SIS Scuola Interdipartimentale delle **Scienze**, dell'**Ingegneria** e della **Salute**

Laurea Magistrale in STN

Applicazioni di Calcolo Scientifico e Laboratorio di ACS

(12 cfu)

prof. Mariarosaria Rizzardi

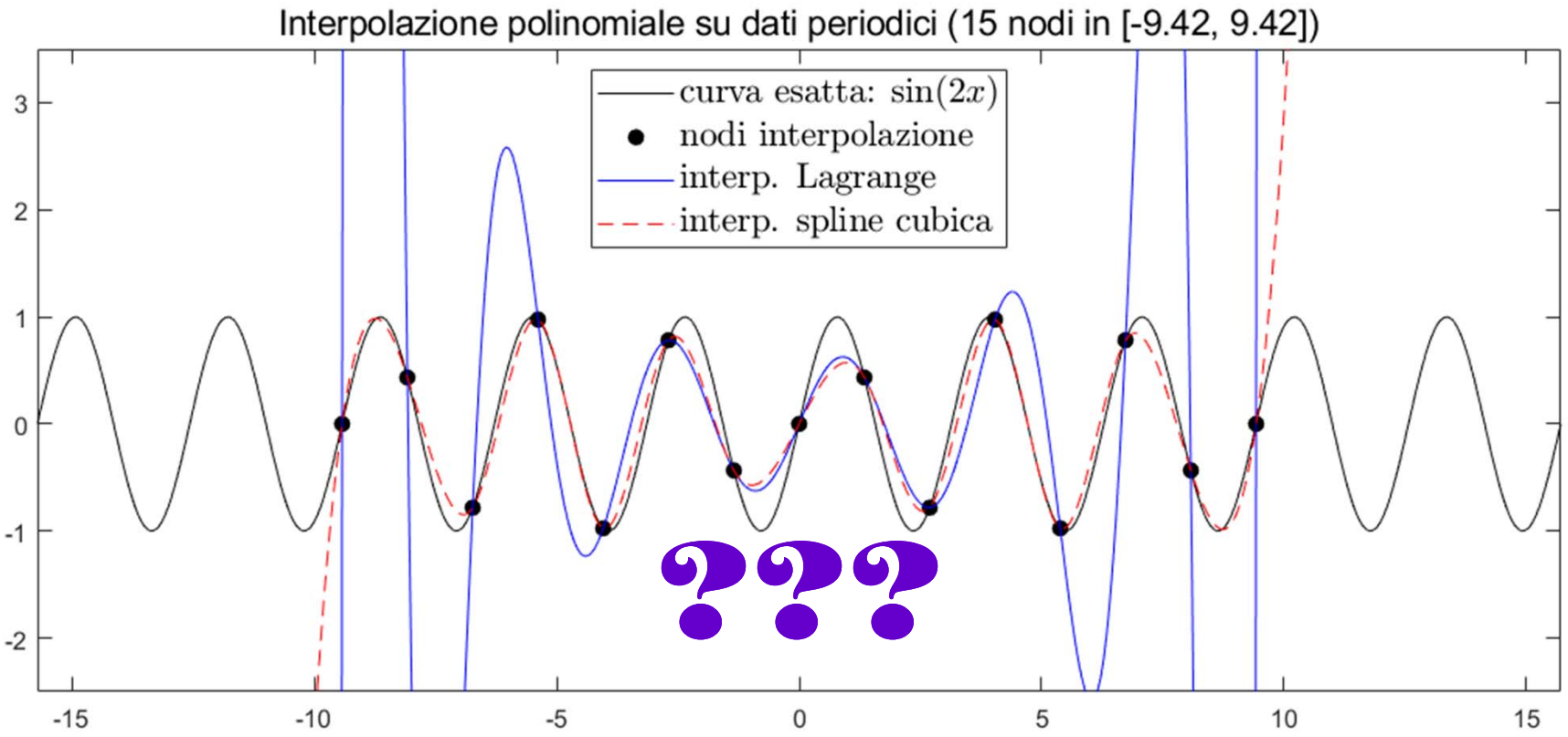
Centro Direzionale di Napoli – Isola C4
stanza: n. 423 – Lato Nord, 4° piano
tel.: 081 547 6545
email: mariarosaria.rizzardi@uniparthenope.it

Argomenti trattati

- Calcolo Numerico in MATLAB:
  - polinomi trigonometrici;
  - interpolazione trigonometrica.

Perché ... Interpolazione Trigonometrica ?

Ricorrendo all'interpolazione polinomiale a partire da **dati periodici**, si ottiene ...



I dati provengono da una funzione periodica ma le funzioni interpolanti non lo sono!

Polinomi Trigonometrici di periodo 2π

Polinomio algebrico di grado N (N+1 coefficienti)

$$P_N(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_N x^N$$

Polinomio trigonometrico di grado N (N+1 coefficienti)

$$S_N(x) = \alpha_0 + \alpha_1 \cos x + \beta_1 \sin x + \alpha_2 \cos 2x + \beta_2 \sin 2x + \cdots$$
$$\cdots + \alpha_{\frac{N}{2}} \cos \tfrac{N}{2} x + \beta_{\frac{N}{2}} \sin \tfrac{N}{2} x$$

$S_N(x) \in \mathbb{R}$, coefficienti reali

oppure

$$T_N(x) = \gamma_{-\frac{N}{2}} e^{-i\frac{N}{2}x} + \cdots + \gamma_{-2} e^{-i2x} + \gamma_{-1} e^{-ix} + \gamma_0 +$$
$$+ \gamma_1 e^{ix} + \gamma_2 e^{i2x} + \cdots + \gamma_{+\frac{N}{2}} e^{+i\frac{N}{2}x}$$

$T_N(x) \in \mathbb{C}$, coefficienti complessi

N pari

forma reale

forma complessa

Dalla forma reale del polinomio trigonometrico:

$$S_N(x) = \alpha_0 + \alpha_1 \cos x + \beta_1 \sin x + \alpha_2 \cos 2x + \beta_2 \sin 2x + \cdots + \alpha_{\frac{N}{2}} \cos \frac{N}{2} x + \beta_{\frac{N}{2}} \sin \frac{N}{2} x$$

