Regressione sinusoidale (se si vuole: regressione trigonometrica)

Dato l'insieme $S = \{(x_1, y_1); (x_2, y_2); \dots; (x_m, y_m)\}$,
si vuole trovare una funzione del tipo $y = B + A\cos(\omega x + \varphi)$ (1)
passante fra i punti che sia la migliore possibile applicando
il metodo dei minimi quadrati.

Applichiamo a (1) le formule di addizione:

$A\cos(\omega x + \varphi) = A\,\text{sen}(\omega x)\cos\varphi + A\cos(\omega x)\,\text{sen}\,\varphi$

Si pone $A\cos\varphi = a$, $A\,\text{sen}\,\varphi = b$ [1]; $B = c$. Pertanto la (1)
diventa $y = a\,\text{sen}(\omega x) + b\cos(\omega x) + c$

La funzione dei quadrati degli scarti, che in generale è:
$F = \sum (\hat{y}_k - y_k)^2$ nel nostro caso diventa:

$$F(a, b, c, \omega) = \sum_1^m \left[a\,\text{sen}(\omega x_k) + b\cos(\omega x_k) + c - y_k\right]^2$$

Le condizioni (necessarie) da imporre per il minimo di F sono:

$$\frac{\partial F}{\partial a} = 0 \;\wedge\; \frac{\partial F}{\partial b} = 0 \;\wedge\; \frac{\partial F}{\partial c} = 0 \;\wedge\; \frac{\partial F}{\partial \omega} = 0.$$

Calcoliamo queste derivate parziali (da qui in avanti $\sum_1^m \dots = \sum \dots$)

$$\frac{\partial F}{\partial a} = 2\sum \left[a\,\text{sen}(\omega x_k) + b\cos(\omega x_k) + c - y_k\right] \cdot \text{sen}(\omega x_k)$$

$$\frac{\partial F}{\partial b} = 2\sum \left[a\,\text{sen}(\omega x_k) + b\cos(\omega x_k) + c - y_k\right] \cdot \cos(\omega x_k)$$

$$\frac{\partial F}{\partial c} = 2\sum \left[a\,\text{sen}(\omega x_k) + b\cos(\omega x_k) + c - y_k\right] \cdot 1$$

$$\frac{\partial F}{\partial \omega} = 2\sum \left[a\,\text{sen}(\omega x_k) + b\cos(\omega x_k) + c - y_k\right] \cdot$$
$$\cdot \left[a \cdot \cos(\omega x_k) \cdot x_k - b \cdot \text{sen}(\omega x_k) \cdot x_k\right]$$

---

(1) Noti a e b si ha: $\text{tg}\,\varphi = \frac{A\,\text{sen}\,\varphi}{A\cos\varphi} = \frac{b}{a} \to \varphi = \text{arctg}\,\frac{b}{a}$ in $\left]-\frac{\pi}{2}, \frac{\pi}{2}\right[$

$A^2\,\text{sen}^2\varphi + A^2\cos^2\varphi = b^2 + a^2 \to A = \sqrt{a^2 + b^2}$.

Scriviamo in modo leggermente più conveniente $\frac{\partial F}{\partial \omega}$:

$$\frac{1}{2}\frac{\partial F}{\partial \omega} = a^2 \sum x_k \operatorname{sen}(\omega x_k)\cos(\omega x_k) - ab\sum x_k \operatorname{sen}^2(\omega x_k) + ab\sum x_k \cos^2(\omega x_k) - b^2 \sum x_k \operatorname{sen}(\omega x_k)\cos(\omega x_k) + ac\sum x_k \cos \omega x_k - bc\sum x_k \operatorname{sen}(\omega x_k) - a\sum x_k y_k \cos(\omega x_k) + b\sum x_k y_k \operatorname{sen}(\omega x_k);$$ cioè:

$$\frac{1}{2}\frac{\partial F}{\partial \omega} = (a^2 - b^2)\sum x_k \operatorname{sen}(\omega x_k)\cos(\omega x_k) + 2ab\sum x_k \cos^2(\omega x_k) - ab\sum x_k + a\left[c\sum x_k \cos(\omega x_k) - \sum x_k y_k \cos(\omega x_k)\right] - b\left[c\sum x_k \operatorname{sen}(\omega x_k) - \sum x_k y_k \operatorname{sen}(\omega x_k)\right]$$

Si tratta in definitiva di risolvere il sistema non lineare nelle incognite a, b, c, ω seguente:

$$\begin{cases} \sum \left[a \operatorname{sen}(\omega x_k) + b\cos(\omega x_k) + c - y_k\right]\cdot \operatorname{sen}(\omega x_k) = 0 \\ \sum \left[a \operatorname{sen}(\omega x_k) + b\cos(\omega x_k) + c - y_k\right]\cdot \cos(\omega x_k) = 0 \\ \sum \left[a \operatorname{sen}(\omega x_k) + b\cos(\omega x_k) + c - y_k\right]\cdot 1 = 0 \\ \dfrac{\partial F}{\partial \omega} = 0 \end{cases}$$

Le prime tre equazioni si scrivono in modo più conveniente:

$$\begin{cases} a\sum \operatorname{sen}^2(\omega x_k) + b\sum \operatorname{sen}(\omega x_k)\cos(\omega x_k) + c\sum \operatorname{sen}(\omega x_k) = \sum y_k \operatorname{sen}(\omega x_k) \\ a\sum \operatorname{sen}(\omega x_k)\cos(\omega x_k) + b\sum \cos^2(\omega x_k) + c\sum \cos(\omega x_k) = \sum y_k \cos(\omega x_k) \\ a\sum \operatorname{sen}(\omega x_k) + b\sum \cos(\omega x_k) + c\cdot n \overset{(2)}{=} \sum y_k \\ \dfrac{\partial F(a,b,c,\omega)}{\partial \omega} = 0 \end{cases}$$

Questo sistema è non lineare nelle variabili a, b, c, ω; bisogna affrontare la sua risoluzione con metodi iterativi.

(2) $\sum_1^n c = c\cdot\sum_1^n 1 = c\cdot n$

Osserviamo intanto che le prime tre equazioni del sistema sono lineari nelle variabili a, b, c; esse quindi non presenterebbero difficoltà se fosse noto ω. L'idea per risolvere il problema sta proprio nell'assegnare un valore di "prova" a ω:

$$\omega = \omega_0$$

Risolvere il sistema formato dalle prime tre equazioni:

$$S_0 = \{a_0, b_0, c_0\}$$

e calcolare $\left(\frac{\partial F}{\partial \omega}\right)_{(a_0, b_0, c_0, \omega_0)} = F'_\omega(a_0, b_0, c_0, \omega_0)$

Eseguire il calcolo per diversi valori di ω per controllare la convergenza del metodo a un valore $\bar{\omega}_0$ in corrispondenza del quale il vettore $(\bar{a}_0, \bar{b}_0, \bar{c}_0, \bar{\omega}_0)$ genera una curva che ben si adatta ai dati. Individuato questo valore $\bar{\omega}_0$, si tratta di affinare la soluzione trovando $\bar{\omega}_{01}$ e $\bar{\omega}_{02}$ tale che $F'_\omega(\bar{a}_{01}, \bar{b}_{01}, \bar{c}_{01}, \bar{\omega}_{01}) \cdot F'_\omega(\bar{a}_{02}, \bar{b}_{02}, \bar{c}_{02}, \bar{\omega}_{02}) < 0$ e continuare iterativamente riducendo l'intervallo $[\bar{\omega}_{01}, \bar{\omega}_{02}]$ fino a quando non si raggiunge una precisione accettabile.

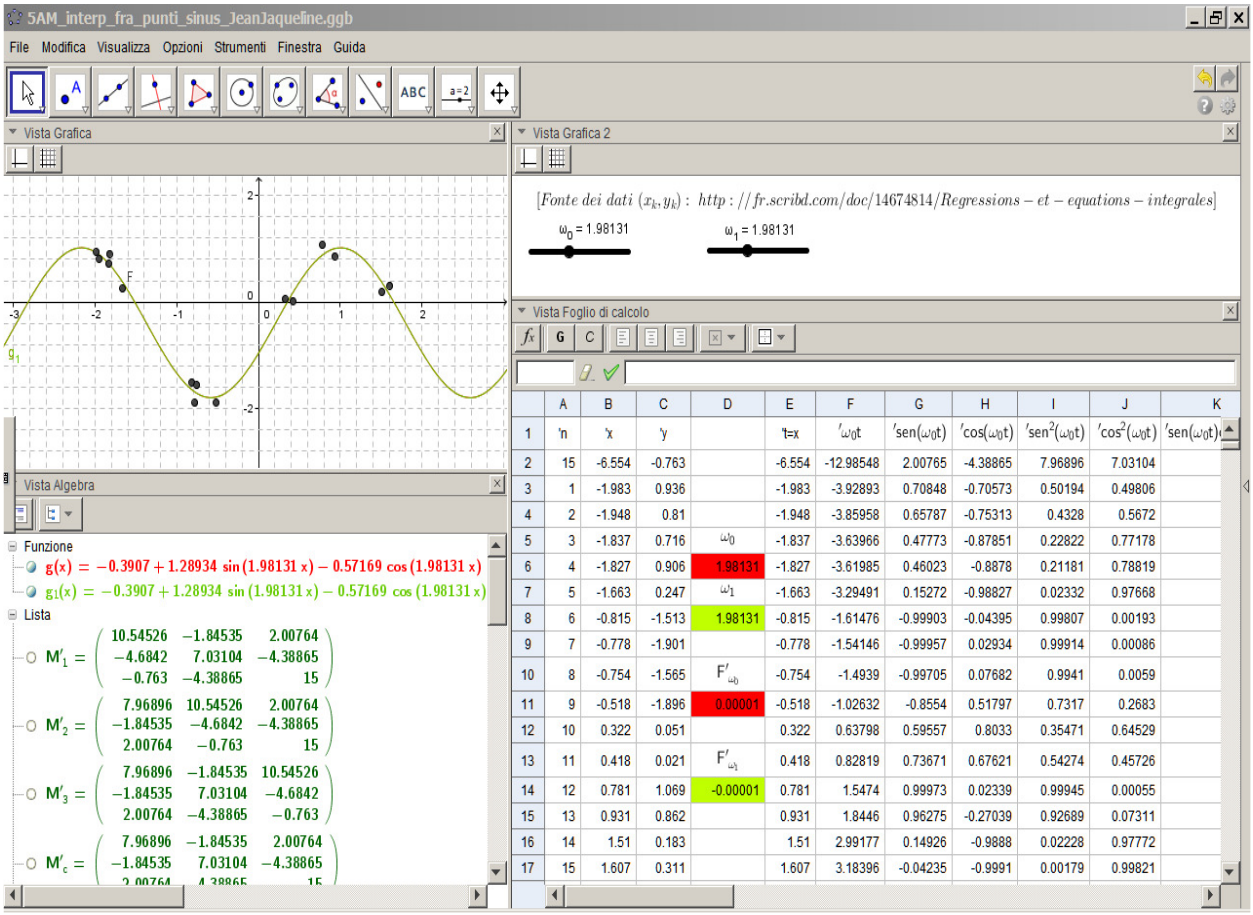
Con schema matriciale:

$$\begin{bmatrix} \sum \text{sen}^2(\omega_0 x_k) & \sum \text{sen}(\omega_0 x_k)\cos(\omega_0 x_k) & \sum \text{sen}(\omega_0 x_k) \\ \sum \text{sen}(\omega_0 x_k)\cos(\omega_0 x_k) & \sum \cos^2(\omega_0 x_k) & \sum \cos(\omega_0 x_k) \\ \sum \text{sen}(\omega_0 x_k) & \sum \cos(\omega_0 x_k) & n \end{bmatrix} \begin{Bmatrix} a \\ b \\ c \end{Bmatrix} = \begin{Bmatrix} \sum y_k \,\text{sen}(\omega_0 x_k) \\ \sum y_k \cos(\omega_0 x_k) \\ \sum y_k \end{Bmatrix}$$

(2)

Valutazione di $F'_\omega(a, b, c, \omega) = \left(\frac{\partial f}{\partial \omega}\right)$ in $(a_0, b_0, c_0, \omega_0)$

