiamo l'angolo a in gradi

$$[10.0\,t\cos(0.01745329252\,d),\ -4.905000000\,t^2 + 10.0\,t\sin(0.01745329252\,d),\ t = 0..2.038735984\sin(0.01745329252\,d)] \tag{6.14}$$

```
plot( [seq(%, d = 5 ..85, 2.5), subs(d = 45, %) ],
      view = [0 ..12 , 0 ..6 ], scaling = constrained,
      color = [gray $ 16, blue], thickness = [1 $ 16, 3 ])   # ecco le traiettorie per angoli da 5 a 85 gradi (procedendo di 5 in 5)
```



Math App

Angolo	*Velocità*	*Traiettoria*
		
45.0	10.0	*Distanza* 10.194