$S_N(x) \in \mathbb{R}$, coefficienti reali

Formula di Eulero: $e^{i\theta} = \cos(\theta) + i\sin(\theta)$ ⇒

con le sostituzioni $\cos kx = \frac{e^{ikx} + e^{-ikx}}{2}$; $\sin kx = \frac{e^{ikx} - e^{-ikx}}{2i}$ si ottiene

$$S_N(x) = 2\alpha_0 + \alpha_1 (e^{ix} + e^{-ix}) - i\beta_1 (e^{ix} - e^{-ix}) + \alpha_2 (e^{i2x} + e^{-i2x}) - i\beta_2 (e^{i2x} - e^{-i2x}) + \cdots + \alpha_{\frac{N}{2}} (e^{i\frac{N}{2}x} + e^{-i\frac{N}{2}x}) - i\beta_{\frac{N}{2}} (e^{i\frac{N}{2}x} - e^{-i\frac{N}{2}x})$$

e si trova la forma complessa

forma complessa

$$S_N(x) = e^{-i\frac{N}{2}x} \underbrace{(\alpha_{\frac{N}{2}} + i\beta_{\frac{N}{2}})}_{\gamma_{-\frac{N}{2}}} + \cdots + e^{-i2x} \underbrace{(\alpha_2 + i\beta_2)}_{\gamma_{-2}} + e^{-ix} \underbrace{(\alpha_1 + i\beta_1)}_{\gamma_{-1}} + \underbrace{2\alpha_0}_{\gamma_0} +$$

$$T_N(x) = \cdots$$

$$+ e^{+ix} \underbrace{(\alpha_1 - i\beta_1)}_{\gamma_{+1}} + e^{+i2x} \underbrace{(\alpha_2 - i\beta_2)}_{\gamma_{+2}} + \cdots + \cdots + e^{+i\frac{N}{2}x} \underbrace{(\alpha_{\frac{N}{2}} - i\beta_{\frac{N}{2}})}_{\gamma_{+\frac{N}{2}}}$$

Viceversa, affinché la forma complessa del polinomio trigonometrico $T_N(x)$ restituisca valori reali, $\forall k$ i coefficienti di e^{-ikx} e di e^{+ikx} devono essere complessi e coniugati:

$$\gamma_{-k} = \overline{\gamma_{+k}}$$

Esempio: polinomi di grado 2

Pol. algebrico di grado 2 (3 coefficienti)

$$P_2(x) = c_0 + c_1 x + c_2 x^2 \qquad P_2(x) = 1 + 2x - x^2$$

Pol. trigonometrico di grado 2 (3 coefficienti)

$$S_2(x) = \alpha_0 + \alpha_1 \cos x + \beta_1 \sin x \qquad S_2(x) = 1 + 2\cos x - \sin x$$

forma reale

$$\cos kx = \frac{e^{ikx} + e^{-ikx}}{2}; \quad \sin kx = \frac{e^{ikx} - e^{-ikx}}{2i}$$

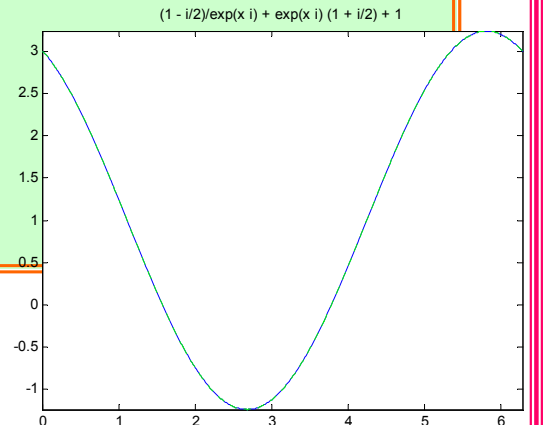
stesso polinomio!

forma complessa

$$T_2(x) = \gamma_{-1} e^{-ix} + \gamma_0 + \gamma_{+1} e^{ix} \qquad T_2(x) = \left(1 - \frac{i}{2}\right) e^{-ix} + 1 + \left(1 + \frac{i}{2}\right) e^{+ix}$$

```
syms x real;  S2=1+2*cos(x)-sin(x);
T2=subs(S2, {cos(x),sin(x)}, {(exp(i*x)+exp(-i*x))/2, (exp(i*x)-exp(-i*x))/(2*i)})
T2 =
exp(-x*1i)*(1 - 1i/2) + exp(x*1i)*(1 + 1i/2) + 1
ezplot(S2,[0 2*pi]); hold on
h=ezplot(T2,[0 2*pi]); set(h,'Color','g','LineStyle',':')
[simplify(real(T2))       simplify(imag(T2))]
ans =
[ 2*cos(x) - sin(x) + 1, 0]
```

$S_2(x) = T_2(x)$



anche in forma complessa il polinomio è a valori reali perché i coefficienti di $e^{\pm ix}$ sono complessi coniugati!

ancora... **Esempio:** polinomi di grado 2

Pol. trigonometrico di grado 2 (3 coefficienti)

$$T_2(x) = \gamma_{-1} e^{-ix} + \gamma_0 + \gamma_{+1} e^{ix}$$

mettendo in evidenza e^{-ix}

$$T_2(x) = e^{-ix}\left(\gamma_{-1} + \gamma_0 e^{ix} + \gamma_{+1} e^{i2x}\right)$$

$$Q(x) = \gamma_{-1} + \gamma_0 e^{ix} + \gamma_{+1} e^{i2x} = c_0 + c_1 e^{ix} + c_2 e^{i2x}$$

3 coefficienti!

grado di $Q(x)$?

forma complessa

$$T_2(x) = \left(1 - \frac{i}{2}\right) e^{-ix} + 1 + \left(1 + \frac{i}{2}\right) e^{+ix}$$

$Q(x)$ si può interpretare come:

$$T_4(x) = 0e^{-i2x} + 0e^{-ix} + c_0 + c_1 e^{ix} + c_2 e^{i2x}$$

$Q(x)$ è un particolare polinomio trigonometrico di grado 4

Un polinomio trigonometrico di grado N (N pari)

(N+1 coefficienti)

in forma reale $(S_N(x) \in \mathbb{R})$

$$S_N(x) = \alpha_0 + \alpha_1 \cos x + \beta_1 \sin x + \alpha_2 \cos 2x + \beta_2 \sin 2x + \cdots$$
$$\cdots + \alpha_{\frac{N}{2}} \cos \tfrac{N}{2} x + \beta_{\frac{N}{2}} \sin \tfrac{N}{2} x$$

oppure in forma complessa $(T_N(x) \in \mathbb{C})$

$$T_N(x) = \gamma_{-\frac{N}{2}} e^{-i\frac{N}{2}x} + \cdots + \gamma_{-2} e^{-i2x} + \gamma_{-1} e^{-ix} + \gamma_0 +$$
$$+ \gamma_1 e^{ix} + \gamma_2 e^{i2x} + \cdots + \gamma_{+\frac{N}{2}} e^{+i\frac{N}{2}x}$$

è una funzione periodica di periodo 2π

Il particolare **polinomio trigonometrico** $Q(x)$ con N+1 coefficienti

in forma complessa $(Q(x) \in \mathbb{C})$

$$Q(x) = c_0 + c_1 e^{ix} + c_2 e^{i2x} + c_3 e^{i3x} + \cdots + c_N e^{iNx}$$

(Q di grado $2N$ con N coefficienti nulli)