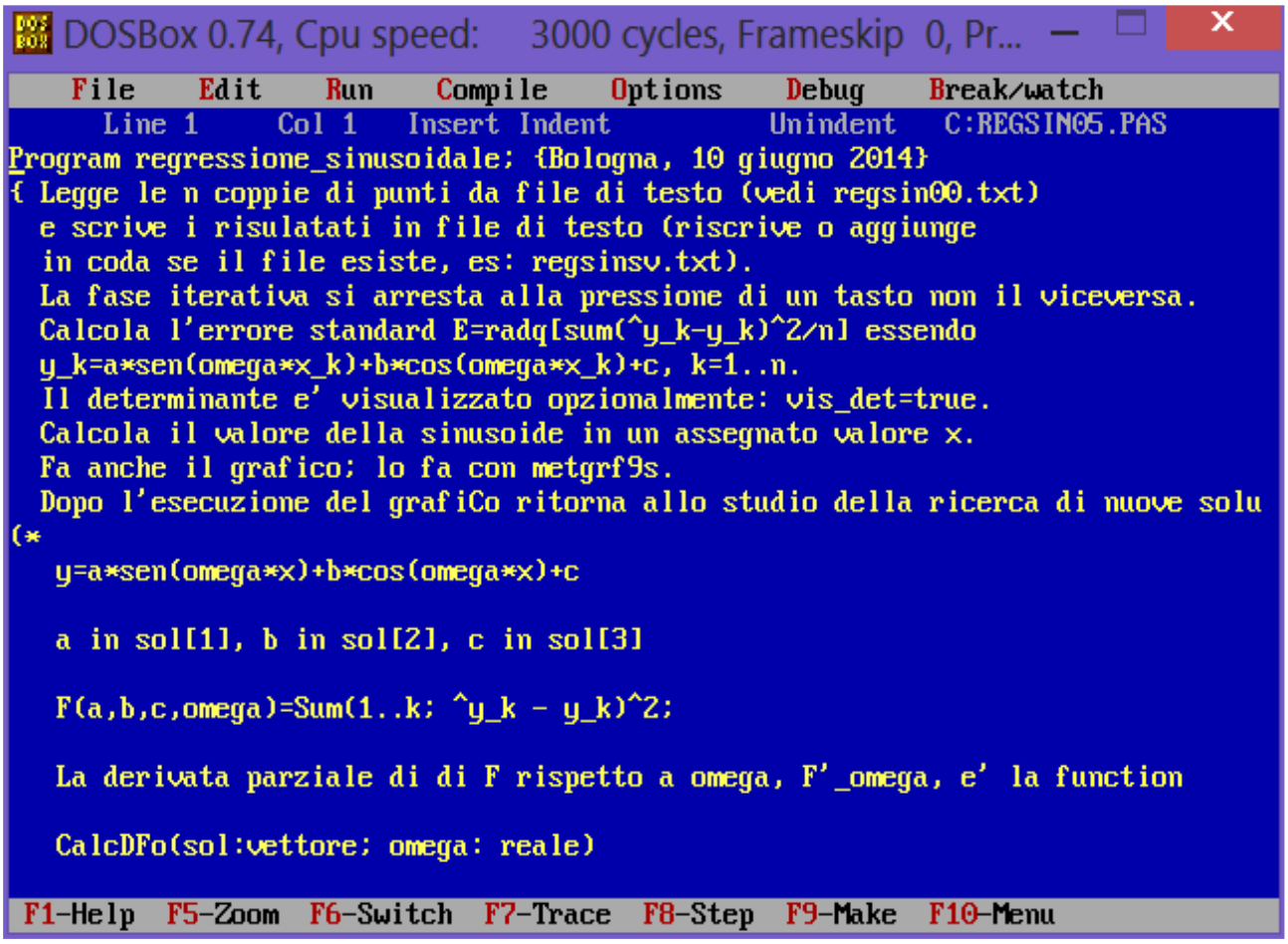
Il procedimento descritto è stato applicato nell'ambiente GEOGEBRA che permette di controllare graficamente il processo iterativo fino alla soluzione accettabile.[3]

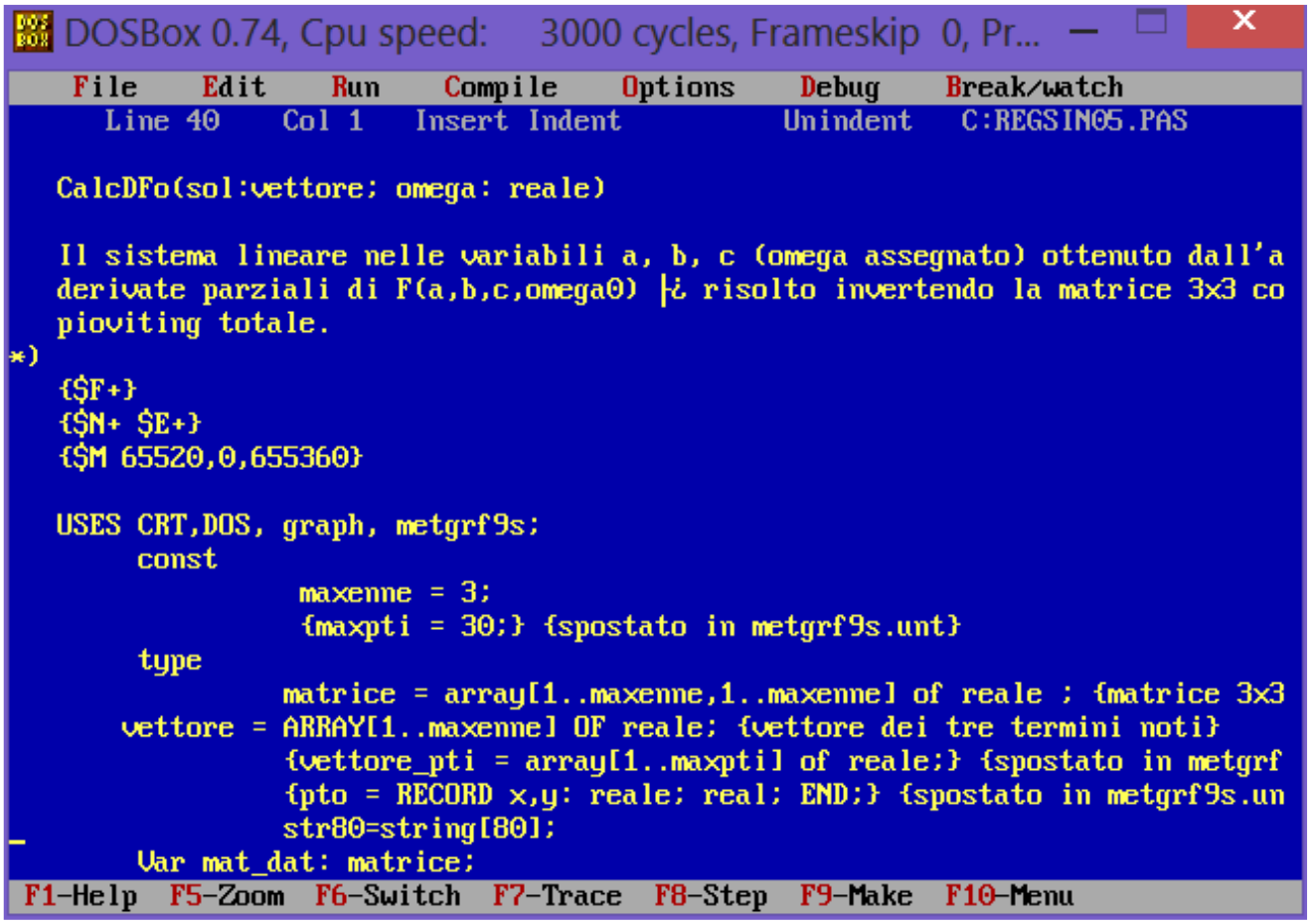


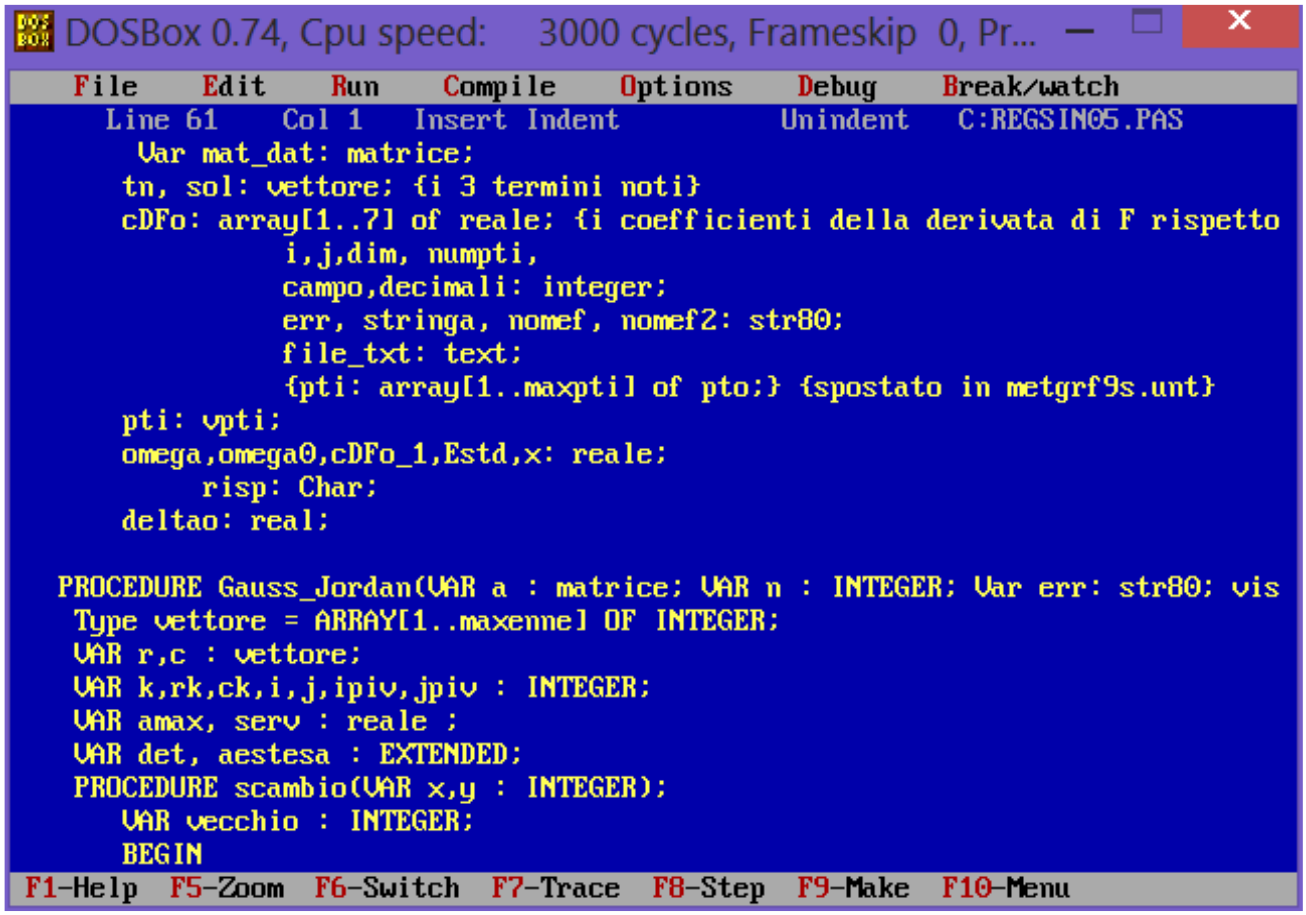
	A	B	C	D	E	F	G	H	I	J	K
1	'n	'x	'y		't=x	'ω_0t	'sen(ω_0t)	'cos(ω_0t)	'sen^2(ω_0t)	'cos^2(ω_0t)	'sen(ω_0t)(
2	15	-6.554	-0.763		-6.554	-12.98548	2.00765	-4.38865	7.96896	7.03104	
3	1	-1.983	0.936		-1.983	-3.92893	0.70848	-0.70573	0.50194	0.49806	
4	2	-1.948	0.81		-1.948	-3.85958	0.65787	-0.75313	0.4328	0.5672	
5	3	-1.837	0.716	ω_0	-1.837	-3.63966	0.47773	-0.87851	0.22822	0.77178	
6	4	-1.827	0.906	1.98131	-1.827	-3.61985	0.46023	-0.8878	0.21181	0.78819	
7	5	-1.663	0.247	ω_1	-1.663	-3.29491	0.15272	-0.98827	0.02332	0.97668	
8	6	-0.815	-1.513	1.98131	-0.815	-1.61476	-0.99903	-0.04395	0.99807	0.00193	
9	7	-0.778	-1.901		-0.778	-1.54146	-0.99957	0.02934	0.99914	0.00086	
10	8	-0.754	-1.565	F'_{ω_0}	-0.754	-1.4939	-0.99705	0.07682	0.9941	0.0059	
11	9	-0.518	-1.896	0.00001	-0.518	-1.02632	-0.8554	0.51797	0.7317	0.2683	
12	10	0.322	0.051		0.322	0.63798	0.59557	0.8033	0.35471	0.64529	
13	11	0.418	0.021	F'_{ω_1}	0.418	0.82819	0.73671	0.67621	0.54274	0.45726	
14	12	0.781	1.069	-0.00001	0.781	1.5474	0.99973	0.02339	0.99945	0.00055	
15	13	0.931	0.862		0.931	1.8446	0.96275	-0.27039	0.92689	0.07311	
16	14	1.51	0.183		1.51	2.99177	0.14926	-0.9888	0.02228	0.97772	
17	15	1.607	0.311		1.607	3.18396	-0.04235	-0.9991	0.00179	0.99821	