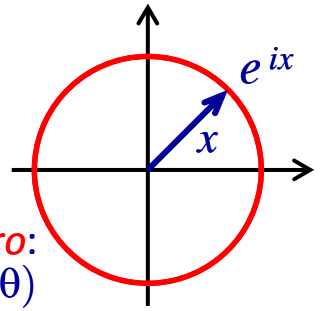
è una funzione periodica di periodo 2π

Relazione con i polinomi algebrici

$$Q(x) = c_0 + c_1 e^{ix} + c_2 e^{i2x} + c_3 e^{i3x} + \cdots + c_N e^{iNx}$$

posto $z = e^{ix}$, per $x \in \mathbb{R}$, si ottiene

$$Q(z) = c_0 + c_1 z + c_2 z^2 + c_3 z^3 + \cdots + c_N z^N$$



per la *Formula di Eulero*:
$e^{i\theta} = \cos(\theta) + i\sin(\theta)$

è un polinomio algebrico di grado N calcolato per $z = e^{ix} \in \Gamma(0,1)$
(cioè sulla circonferenza unitaria di centro O)

per $Q(z)$ si può sfruttare il Teor. di esistenza ed unicità
del polinomio interpolante di Lagrange

Relazione tra polinomi trigonometrici ed algebrici

N pari

In generale, se si scrive

$$T_N(x) = \gamma_{-\frac{N}{2}} e^{-i\frac{N}{2}x} + \cdots + \gamma_{-2} e^{-i2x} + \gamma_{-1} e^{-ix} + \gamma_0 + \gamma_1 e^{ix} + \gamma_2 e^{i2x} + \cdots + \gamma_{+\frac{N}{2}} e^{+i\frac{N}{2}x}$$

come

$$T_N(x) = e^{-i\frac{N}{2}x}\left(\gamma_{-\frac{N}{2}} + \gamma_{-\frac{N}{2}+1} e^{ix} + \gamma_{-\frac{N}{2}+2} e^{i2x} + \cdots + \gamma_0 e^{i\frac{N}{2}x} + \right.$$
$$\left. + \gamma_1 e^{i(\frac{N}{2}+1)x} + \gamma_2 e^{i(\frac{N}{2}+2)x} + \cdots + \gamma_{+\frac{N}{2}} e^{iNx}\right)$$

posto $z = e^{ix}$, per $x \in \mathbb{R}$, si ottiene

$$T_N(z) = z^{-\frac{N}{2}}\left(\gamma_{-\frac{N}{2}} + \gamma_{-\frac{N}{2}+1} z + \gamma_{-\frac{N}{2}+2} z^2 + \cdots + \gamma_0 z^{\frac{N}{2}} + \right.$$
$$\left. + \gamma_1 z^{\frac{N}{2}+1} + \gamma_2 z^{\frac{N}{2}+2} + \cdots + \gamma_{+\frac{N}{2}} z^N\right)$$

$e^{-i\frac{N}{2}x}$

$$Q(z) = c_0 + c_1 z + c_2 z^2 + c_3 z^3 + \cdots + c_N z^N$$

Come valutare un polinomio trigonometrico in MATLAB

N qualsiasi

$$Q(x) = c_0 + c_1 e^{ix} + c_2 e^{i2x} + \cdots + c_N e^{iNx}$$

$$Q(z) = c_0 + c_1 z + c_2 z^2 + \cdots + c_N z^N \quad \Leftarrow z = e^{ix}$$

```
z=exp(i*x);  Q=polyval(fliplr(c), z);
```

`fliplr()`: perché c=[c_0, c_1, c_2,..., c_N] è un vettore riga,
`flipud()`: se c è un vettore colonna,
`flip(A,dim)`: funziona come `flipud()` se `dim=1`; come `fliplr()` se `dim=2`.

N pari

$$T_N(z) = z^{-\frac{N}{2}} \left(\gamma_{-\frac{N}{2}} + \gamma_{-\frac{N}{2}+1} z + \gamma_{-\frac{N}{2}+2} z^2 + \cdots + \gamma_0 z^{\frac{N}{2}} + \right.$$
$$\left. + \gamma_1 z^{\frac{N}{2}+1} + \gamma_2 z^{\frac{N}{2}+2} + \cdots + \gamma_{+\frac{N}{2}} z^N \right)$$

$$T_N(z) = z^{-\frac{N}{2}} \left(c_0 + c_1 z + c_2 z^2 + \cdots + \right.$$
$$\left. + \cdots + c_N z^N \right) \quad \Leftarrow z = e^{ix}$$

$Q(z)$

```
z=exp(i*x);  T_N=z.^(-N/2) .* polyval(fliplr(c), z);
```

Richiami: Interpolazione polinomiale di Lagrange

Teorema: Assegnati $N+1$ punti (x_k, y_k) con $x_j \neq x_k$ per $j \neq k$, esiste un unico polinomio (algebrico) $P_N(x) \in \Pi_N$, cioè di grado al più N, che sia interpolante sui nodi assegnati:

$$P_N(x_k) = y_k \quad \forall\, k = 0, 1, 2, \ldots, N$$

dove $P_N(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_N x^N$

$\{1, x, x^2, x^3, \ldots, x^N\}$ base di Π_N

Metodo dei coefficienti indeterminati

La dimostrazione (costruttiva) del teorema si basa sul determinare i coefficienti $\{c_h\}$ del polinomio come soluzione del sistema lineare determinato che si ottiene dalle condizioni di interpolazione:

$$P_N(x_k) = y_k, \quad k = 0, 1, \ldots, N$$

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^N \\ 1 & x_1 & x_1^2 & \cdots & x_1^N \\ 1 & x_2 & x_2^2 & \cdots & x_2^N \\ \vdots & & \vdots & \cdots & \vdots \\ 1 & x_N & x_N^2 & \cdots & x_N^N \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_N \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}$$

matrice di Vandermonde

Complessità di tempo (m. Gauss): $= O\left(\frac{(N+1)^3}{3}\right)$

Problema generale di Interpolazione Trigonometrica
(periodo T=2π)

Assegnati $N+1$ nodi distinti x_k qualsiasi in $[0, 2\pi[$ ed $N+1$ valori $y_k \in \mathbb{C}$, si cerca un polinomio trigonometrico di grado N interpolante $Q(x)$: $Q(x_k)=y_k$, $k=0,1,\ldots,N$, del tipo (cioè con $N+1$ coefficienti):

$$Q(x) = c_0 + c_1 e^{ix} + c_2 e^{i2x} + c_3 e^{i3x} + \cdots + c_N e^{iNx}$$

$$z = e^{ix} \Rightarrow Q(z) = c_0 + c_1 z + c_2 z^2 + c_3 z^3 + \cdots + c_N z^N$$

per il ***Teor. di interpolazione di Lagrange***, imponendo le condizioni di interpolazione si ottiene il sistema lineare nelle incognite c_h

$k=0,1,\ldots,N \quad Q(x_k)=Q(z_k)=y_k$

$$\begin{pmatrix} 1 & e^{ix_0} & e^{i2x_0} & \cdots & e^{iNx_0} \\ 1 & e^{ix_1} & e^{i2x_1} & \cdots & e^{iNx_1} \\ 1 & e^{ix_2} & e^{i2x_2} & \cdots & e^{iNx_2} \\ \vdots & & \vdots & \cdots & \vdots \\ 1 & e^{ix_N} & e^{i2x_N} & \cdots & e^{iNx_N} \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_N \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}$$

1° ALGORITMO

si risolve il sistema lineare nelle incognite c_h, e si trovano i coefficienti

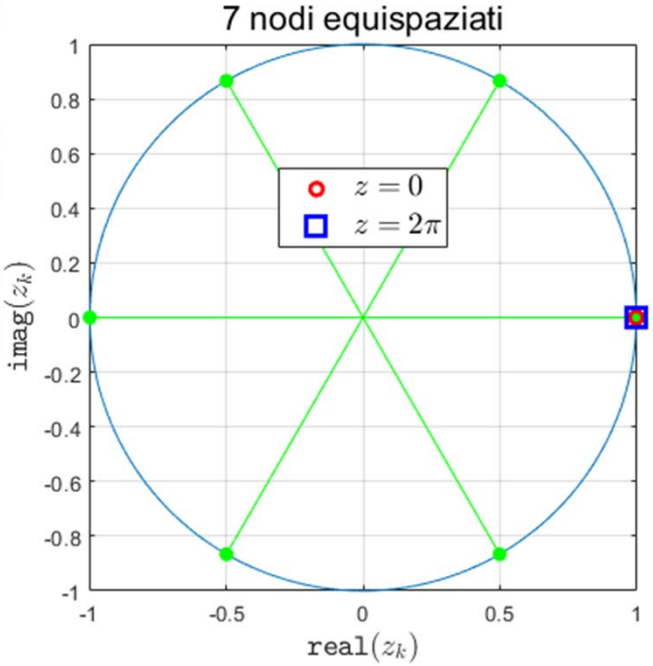
Esempio: perché $[0,2\pi[$ e non $[0,2\pi]$?

semiaperto ↑ chiuso ↑

$$Q(z) = c_0 + c_1 z + c_2 z^2 + c_3 z^3 + \cdots + c_N z^N, \quad z = e^{ix}$$

Per applicare il Teor. di esistenza ed unicità del polinomio interpolante di Lagrange è richiesto che gli $N+1$ nodi $z_k = e^{ix_k}$ con $x_k \in [0, 2\pi]$ (di spaziatura qualsiasi) siano distinti.

```
N=6; k=0:N; % N: grado polinomio
xk=linspace(0,2*pi,N+1)'; % valori equispaziati
zk=exp(i*xk); yk=sin(xk);        compresi 0 e 2π
fplot(@sin,[0 2*pi]); axis equal; hold on
plot(xk,yk,'sb')
% A: matrice dei coefficienti del sistema
A=repmat(zk,1,N+1).^repmat(k,N+1,1);
disp(det(A))
 -7.6738e-28 - 3.1743e-13i                ???
```



Se il determinante di A è nullo,
non esiste un'unica soluzione del sistema!

```
disp([zk(1) zk(end)])
        1   ≈   1 - 2.4493e-16i
```

Ora i nodi sono (zk,yk): le zk devono essere distinte!

Esempio: perché $[0,2\pi[$ e non $[0,2\pi]$?

semiaperto ↑ chiuso ↑

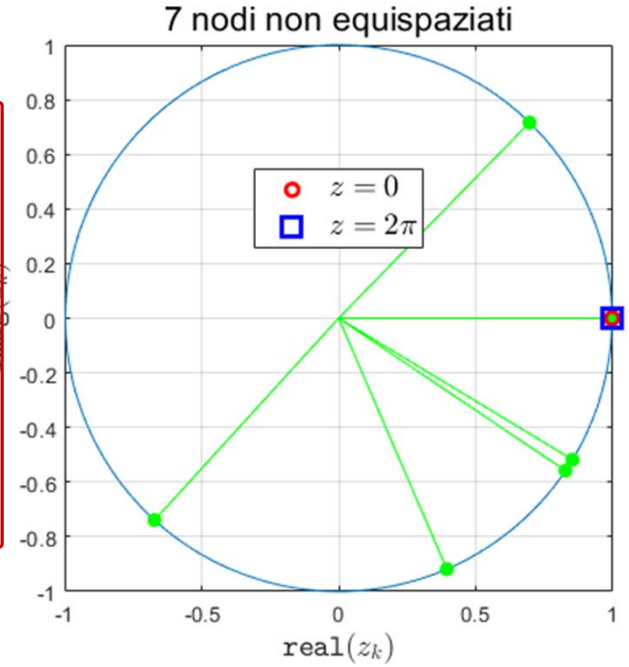
$$Q(z) = c_0 + c_1 z + c_2 z^2 + c_3 z^3 + \cdots + c_N z^N, \quad z = e^{ix}$$

Per applicare il Teor. di esistenza ed unicità del polinomio interpolante di Lagrange è richiesto che gli $N+1$ nodi $z_k = e^{ix_k}$ con $x_k \in [0, 2\pi]$ (di spaziatura qualsiasi) siano distinti.

```
N=6; k=0:N; % N: grado polinomio
xk=[0;2*pi*rand(N-1,1);2*pi]; % valori non equispaziati
zk=exp(i*xk); yk=sin(xk);
fplot(@sin,[0 2*pi]); axis equal; hold on
plot(xk,yk,'sb')
% A: matrice dei coefficienti del sistema
A=repmat(zk, 1,N+1) .^ repmat(k, N+1,1);
disp(det(A))
 -7.6738e-28 - 3.1743e-13i
```

compresi 0 e 2π

???



Se il determinante di A è nullo,
non esiste un'unica soluzione del sistema!

```
disp([zk(1) zk(end)])
        1   ≈   1 - 2.4493e-16i
```

Ora i nodi sono (zk,yk): le zk devono essere distinte!