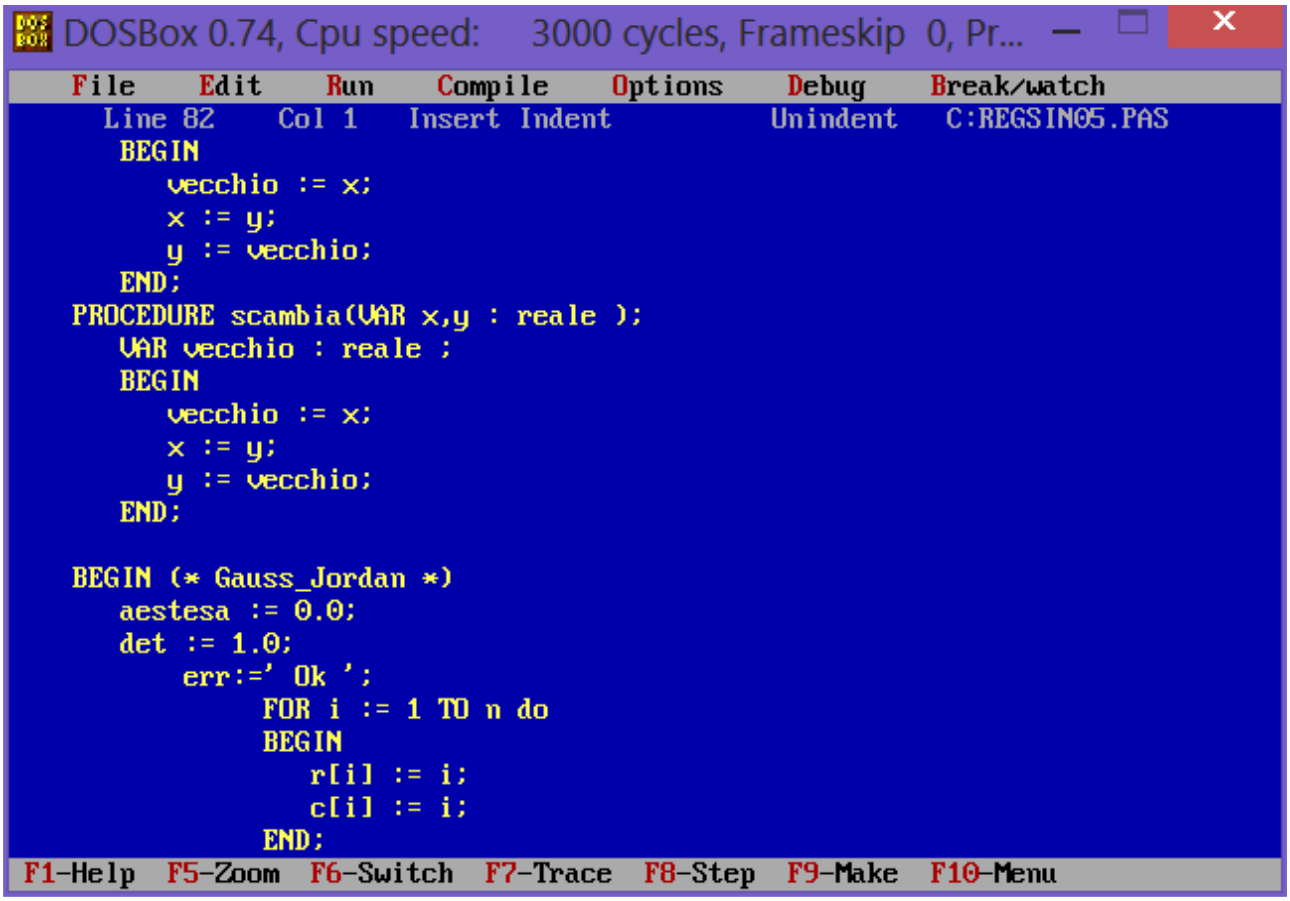
È stato poi scritto un programma in linguaggio Pascal nella implementazione "Turbo Pascal" che oggi gira nell'ambiente "Dos Box". Così facendo è stato possibile rintracciare gli strumenti analitici e grafici adoperati per l'interpolazione lineare, esponenziale, polinomiale scritti da me diversi anni fa (si cita in particolare il metodo di Gauss-Jordan con pivoting totale per invertire la matrice 3 x 3 nelle variabili a, b, c).

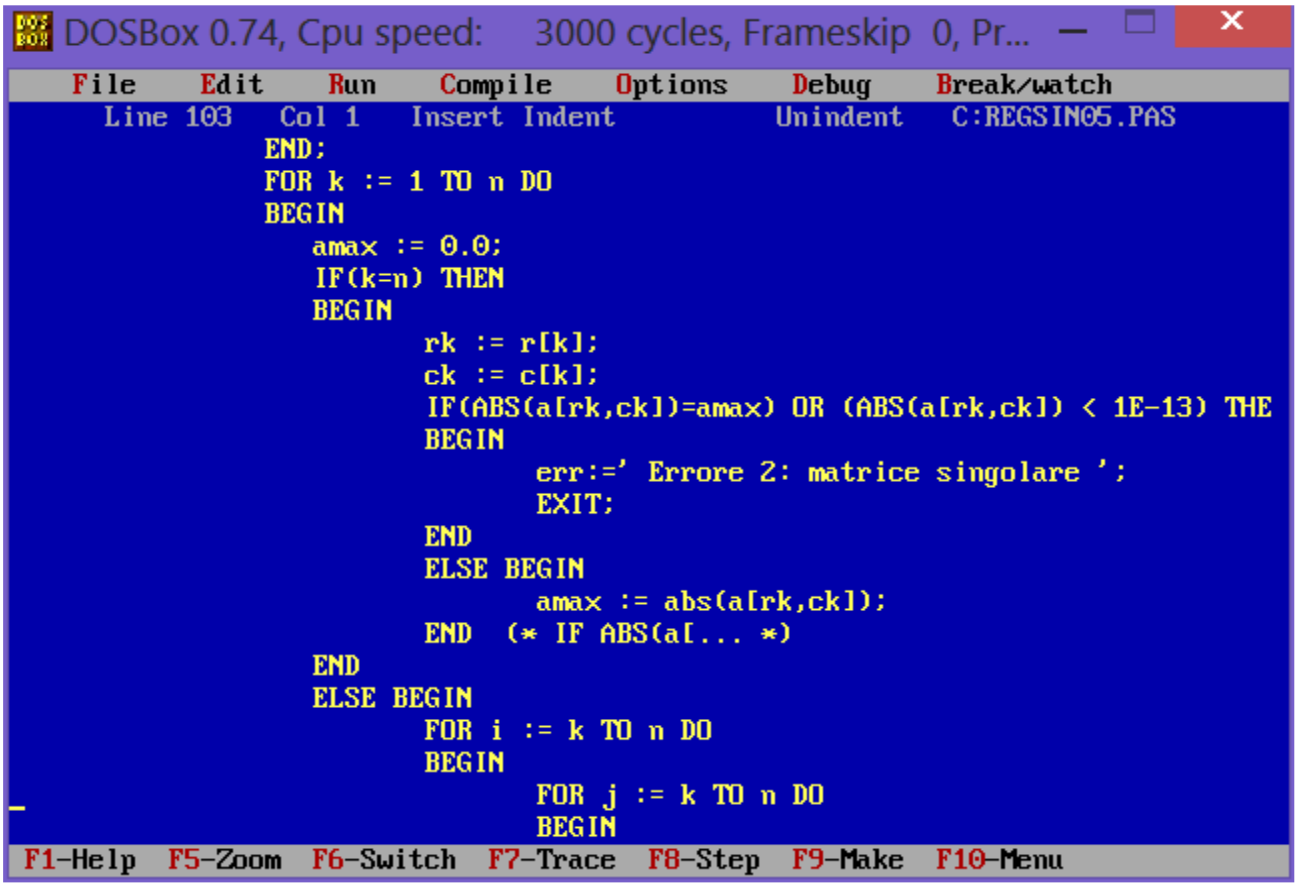
[3] GEOGEBRA possiede un comando che calcola la funzione $y = B + A\cos(\omega x + \varphi)$, esso è RegSin[<lista_punti>]. La bontà del procedimento indicato è pertanto facilmente controllabile.

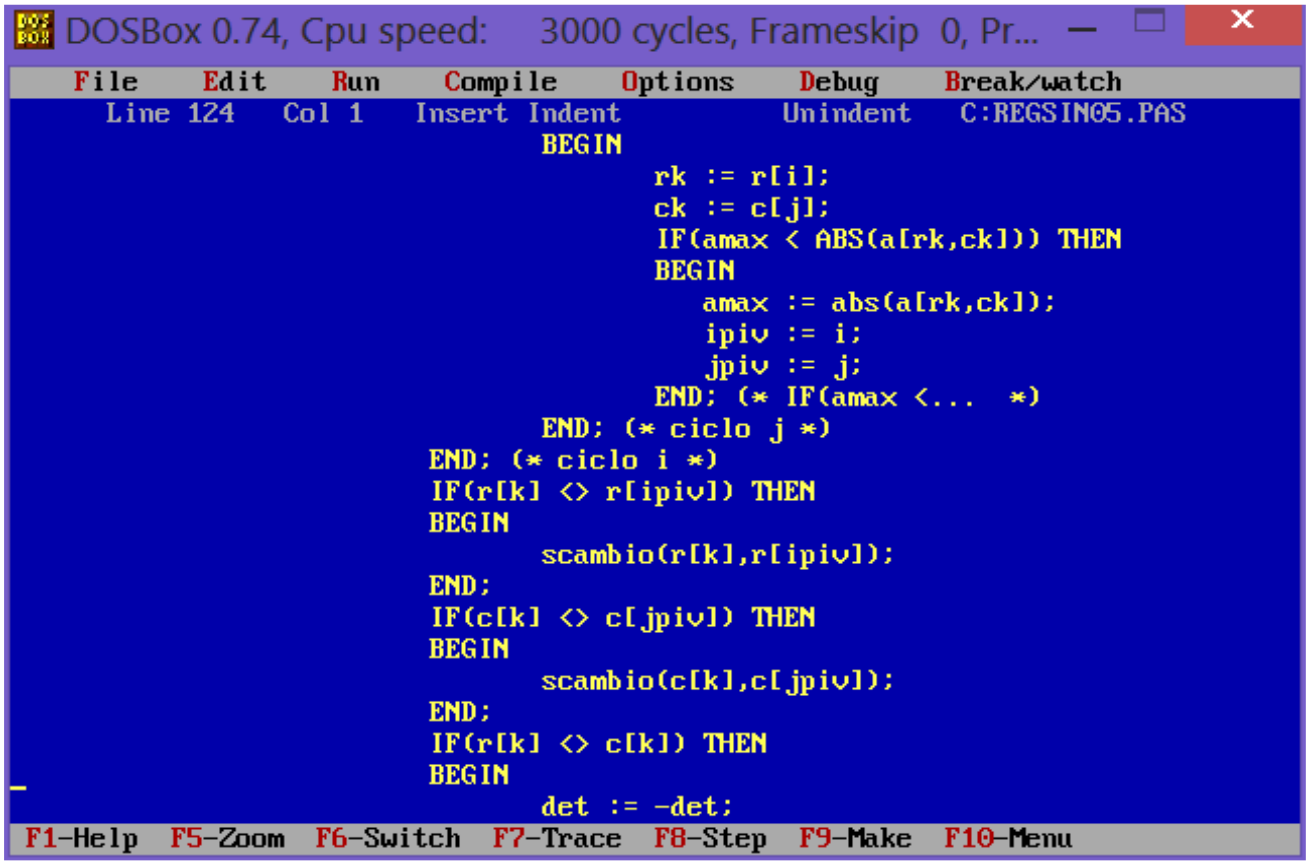

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pr...
File Edit Run Compile Options Debug Break/watch
Line 1 Col 1 Insert Indent Unindent C:REGSIN05.PAS
Program regressione_sinusoidale; {Bologna, 10 giugno 2014}
{ Legge le n coppie di punti da file di testo (vedi regsin00.txt)
e scrive i risulatati in file di testo (riscrive o aggiunge
in coda se il file esiste, es: regsinsv.txt).
La fase iterativa si arresta alla pressione di un tasto non il viceversa.
Calcola l'errore standard E=radq[sum(^y_k-y_k)^2/n] essendo
y_k=a*sen(omega*x_k)+b*cos(omega*x_k)+c, k=1..n.
Il determinante e' visualizzato opzionalmente: vis_det=true.
Calcola il valore della sinusoide in un assegnato valore x.
Fa anche il grafico; lo fa con metgrf9s.
Dopo l'esecuzione del grafiCo ritorna allo studio della ricerca di nuove solu
(*
y=a*sen(omega*x)+b*cos(omega*x)+c
a in sol[1], b in sol[2], c in sol[3]
F(a,b,c,omega)=Sum(1..k; ^y_k - y_k)^2;
La derivata parziale di di F rispetto a omega, F'_omega, e' la function
CalcDFo(sol:vettore; omega: reale)
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F10-Menu

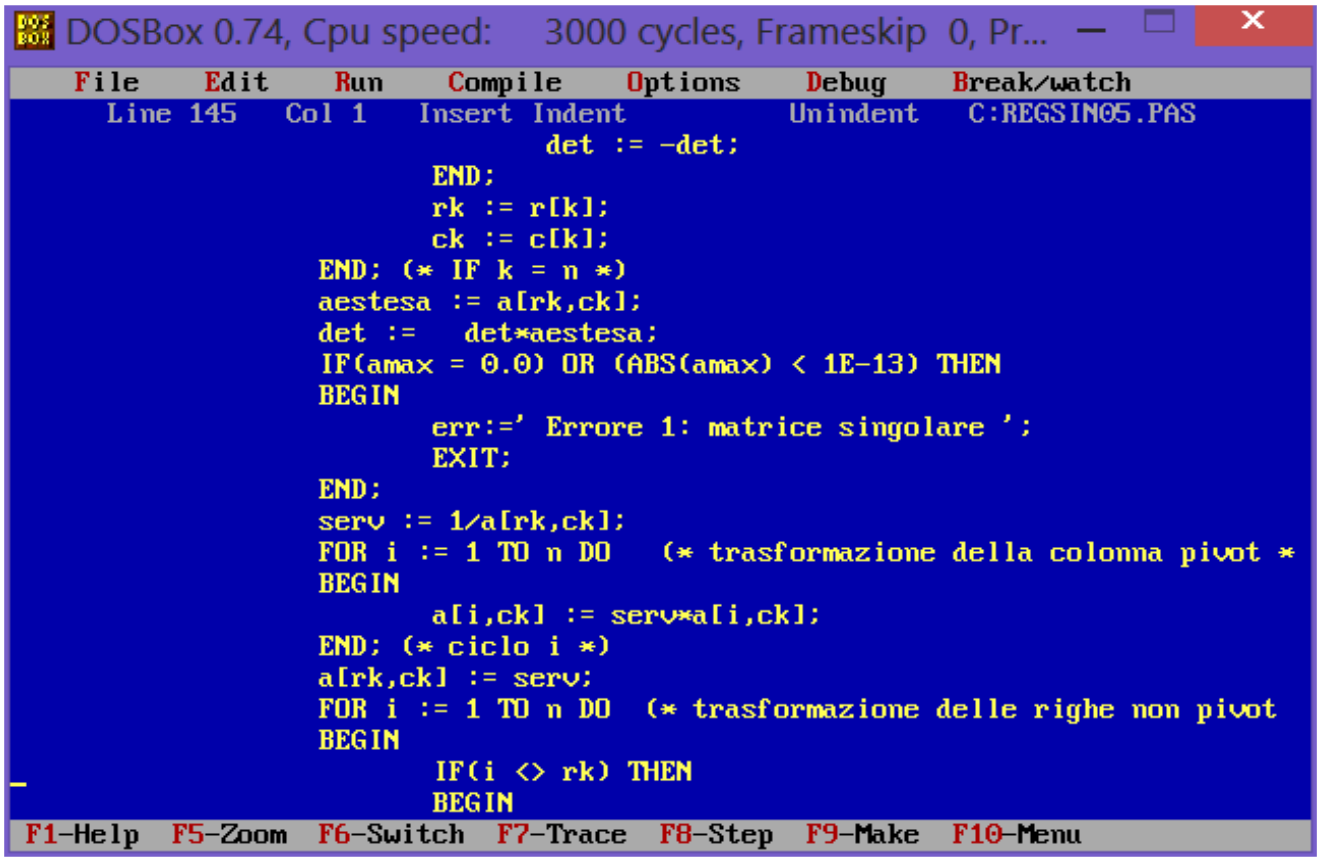

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pr...
File Edit Run Compile Options Debug Break/watch
Line 40 Col 1 Insert Indent Unindent C:REGSIN05.PAS
CalcDFo(sol:vettore; omega: reale)
Il sistema lineare nelle variabili a, b, c (omega assegnato) ottenuto dall'a
derivate parziali di F(a,b,c,omega0) è risolto invertendo la matrice 3x3 co
pioviting totale.
*)
{$F+}
{$N+ $E+}
{$M 65520,0,655360}
USES CRT,DOS, graph, metgrf9s;
const
maxenne = 3;
{maxpti = 30;} {spostato in metgrf9s.unt}
type
matrice = array[1..maxenne,1..maxenne] of reale ; {matrice 3x3
vettore = ARRAY[1..maxenne] OF reale; {vettore dei tre termini noti}
{vettore_pti = array[1..maxpti] of reale;} {spostato in metgrf
{pto = RECORD x,y: reale; real; END;} {spostato in metgrf9s.un
str80=string[80];
Var mat_dat: matrice;
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F10-Menu

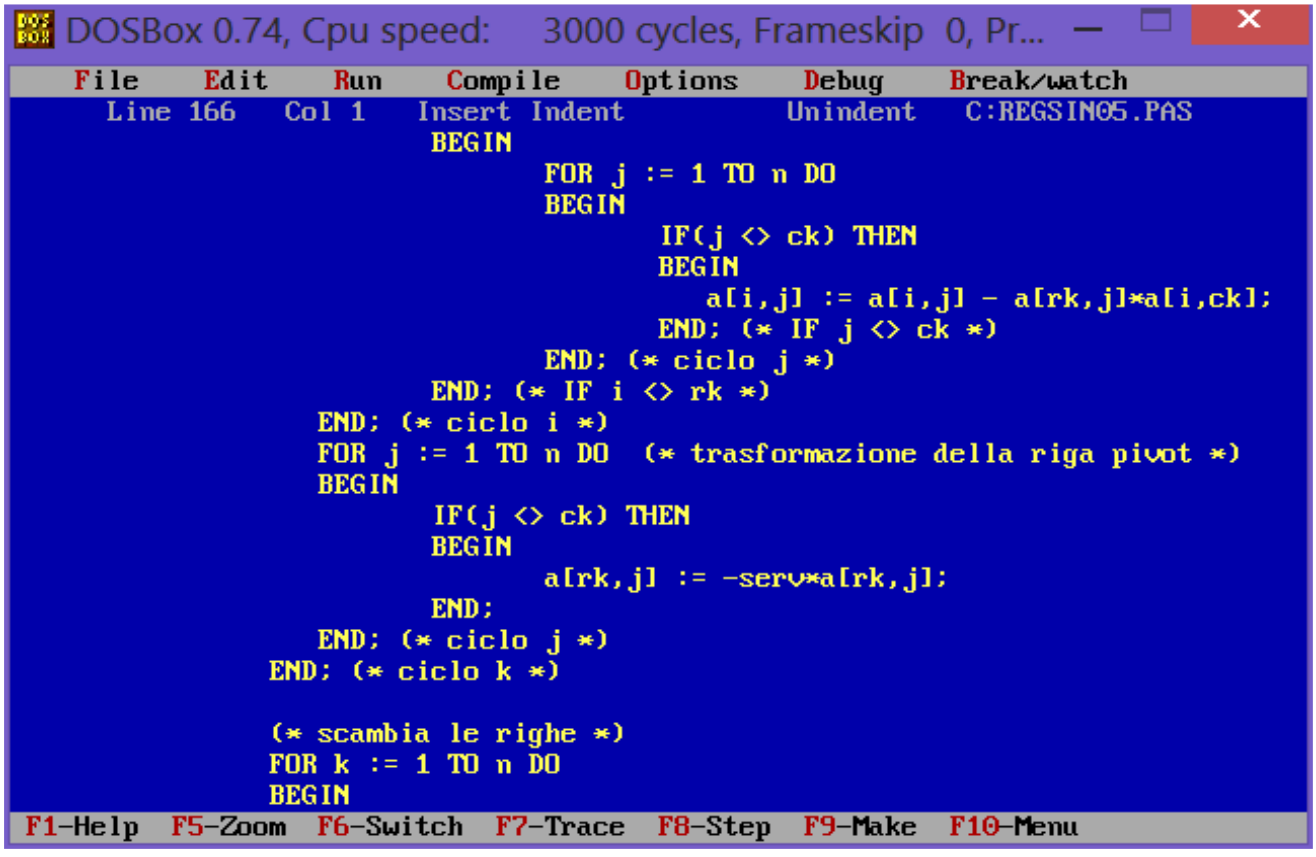

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pr...
File Edit Run Compile Options Debug Break/watch
Line 61 Col 1 Insert Indent Unindent C:REGSIN05.PAS
Var mat_dat: matrice;
tn, sol: vettore; {i 3 termini noti}
cDFo: array[1..7] of reale; {i coefficienti della derivata di F rispetto
i,j,dim, numpti,
campo,decimali: integer;
err, stringa, nomef, nomef2: str80;
file_txt: text;
{pti: array[1..maxpti] of pto;} {spostato in metgrf9s.unt}
pti: vpti;
omega,omega0,cDFo_1,Estd,x: reale;
risp: Char;
deltao: real;
PROCEDURE Gauss_Jordan(VAR a : matrice; VAR n : INTEGER; Var err: str80; vis
Type vettore = ARRAY[1..maxenne] OF INTEGER;
VAR r,c : vettore;
VAR k,rk,ck,i,j,ipiv,jpiv : INTEGER;
VAR amax, serv : reale ;
VAR det, aestesa : EXTENDED;
PROCEDURE scambio(VAR x,y : INTEGER);
VAR vecchio : INTEGER;
BEGIN
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F10-Menu