Esempio: N=5 (6 nodi di interpolazione)

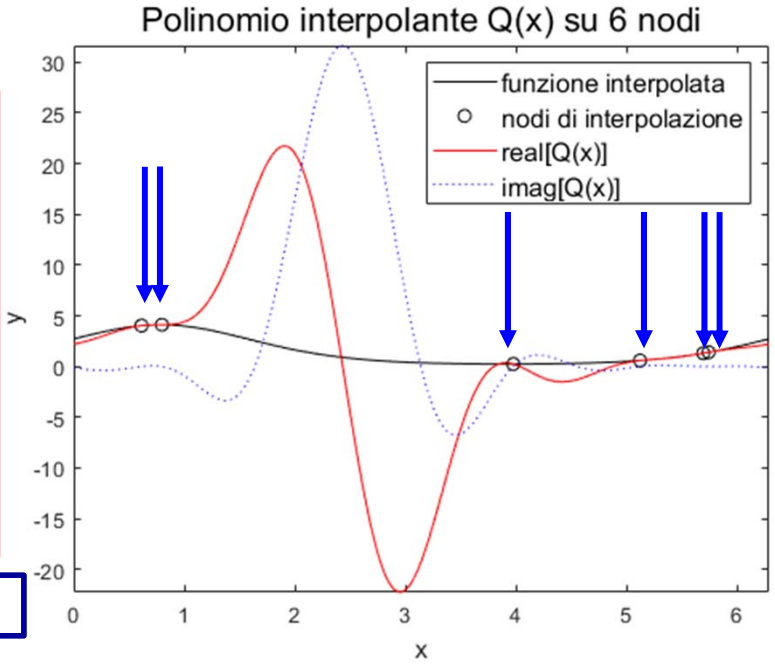
grado dispari

6 nodi qualsiasi (anche non equispaziati)

$$Q_5(x) = c_0 + c_1 e^{ix} + c_2 e^{2ix} + c_3 e^{3ix} + c_4 e^{4ix} + c_5 e^{5ix}$$

```
myfun=@(t)exp(cos(t)+sin(t));
x=linspace(0,2*pi,199); y=myfun(x);
%% (xk,yk) dati di interpolazione di myfun
Npts=6; xk=2*pi*rand(Npts,1); yk=myfun(xk);
N=Npts-1;  k=0:N;  A=exp(i*xk*k);  cQ=A\yk;
Q=polyval(flipud(cQ),exp(i*x));
plot(x,y,'k',xk,yk,'ok'); hold on
plot(x,real(Q),'r',x,imag(Q),'b:')
legend(…)
```

xk*k: prodotto esterno



c'è una parte immaginaria ≠ 0 nel polinomio!

La funzione interpolante è solo **la parte reale** del polinomio

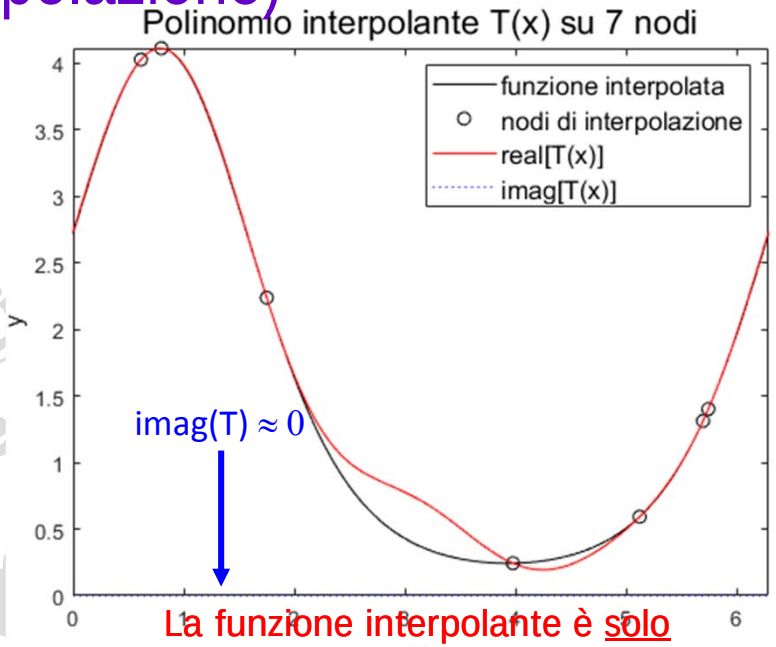
$$\begin{pmatrix} 1 & e^{ix_0} & e^{i2x_0} & \cdots & e^{iNx_0} \\ 1 & e^{ix_1} & e^{i2x_1} & \cdots & e^{iNx_1} \\ 1 & e^{ix_2} & e^{i2x_2} & \cdots & e^{iNx_2} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & e^{ix_N} & e^{i2x_N} & \cdots & e^{iNx_N} \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_N \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} \quad \Leftrightarrow \quad Ac = y$$

Esempio: N=6 (7 nodi di interpolazione)

Grado pari

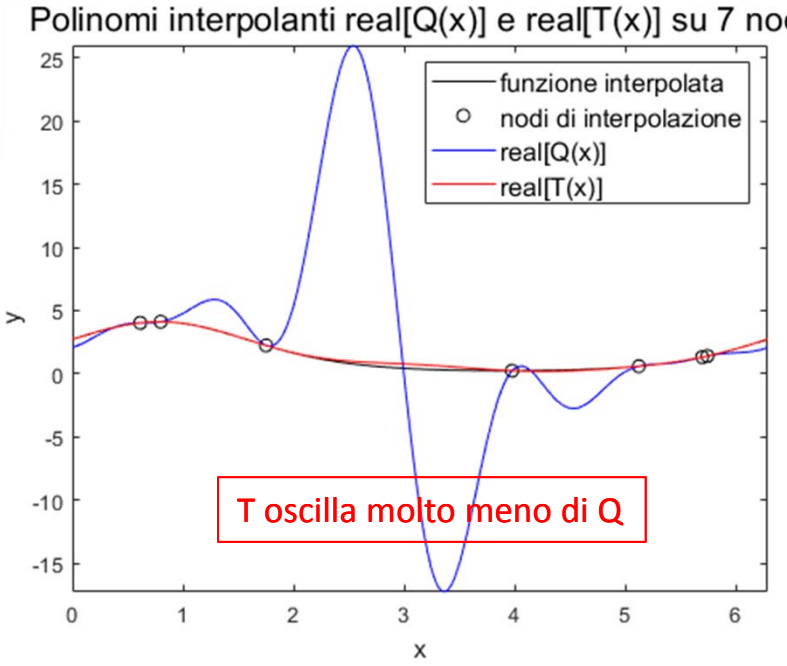
$$T_N(z) = z^{-\frac{N}{2}}\left(\gamma_{-\frac{N}{2}} + \gamma_{-\frac{N}{2}+1}z + \gamma_{-\frac{N}{2}+2}z^2 + \cdots + \gamma_0 z^{\frac{N}{2}} + \right.$$
$$\left. + \gamma_1 z^{\frac{N}{2}+1} + \gamma_2 z^{\frac{N}{2}+2} + \cdots + \gamma_{+\frac{N}{2}} z^N\right)$$

```
myfun=@(t)exp(cos(t)+sin(t));
x=linspace(0,2*pi,199); y=myfun(x);
%% (xk,yk) dati di interpolazione di myfun
Npts=7; xk=2*pi*rand(Npts,1); yk=myfun(xk);

N=Npts-1; k=0:N; A=exp(i*xk*k);
zk=exp(1i*xk); cT=A\(yk.*zk.^(N/2));

z=exp(1i*x); T=z.^(-N/2).*polyval(flipud(cT),z);
plot(x,y,'k',xk,yk,'ok'); hold on
plot(x,real(T),'r',x,imag(T),'b:')
legend(…)
```