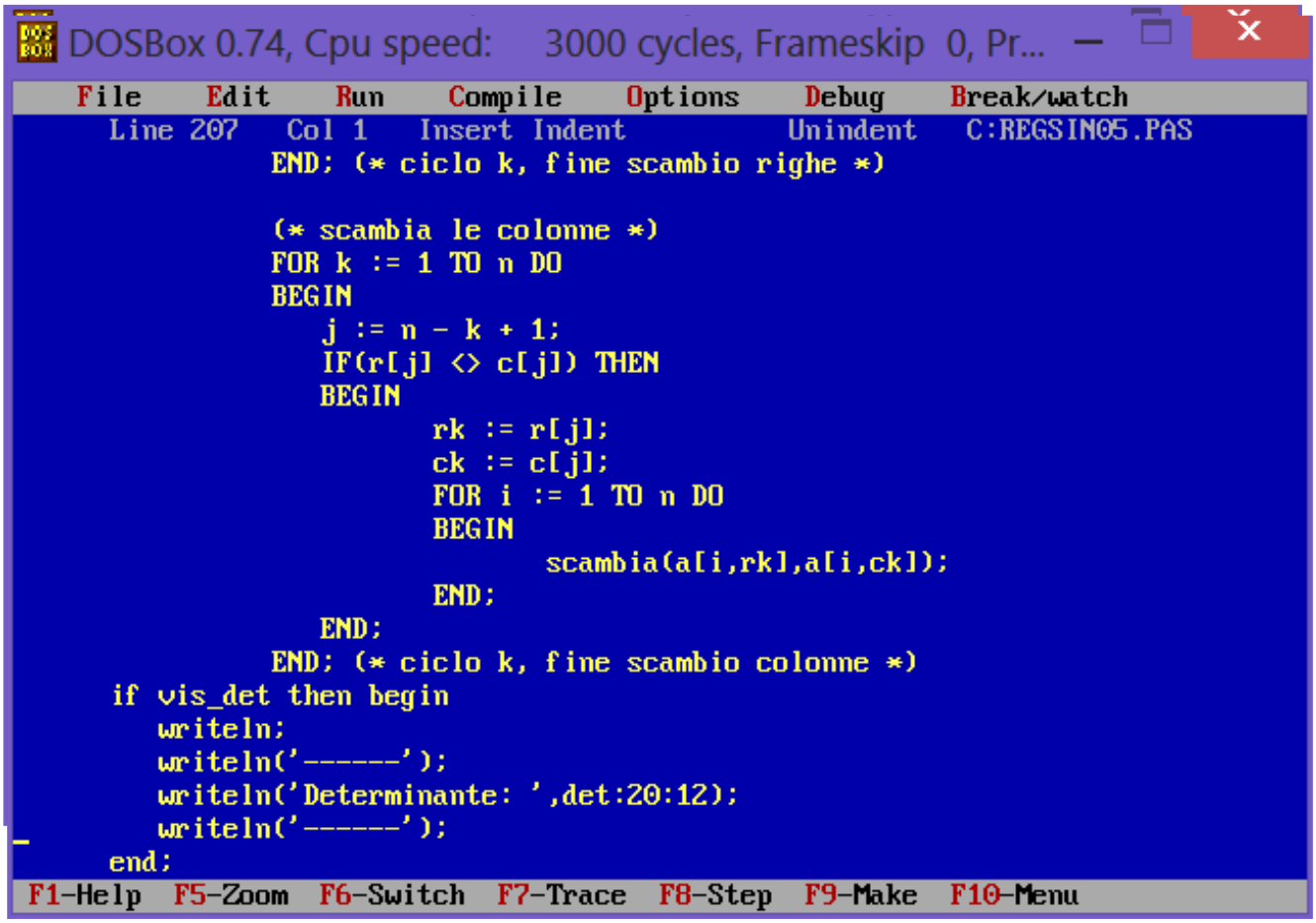

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pr...
File Edit Run Compile Options Debug Break/watch
Line 82 Col 1 Insert Indent Unindent C:REGSIN05.PAS
BEGIN
vecchio := x;
x := y;
y := vecchio;
END;
PROCEDURE scambia(VAR x,y : reale);
VAR vecchio : reale ;
BEGIN
vecchio := x;
x := y;
y := vecchio;
END;
BEGIN (* Gauss_Jordan *)
aestesa := 0.0;
det := 1.0;
err:=' Ok ';
FOR i := 1 TO n do
BEGIN
r[i] := i;
c[i] := i;
END;
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F10-Menu

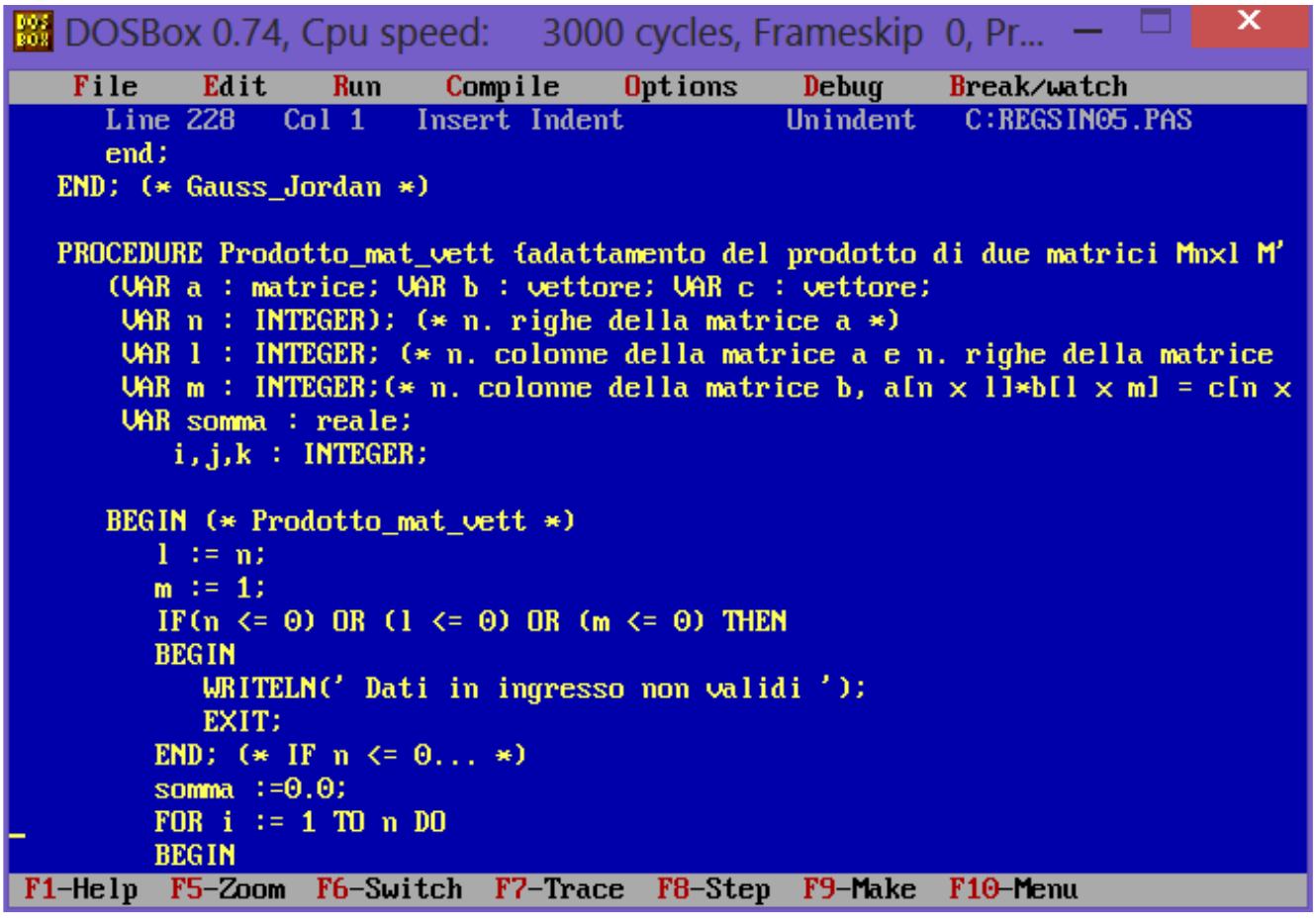

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pr...
File Edit Run Compile Options Debug Break/watch
Line 103 Col 1 Insert Indent Unindent C:REGSIN05.PAS
END;
FOR k := 1 TO n DO
BEGIN
amax := 0.0;
IF(k=n) THEN
BEGIN
rk := r[k];
ck := c[k];
IF(ABS(a[rk,ck])=amax) OR (ABS(a[rk,ck]) < 1E-13) THE
BEGIN
err:=' Errore 2: matrice singolare ';
EXIT;
END
ELSE BEGIN
amax := abs(a[rk,ck]);
END (* IF ABS(a[... *)
END
ELSE BEGIN
FOR i := k TO n DO
BEGIN
FOR j := k TO n DO
BEGIN
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F10-Menu


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pr...
File Edit Run Compile Options Debug Break/watch
Line 124 Col 1 Insert Indent Unindent C:REGSIN05.PAS
BEGIN
rk := r[i];
ck := c[j];
IF(amax < ABS(a[rk,ck])) THEN
BEGIN
amax := abs(a[rk,ck]);
ipiv := i;
jpiv := j;
END; (* IF(amax <... *)
END; (* ciclo j *)
END; (* ciclo i *)
IF(r[k] <> r[ipiv]) THEN
BEGIN
scambio(r[k],r[ipiv]);
END;
IF(c[k] <> c[jpiv]) THEN
BEGIN
scambio(c[k],c[jpiv]);
END;
IF(r[k] <> c[k]) THEN
BEGIN
det := -det;
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F10-Menu


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pr...
File Edit Run Compile Options Debug Break/watch
Line 145 Col 1 Insert Indent Unindent C:REGSIN05.PAS
det := -det;
END;
rk := r[k];
ck := c[k];
END; (* IF k = n *)
aestesa := a[rk,ck];
det := det*aestesa;
IF(amax = 0.0) OR (ABS(amax) < 1E-13) THEN
BEGIN
err:=' Errore 1: matrice singolare ';
EXIT;
END;
serv := 1/a[rk,ck];
FOR i := 1 TO n DO (* trasformazione della colonna pivot *
BEGIN
a[i,ck] := serv*a[i,ck];
END; (* ciclo i *)
a[rk,ck] := serv;
FOR i := 1 TO n DO (* trasformazione delle righe non pivot
BEGIN
IF(i <> rk) THEN
BEGIN
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F10-Menu


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pr...
File Edit Run Compile Options Debug Break/watch
Line 166 Col 1 Insert Indent Unindent C:REGSIN05.PAS
BEGIN
FOR j := 1 TO n DO
BEGIN
IF(j <> ck) THEN
BEGIN
a[i,j] := a[i,j] - a[rk,j]*a[i,ck];
END; (* IF j <> ck *)
END; (* ciclo j *)
END; (* IF i <> rk *)
END; (* ciclo i *)
FOR j := 1 TO n DO (* trasformazione della riga pivot *)
BEGIN
IF(j <> ck) THEN
BEGIN
a[rk,j] := -serv*a[rk,j];
END;
END; (* ciclo j *)
END; (* ciclo k *)
(* scambia le righe *)
FOR k := 1 TO n DO
BEGIN
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F10-Menu


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pr...
File Edit Run Compile Options Debug Break/watch
Line 207 Col 1 Insert Indent Unindent C:REGSIN05.PAS
END; (* ciclo k, fine scambio righe *)
(* scambia le colonne *)
FOR k := 1 TO n DO
BEGIN
j := n - k + 1;
IF(r[j] <> c[j]) THEN
BEGIN
rk := r[j];
ck := c[j];
FOR i := 1 TO n DO
BEGIN
scambia(a[i,rk],a[i,ck]);
END;
END;
END; (* ciclo k, fine scambio colonne *)
if vis_det then begin
writeln;
writeln('------');
writeln('Determinante: ',det:20:12);
writeln('------');
end;
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F10-Menu


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pr...
File Edit Run Compile Options Debug Break/watch
Line 228 Col 1 Insert Indent Unindent C:REGSIN05.PAS
end;
END; (* Gauss_Jordan *)
PROCEDURE Prodotto_mat_vett {adattamento del prodotto di due matrici Mnxl M'
(VAR a : matrice; VAR b : vettore; VAR c : vettore;
VAR n : INTEGER); (* n. righe della matrice a *)
VAR l : INTEGER; (* n. colonne della matrice a e n. righe della matrice
VAR m : INTEGER;(* n. colonne della matrice b, a[n x l]*b[l x m] = c[n x
VAR somma : reale;
i,j,k : INTEGER;
BEGIN (* Prodotto_mat_vett *)
l := n;
m := 1;
IF(n <= 0) OR (l <= 0) OR (m <= 0) THEN
BEGIN
WRITELN(' Dati in ingresso non validi ');
EXIT;
END; (* IF n <= 0... *)
somma :=0.0;
FOR i := 1 TO n DO
BEGIN
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F10-Menu

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pr...

File Edit Run Compile Options Debug Break/watch

Line 249 Col 1 Insert Indent Unindent C:REGSIN05.PAS

```
      BEGIN
            FOR k := 1 TO l DO
            BEGIN
               somma := somma + a[i,k]*b[k];
            END; (* ciclo k *)
            c[i] := somma;
            somma := 0.0;
      END; (* ciclo i *)
   END; (* Prodotto_mat_vett *)

FUNCTION Leggi_xy(nome:str80):BOOLEAN;
           {lettura delle coordinate da un file di testo}
VAR wy:INTEGER;

PROCEDURE PausaPag(riga:INTEGER);
BEGIN
   IF (WHEREY = 24) THEN BEGIN
      WRITE(' Premere un tasto per continuare ... '); READLN;
      WHILE WHEREY>riga DO BEGIN GOTOXY(1,WHEREY-1); CLREOL; END;
   END;
END;
```

F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F10-Menu

Per risparmiare spazio si omette il codice inerente alla lettura dei dati da file di testo e alla scrittura delle elaborazioni in file di testo.

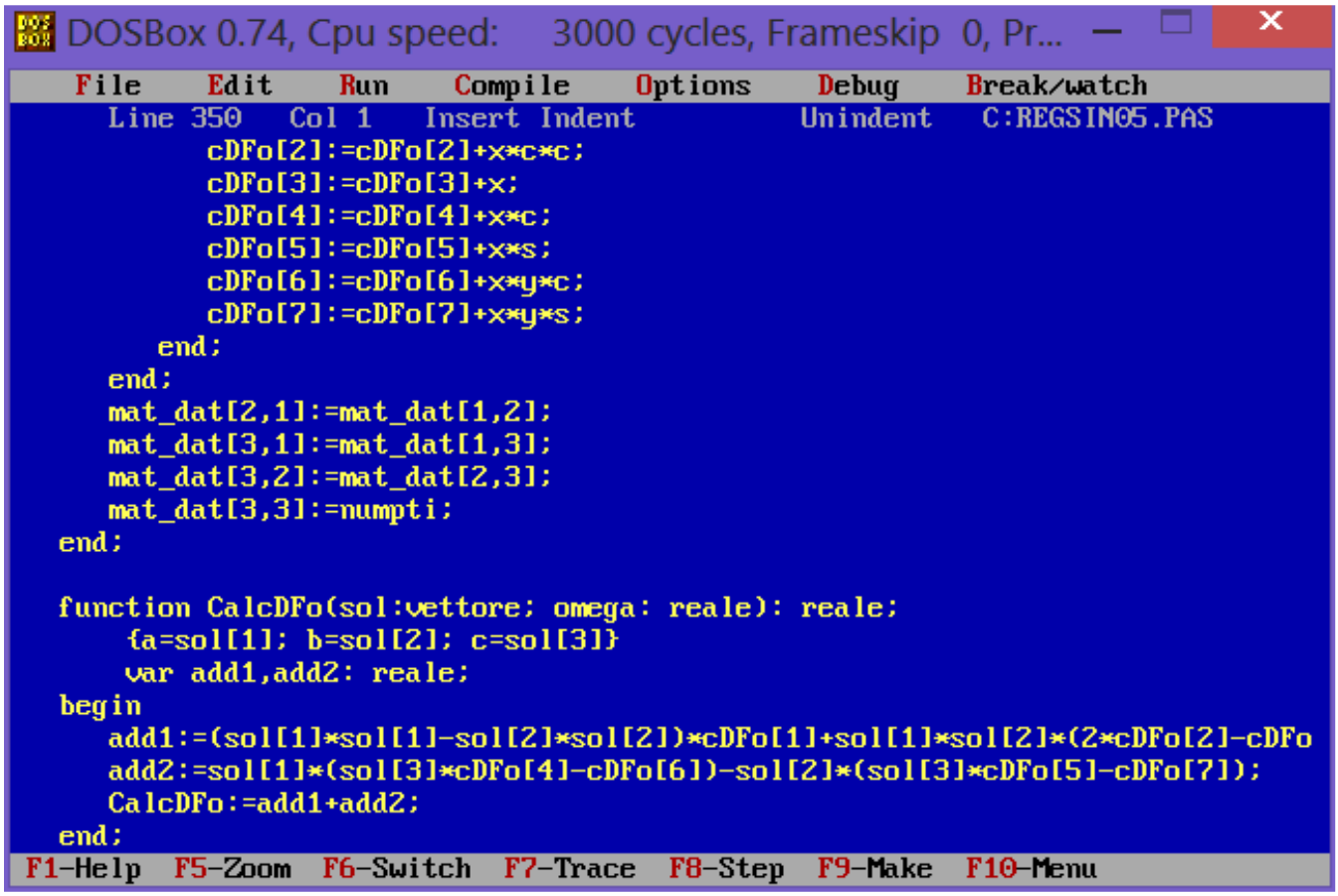
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pr...

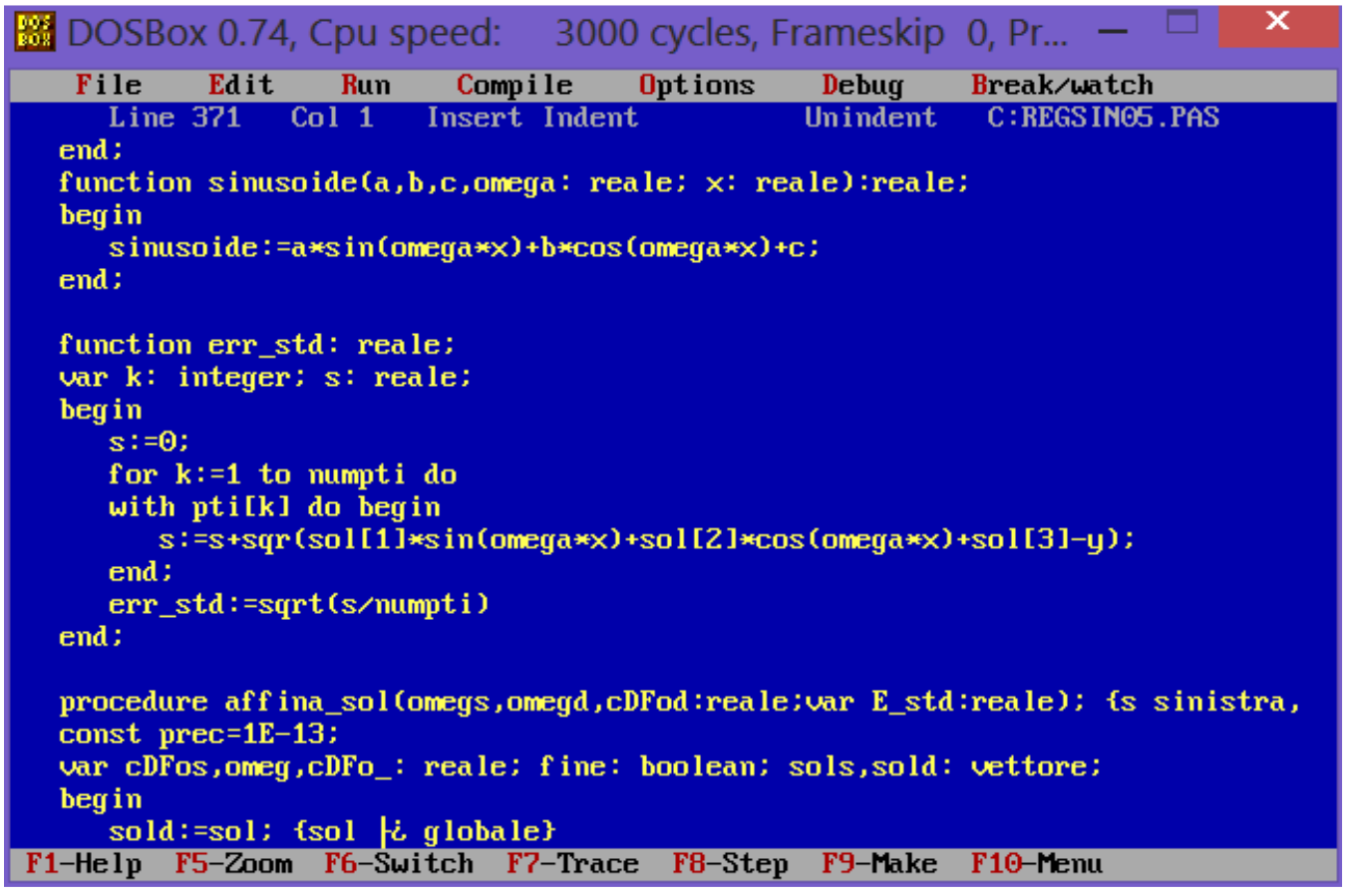
File Edit Run Compile Options Debug Break/watch

Line 329 Col 1 Insert Indent Unindent C:REGSIN05.PAS

```
procedure calc_mat(omega: reale);
var i,j,k: integer;
    s,c: reale;
begin
   for i:=1 to 3 do for j:=1 to 3 do mat_dat[i,j]:=0;
   for k:=1 to 3 do tn[k]:=0;
   for i:=1 to 7 do cDFo[i]:=0;
   for k:= 1 to numpti do begin
      with pti[k] do begin
         s:=sin(omega*x);
         c:=cos(omega*x);
         mat_dat[1,3]:=mat_dat[1,3]+s;
         mat_dat[2,3]:=mat_dat[2,3]+c;
         mat_dat[1,2]:=mat_dat[1,2]+s*c;
         mat_dat[1,1]:=mat_dat[1,1]+s*s;
         mat_dat[2,2]:=mat_dat[2,2]+c*c;
         tn[1]:=tn[1]+y*s;
         tn[2]:=tn[2]+y*c;
         tn[3]:=tn[3]+y;
         cDFo[1]:=cDFo[1]+x*s*c; {coefficienti della derivata di F rispetto
         cDFo[2]:=cDFo[2]+x*c*c;
```

F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F10-Menu


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pr...
File Edit Run Compile Options Debug Break/watch
Line 350 Col 1 Insert Indent Unindent C:REGSIN05.PAS
cDFo[2]:=cDFo[2]+x*c*c;
cDFo[3]:=cDFo[3]+x;
cDFo[4]:=cDFo[4]+x*c;
cDFo[5]:=cDFo[5]+x*s;
cDFo[6]:=cDFo[6]+x*y*c;
cDFo[7]:=cDFo[7]+x*y*s;
end;
end;
mat_dat[2,1]:=mat_dat[1,2];
mat_dat[3,1]:=mat_dat[1,3];
mat_dat[3,2]:=mat_dat[2,3];
mat_dat[3,3]:=numpti;
end;
function CalcDFo(sol:vettore; omega: reale): reale;
{a=sol[1]; b=sol[2]; c=sol[3]}
var add1,add2: reale;
begin
add1:=(sol[1]*sol[1]-sol[2]*sol[2])*cDFo[1]+sol[1]*sol[2]*(2*cDFo[2]-cDFo
add2:=sol[1]*(sol[3]*cDFo[4]-cDFo[6])-sol[2]*(sol[3]*cDFo[5]-cDFo[7]);
CalcDFo:=add1+add2;
end;
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F10-Menu


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pr...
File Edit Run Compile Options Debug Break/watch
Line 371 Col 1 Insert Indent Unindent C:REGSIN05.PAS
end;
function sinusoide(a,b,c,omega: reale; x: reale):reale;
begin
sinusoide:=a*sin(omega*x)+b*cos(omega*x)+c;
end;
function err_std: reale;
var k: integer; s: reale;
begin
s:=0;
for k:=1 to numpti do
with pti[k] do begin
s:=s+sqr(sol[1]*sin(omega*x)+sol[2]*cos(omega*x)+sol[3]-y);
end;
err_std:=sqrt(s/numpti)
end;
procedure affina_sol(omegs,omegd,cDFod:reale;var E_std:reale); {s sinistra,
const prec=1E-13;
var cDFos,omeg,cDFo_: reale; fine: boolean; sols,sold: vettore;
begin
sold:=sol; {sol ¿ globale}
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F10-Menu

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pr...

File Edit Run Compile Options Debug Break/watch

Line 392 Col 1 Insert Indent Unindent C:REGSIN05.PAS

```
sold:=sol; {sol è globale}
calc_mat(omegs);
Gauss_Jordan(mat_dat, dim, err, false);
Prodotto_mat_vett(mat_dat,tn,sol,dim); {mat_dat,tn,sol,dim tutte variabil
sols:=sol;
cDFos:=CalcDFo(sols,omegs);
fine := false;
if (cDfos*cDFod < 0) then begin
   repeat
      omeg:=0.5*(omegs+omegd);
      calc_mat(omeg);
      Gauss_Jordan(mat_dat, dim, err, false);
      Prodotto_mat_vett(mat_dat,tn,sol,dim);
      cDFo_:=CalcDFo(sol,omeg);
      {writeln(' omega: ',omegs:15:12,omeg:15:12,omegd:15:12,' cDFo_ ',cD
      if keypressed then begin write(' Stop ? '); risp:=readkey; end;
      if risp in [' ','S','s'] then begin
         readln(risp);
         if risp in ['S','s'] then exit;
      end;
      if (abs(cDFo_)<=prec) then begin
         fine:=true;
```

F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F10-Menu

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pr...