La funzione interpolante è solo **la parte reale** del polinomio

```
plot(x,y,'k',xk,yk,'ok'); hold on
plot(x,real(Q),'b', x,real(T),'r')
legend(…)
```



Problema particolare di Interpolazione Trigonometrica
(periodo T=2π e nodi equispaziati)

Assegnati $N+1$ nodi distinti x_k equispaziati in $[0, 2\pi[$ ed $N+1$ valori $y_k \in \mathbb{C}$, si cerca un polinomio trigonometrico interpolante $Q(x)$ [cioè tale che: $Q(x_k)=y_k$, $k=0,1,\ldots,N$] del tipo:

$$Q(x) = c_0 + c_1 e^{ix} + c_2 e^{2ix} + \cdots + c_N e^{iNx}$$

$N+1$ nodi x_k equispaziati in $[0, 2\pi[$: $x_k = 2\pi k/(N+1)$, $k=0,1,\ldots,N$

$x_k=$	0	$2\pi/(N+1)$	$4\pi/(N+1)$	$6\pi/(N+1)$	...	$2\pi(N-1)/(N+1)$	$2\pi N/(N+1)$

condizioni di interpolazione

$k=0,1,\ldots,N \quad Q(x_k) = y_k$

$$\begin{pmatrix} x_0 = 0 \\ x_1 = \frac{2\pi}{N+1} \\ x_2 = \frac{4\pi}{N+1} \\ \vdots \\ x_N = \frac{2\pi N}{N+1} \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & e^{i\frac{2\pi}{N+1}} & e^{i\frac{4\pi}{N+1}} & \cdots & e^{i\frac{2\pi N}{N+1}} \\ 1 & e^{i\frac{4\pi}{N+1}} & e^{i\frac{8\pi}{N+1}} & \cdots & e^{i\frac{4\pi N}{N+1}} \\ \vdots & & \vdots & \cdots & \vdots \\ 1 & e^{i\frac{2\pi N}{N+1}} & e^{i\frac{4\pi N}{N+1}} & \cdots & e^{i\frac{2\pi N^2}{N+1}} \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_N \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}$$

Come prima, si può sfruttare il Teorema di esistenza ed unicità del polinomio interpolante di Lagrange per determinare il polinomio trigonometrico interpolante Q

Teorema: Assegnati $N+1$ punti (x_k,y_k) con nodi equispaziati $x_k \in [0,2\pi[$, $x_k=2\pi k/(N+1)$ per $k=0,1,\ldots,N$, esiste un unico polinomio trigonometrico, di tipo $Q(x)=c_0+c_1e^{ix}+c_2e^{2ix}+\cdots+c_Ne^{iNx}$ e tale che $Q(x_k)=y_k \ \forall k=0,1,\ldots,N$ (interpolante), i cui coefficienti sono soluzioni del seguente sistema $Ac=y$:

A è una matrice particolare

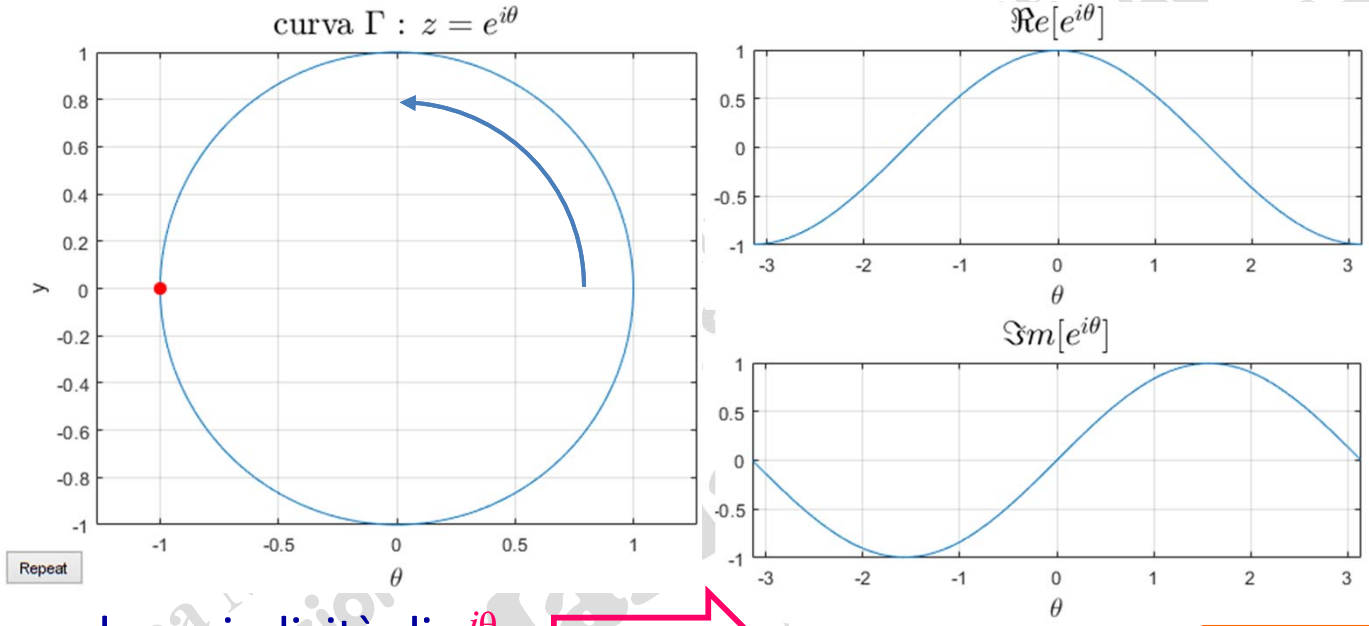
$$\begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & e^{i\frac{2\pi}{N+1}} & \left(e^{i\frac{2\pi}{N+1}}\right)^2 & \cdots & \left(e^{i\frac{2\pi}{N+1}}\right)^N \\ 1 & \left(e^{i\frac{2\pi}{N+1}}\right)^2 & \left(e^{i\frac{2\pi}{N+1}}\right)^4 & \cdots & \left(e^{i\frac{2\pi}{N+1}}\right)^{2N} \\ 1 & \left(e^{i\frac{2\pi}{N+1}}\right)^3 & \left(e^{i\frac{2\pi}{N+1}}\right)^6 & \cdots & \left(e^{i\frac{2\pi}{N+1}}\right)^{3N} \\ \vdots & & \vdots & \cdots & \vdots \\ 1 & \left(e^{i\frac{2\pi}{N+1}}\right)^N & \left(e^{i\frac{2\pi}{N+1}}\right)^{2N} & \cdots & \left(e^{i\frac{2\pi}{N+1}}\right)^{N^2} \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_N \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_N \end{pmatrix}$$

Esempio: funzione e^{ix}

Formula di Eulero* $e^{i\theta}=\cos(\theta)+i\sin(\theta)$

* scoperta intorno al 1740

```
syms th real;z=exp(i*th);ezplot3(real(z),imag(z),th,[-pi pi],'animate');view(2);axis equal
figure;fplot(real(z),[-pi pi]);axis equal; figure; fplot(imag(z),[-pi pi]);axis equal
```



$e^{i\theta}$ è periodica di periodo 2π

$e^{i\theta}$ percorre la circonferenza in verso antiorario.