File Edit Run Compile Options Debug Break/watch

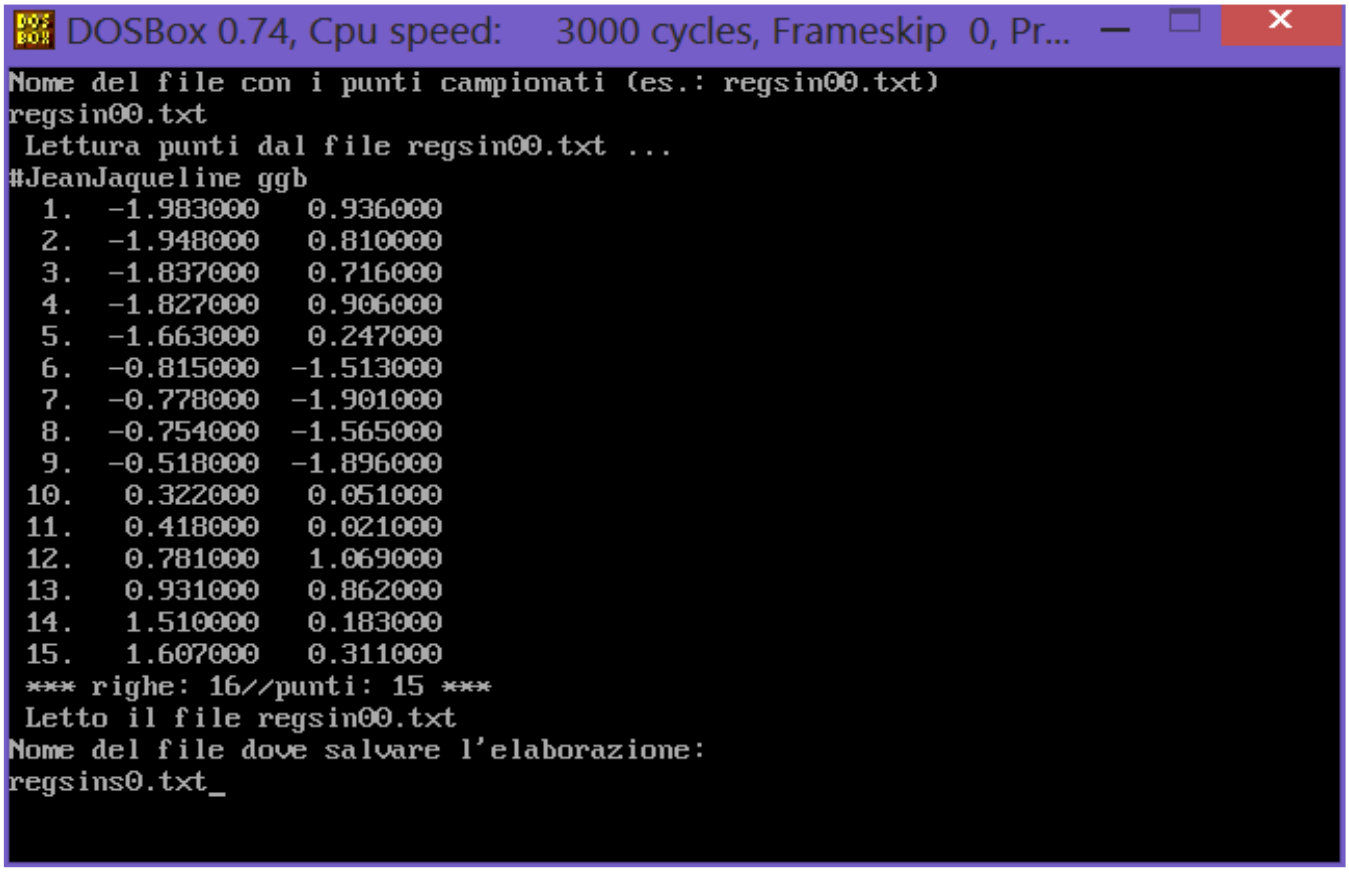
Line 413 Col 1 Insert Indent Unindent C:REGSIN05.PAS

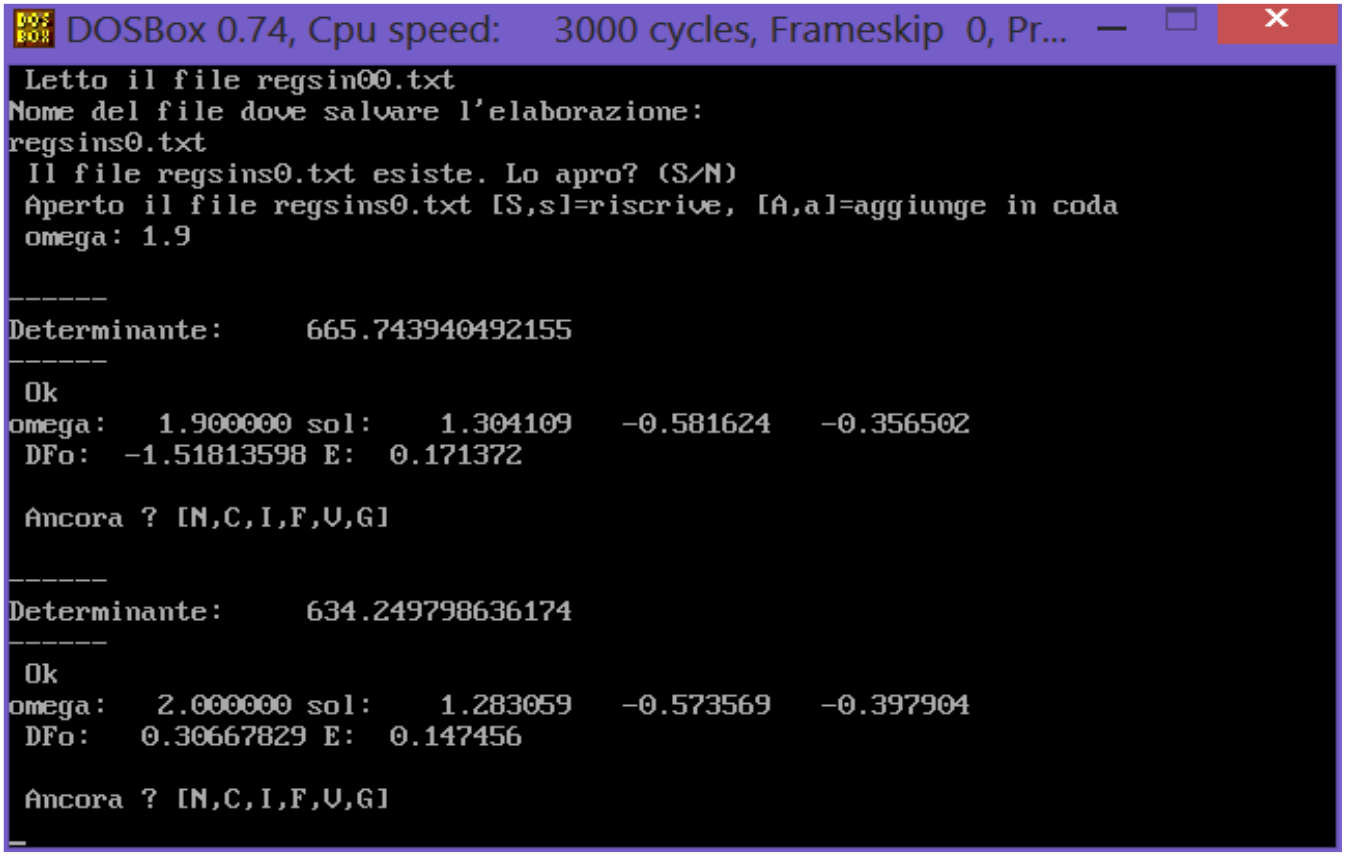
```
         fine:=true;
         omega:=omeg; {aggiornamento del valore globale di omega con il v
         E_std:=err_std;
         writeln(' omega: ',omeg:16:12,' ',sol[1]:campo:decimali,' ',sol[
         sol[3]:campo:decimali);
         writeln(' cDFo: ',cdFo_:18:15,' E: ',E_std:7:4);
      end
      else begin
         if (CdFo_*cdFos >0) then omegs:=omeg else omegd:=omeg
      end;
    until fine;
  end
  else writeln(' Errore: funzione concorde agli estremi ', cDFos:10,' ',cDf
end;

function f1(x:reale):reale;
begin {f1}
  f1:=sinusoide(sol[1],sol[2],sol[3],omega,x);
end;

procedure grafico2;
const s1:str80='y=a*sen(omega*x)+b*cos(omega*x)+c';
```

F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F10-Menu

Sempre per ragioni di spazio si omette il codice per l'esecuzione del grafico e si passa alle diapositive con gli esempi elaborati.

Esempio n. 1.


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pr...
Nome del file con i punti campionati (es.: regsin00.txt)
regsin00.txt
Lettura punti dal file regsin00.txt ...
#JeanJaqueline ggb
1. -1.983000 0.936000
2. -1.948000 0.810000
3. -1.837000 0.716000
4. -1.827000 0.906000
5. -1.663000 0.247000
6. -0.815000 -1.513000
7. -0.778000 -1.901000
8. -0.754000 -1.565000
9. -0.518000 -1.896000
10. 0.322000 0.051000
11. 0.418000 0.021000
12. 0.781000 1.069000
13. 0.931000 0.862000
14. 1.510000 0.183000
15. 1.607000 0.311000
*** righe: 16//punti: 15 ***
Letto il file regsin00.txt
Nome del file dove salvare l'elaborazione:
regsins0.txt_


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pr...
Letto il file regsin00.txt
Nome del file dove salvare l'elaborazione:
regsins0.txt
Il file regsins0.txt esiste. Lo apro? (S/N)
Aperto il file regsins0.txt [S,s]=riscrive, [A,a]=aggiunge in coda
omega: 1.9

Determinante: 665.743940492155

Ok
omega: 1.900000 sol: 1.304109 -0.581624 -0.356502
DFo: -1.51813598 E: 0.171372
Ancora ? [N,C,I,F,V,G]

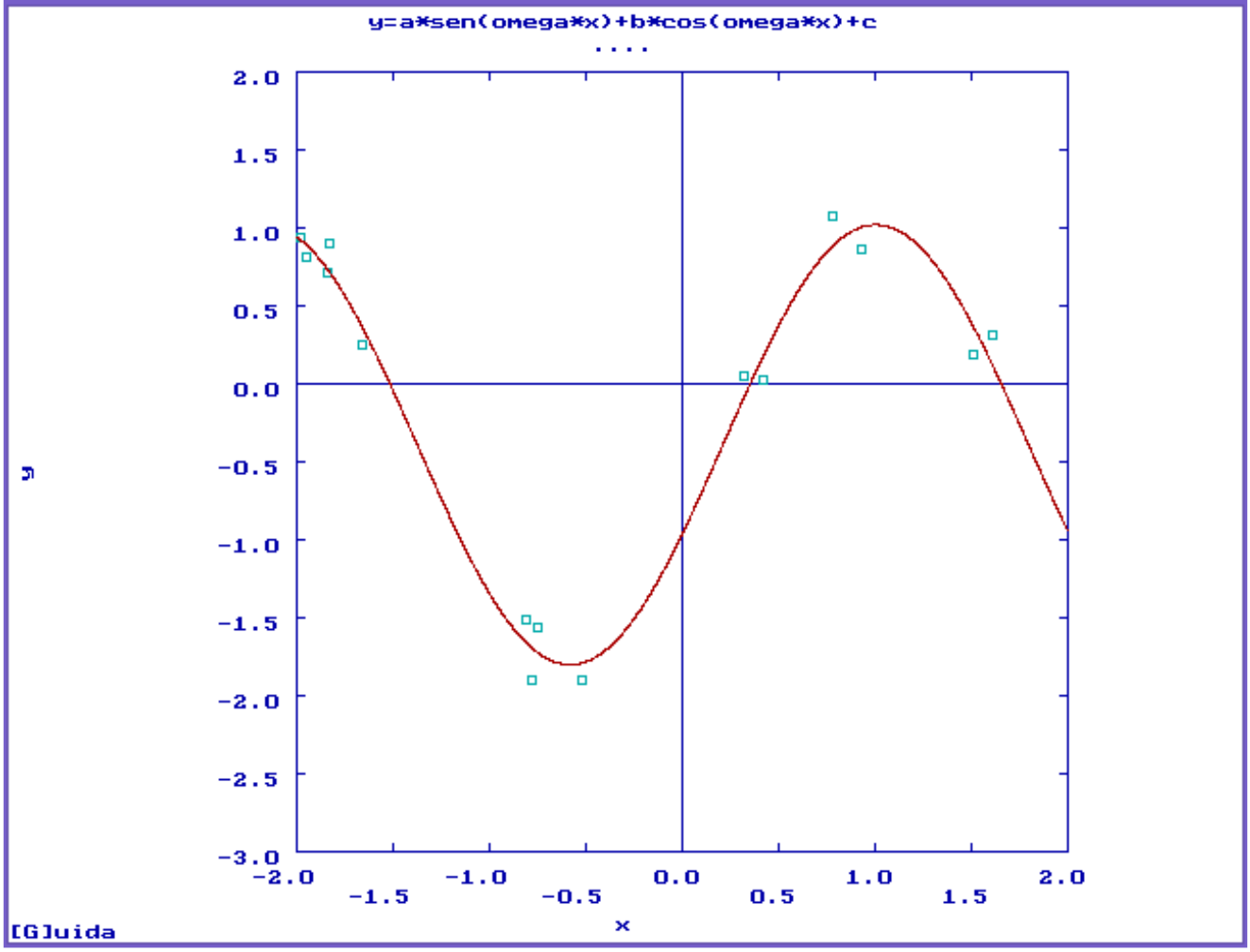
Determinante: 634.249798636174

Ok
omega: 2.000000 sol: 1.283059 -0.573569 -0.397904
DFo: 0.30667829 E: 0.147456
Ancora ? [N,C,I,F,V,G]
_

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pr...

```
omega:   1.900000 sol:    1.304109   -0.581624   -0.356502
 DFo:  -1.51813598 E:  0.171372

 Ancora ? [N,C,I,F,V,G]

------
Determinante:     634.249798636174
------
 Ok
omega:   2.000000 sol:    1.283059   -0.573569   -0.397904
 DFo:   0.30667829 E:  0.147456

 Ancora ? [N,C,I,F,V,G]
 omega:   1.981305611028   1.289338  -0.571687  -0.390698
 cDFo:  0.000000000000070 E:  0.1461

------
Determinante:     640.065685182867
------
 Ok
omega:   1.981306 sol:    1.289338   -0.571687   -0.390698
 DFo:   0.00000000 E:  0.146140

 Ancora ? [N,C,I,F,V,G]
_
```

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pr...

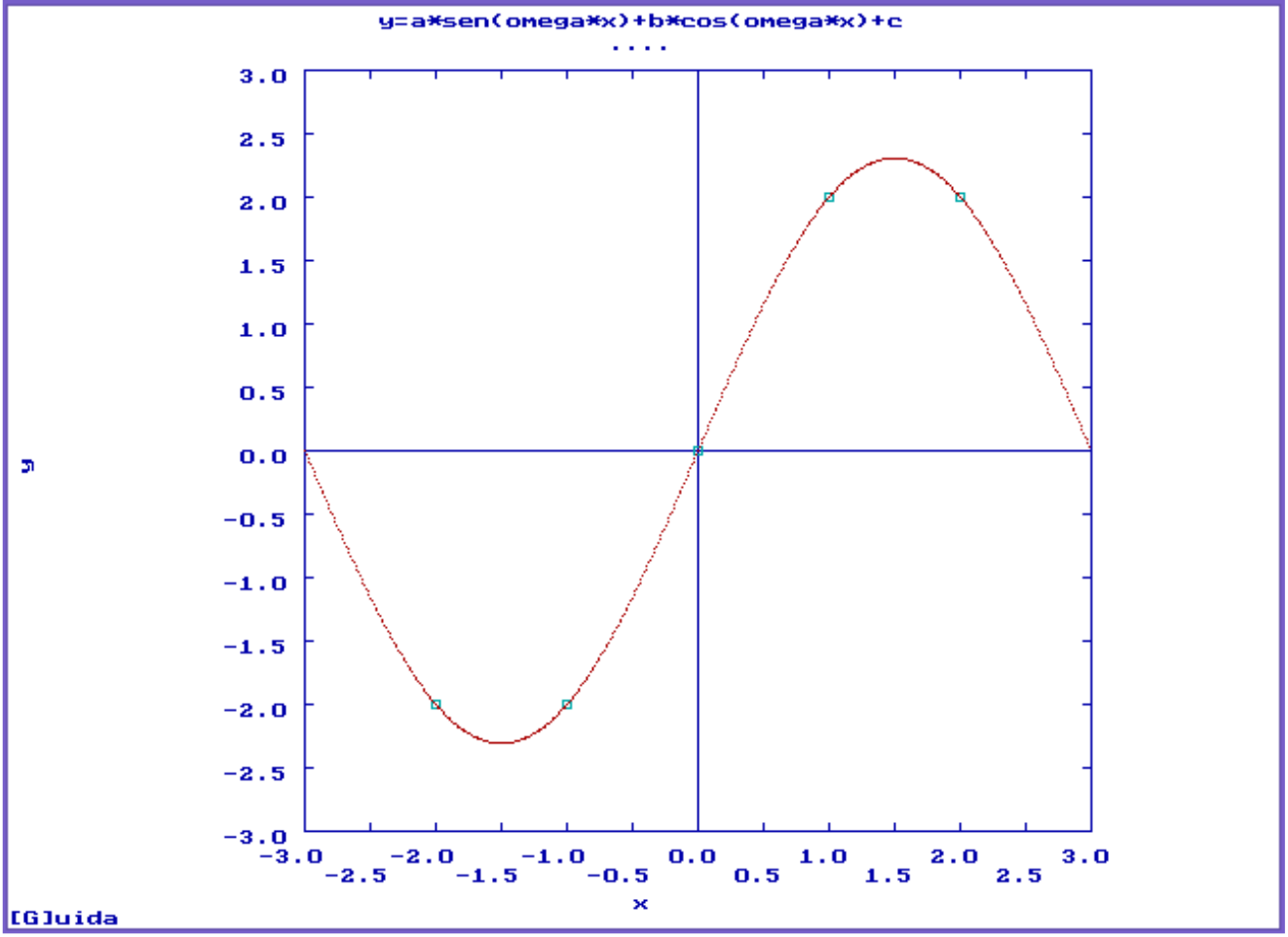


Esempio n. 2 (dati antisimmetrici)

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pr...

```
omega:   1.000000 sol:    2.281287   -0.000000    0.000000
 DFo:  -0.48051581 E:  0.069250

 Ancora ? [N,C,I,F,V,G]

------
Determinante:      28.922583896797
------
 Ok
omega:   1.100000 sol:    2.347792    0.000000   -0.000000
 DFo:   0.75946056 E:  0.086946

 Ancora ? [N,C,I,F,V,G]
 omega:   1.047197551197   2.309401  -0.000000   0.000000
 cDFo: -0.000000000000006 E:  0.0000

------
Determinante:      27.000000000000
------
 Ok
omega:   1.047198 sol:    2.309401   -0.000000    0.000000
 DFo:  -0.00000000 E:  0.000000

 Ancora ? [N,C,I,F,V,G]
_
```

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pr...

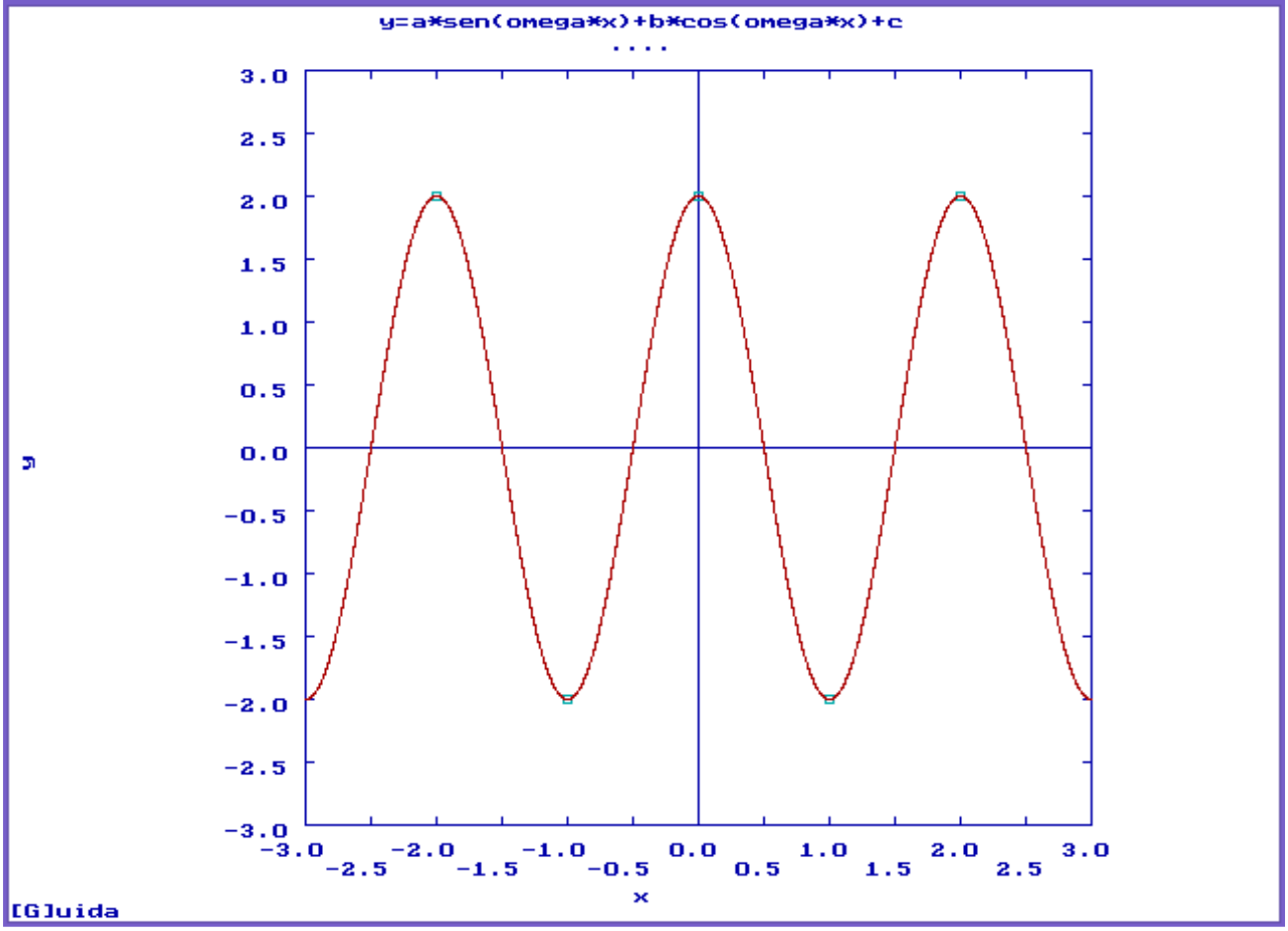


Esempio n. 3 (dati simmetrici)

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pr...

```
omega:   3.100000 sol:    0.000000    2.003172    0.001443
 DFo:  -0.00153965 E:  0.002529

 Ancora ? [N,C,I,F,V,G]

------
Determinante:       0.810497350216
------
 Ok
omega:   3.200000 sol:    0.000000    2.006255    0.002850
 DFo:   0.00427613 E:  0.004993

 Ancora ? [N,C,I,F,V,G]
 omega:   3.141601562500   0.000000   2.000000   0.000000
 cDFo:  0.000000000000015 E:  0.0000

------
Determinante:       0.000000019048
------
 Ok
omega:   3.141602 sol:    0.000000    2.000000    0.000000
 DFo:   0.00000000 E:  0.000000

 Ancora ? [N,C,I,F,V,G]
_
```

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pr...



Esempio n. 4 (dati simmetrici).

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pr...

```
omega:   1.500000 sol:    0.000000   -0.993895    0.033322
 DFo:  -0.06364233 E:  0.031262

 Ancora ? [N,C,I,F,V,G]

------
Determinante:      27.793729580122
------
 Ok
omega:   1.600000 sol:    0.000000   -1.004798   -0.012010
 DFo:   0.03690451 E:  0.014439

 Ancora ? [N,C,I,F,V,G]
 omega:   1.570796326795   0.000000  -1.000000   0.000000
 cDFo: -0.000000000000001 E:  0.0000

------
Determinante:      28.000000000000
------
 Ok
omega:   1.570796 sol:    0.000000   -1.000000    0.000000
 DFo:  -0.00000000 E:  0.000000

 Ancora ? [N,C,I,F,V,G]
_
```

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pr...