$e^{-i\theta}$ percorre la circonferenza in verso orario.

per la periodicità di $e^{i\theta}$

$$\omega_M^{hk} = \left\{\left(e^{\pm\frac{2\pi i}{M}}\right)^{hk}\right\}_{h,k=0,\ldots,M-1} = \left\{e^{\pm\frac{2\pi i}{M}hk}\right\}_{h,k=0,\ldots,M-1} = \left\{e^{\pm\frac{2\pi i}{M}\bmod(hk,M)}\right\}_{h,k=0,\ldots,M-1}$$

`rem` restituisce il resto (mod) della divisione di hk per M

```
W1 =
[1,   1,  1,   1]
[1,  1i, -1, -1i]
[1,  -1,  1,  -1]
[1, -1i, -1,  1i]
```

```
PI=sym(pi);
M=4; h=0:M-1; k=0:M-1;
W1=exp(2i*PI/M*(h'*k))
W2=exp(2i*PI/M*rem(h'*k,M))
W2 =
[1,   1,  1,   1]
[1,  1i, -1, -1i]
[1,  -1,  1,  -1]
[1, -1i, -1,  1i]
```

Esempio

$$A = \left[\left(e^{\frac{2\pi i}{M}} \right)^{\mathrm{mod}(hk,M)} \right]_{h,k=0,1,\ldots,M-1}$$

$$\omega_M = e^{+i\frac{2\pi}{M}} = \cos\left(+\frac{2\pi}{M}\right) + i\sin\left(+\frac{2\pi}{M}\right)$$

$$\overline{\omega_M} = e^{-i\frac{2\pi}{M}} = \cos\left(-\frac{2\pi}{M}\right) + i\sin\left(-\frac{2\pi}{M}\right)$$

complessi
e
coniugati

```
PI=sym(pi); syms h k real
M=4; h=0:M-1; k=0:M-1;
A=exp(2i*PI/M*rem(h'*k,M))
A1=exp(-2i*PI/M*rem(h'*k,M))
fprintf('\nA*A1 =\n')
disp(A*A1)
```

```
A*A1 =
[4, 0, 0, 0]
[0, 4, 0, 0]
[0, 0, 4, 0]
[0, 0, 0, 4]
```

```
fprintf('\nA1*A =\n')
disp(A1*A)
```

```
A1*A =
[4, 0, 0, 0]
[0, 4, 0, 0]
[0, 0, 4, 0]
```

```
fprintf('\nA1*A/M =\n')
disp(A1*A/M)
```

```
A1*A/M =
[1, 0, 0, 0]
[0, 1, 0, 0]
[0, 0, 1, 0]
[0, 0, 0, 1]
```

$$\Rightarrow \quad \Omega_M^{-1} = \frac{1}{M}\overline{\Omega_M}$$

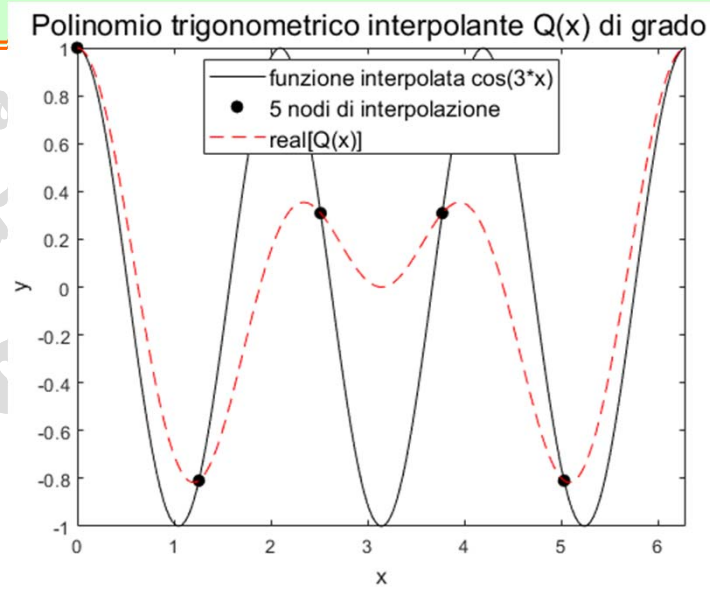
$$A^{-1} = \frac{1}{M}\left[\left(e^{-\frac{2\pi i}{M}} \right)^{\mathrm{mod}(hk,M)} \right]$$

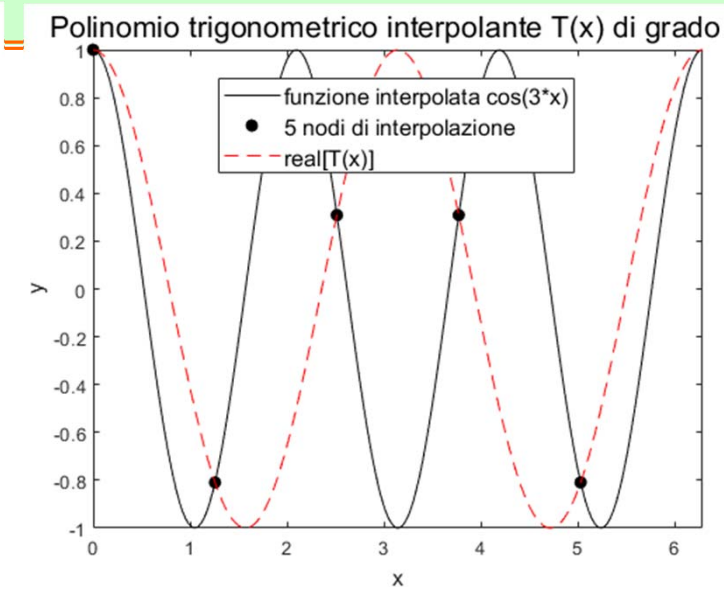
2° ALGORITMO

Per il polinomio $Q(x)$ la soluzione del sistema $A \cdot c = y$ è data direttamente da

$$c = \Omega_M^{-1} \cdot y = \frac{1}{M} \overline{\Omega_M} \cdot y$$

```
myfun=@(x) cos(3*x);  x=linspace(0,2*pi,199);  y=feval(myfun,x);
M=5; N=M-1; % M num. nodi, N grado polinomio
xk=linspace(0,2*pi,M+1)';  xk(end)=[];  yk=feval(myfun,xk);
k=0:N;  invA=exp(-2i*pi/M*mod(k'*k,M))/M;  c=invA*yk; % prodotto mat-vet
Q=polyval(flipud(c),exp(i*x)); plot(x,y,'k',xk,yk,'ok',x,real(Q),'r--')
axis tight
```





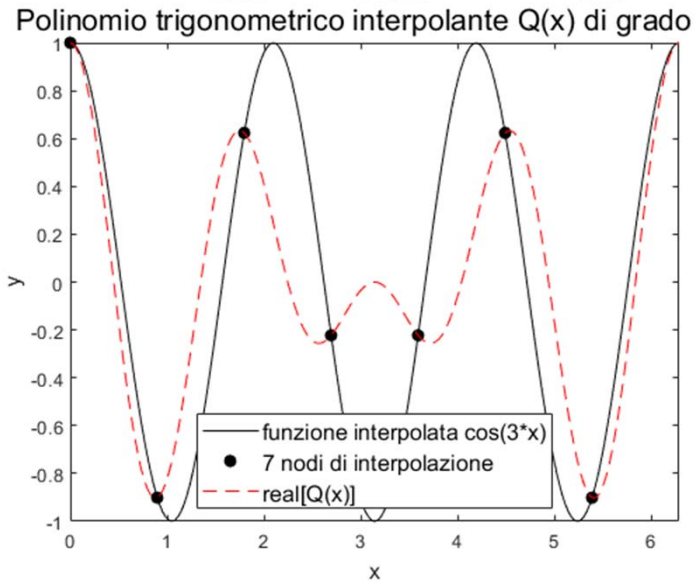
2° ALGORITMO

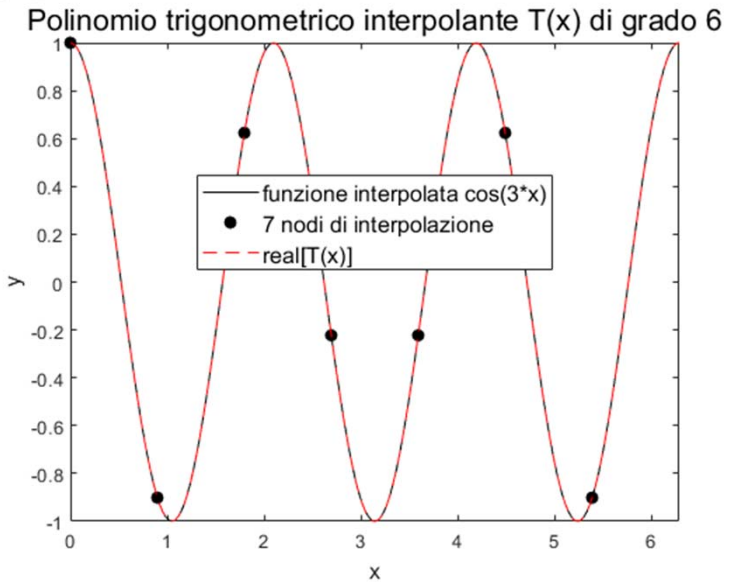
Per il polinomio $Q(x)$ la soluzione del sistema $A \cdot c = y$ è data direttamente da

$$c = \Omega_M^{-1} \cdot y = \frac{1}{M} \overline{\Omega_M} \cdot y$$

con soli 2 nodi in più

```
myfun=@(x) cos(3*x);  x=linspace(0,2*pi,199);  y=feval(myfun,x);
M=7; N=M-1; % M num. nodi, N grado polinomio
xk=linspace(0,2*pi,M+1)';  xk(end)=[];  yk=feval(myfun,xk);
k=0:N;  invA=exp(-2i*pi/M*mod(k'*k,M))/M;  c=invA*yk; % prodotto mat-vet
Q=polyval(flipud(c),exp(i*x)); plot(x,y,'k',xk,yk,'ok',x,real(Q),'r--')
axis tight
```





3° ALGORITMO

Posto $M = N+1$, il numero di punti di interpolazione, si ha

$$A = \left(a_{h,k}\right) = \left(e^{i\frac{2\pi}{M}}\right)^{hk} = \left(\overline{\omega_M}\right)^{hk} \qquad h,k=0,\ldots,M-1$$

dove $\omega_M = e^{-i\frac{2\pi}{M}}$

Cos'è la matrice A del sistema per l'interpolazione trigonometrica?

A meno del fattore moltiplicativo $1/M$, A è la matrice della **IDFT** Ω_M^{-1} ; cioè $A = M\left(\Omega_M\right)^{-1}$

Quindi la soluzione del sistema $A \cdot c = y$ è data direttamente da $c = A^{-1} \cdot y \quad \Rightarrow$

$$c = \frac{1}{M}\Omega_M \cdot y = \frac{1}{M}\mathrm{DFT}_M\left(y\right)$$

Cos'è una DFT ?

DFT sta per

Discrete Fourier Transform

cioè

Trasformata Discreta di Fourier

IDFT sta per

Inverse Discrete Fourier Transform

cioè

Trasformata Discreta Inversa di Fourier

Anticipazione: Discrete Fourier Transform (DFT)

sarà trattata nella parte II del corso

DEFINIZIONE: $\underline{\mathbf{F}}$ è la DFT di $\underline{\mathbf{f}}$ $\qquad \underline{\mathbf{F}} = \mathbf{DFT}(\underline{\mathbf{f}})$

Input: $\underline{\mathbf{f}} = (f_0, f_1, f_2, \ldots, f_{N-1})^T$ vettore reale o complesso ($\underline{\mathbf{f}} \in \mathbb{C}^N$)

Output: $\underline{\mathbf{F}} = (F_0, F_1, F_2, \ldots, F_{N-1})^T$ $\qquad (\underline{\mathbf{F}} \in \mathbb{C}^N)$

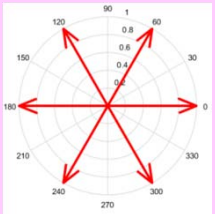
$$\underline{\mathbf{F}} = \mathbf{DFT}(\underline{\mathbf{f}}) = \Omega_N \cdot \underline{\mathbf{f}}$$ forma matriciale DFT

dove Ω_N è la matrce quadrata $\Omega_N = \left(\omega_N^{kj}\right)_{k,j=0,1,\ldots,N-1}$ i cui elementi sono le potenze di $\omega_N = e^{-i\frac{2\pi}{N}}$

(ω_N è una radice N sima dell'unità primitiva*)

* Una radice N sima dell'unità primitiva z è una radice N sima di 1 tale che le potenze $\{(z)^k\}_{k=0,\ldots,N-1}$ danno tutte le radici N sime di 1 ζ :

$$\zeta = \sqrt[N]{1} = \sqrt[N]{[\rho = 1,\ \theta = 0]} = \left[1, \frac{2\pi}{N} k\right]_{k=0,1,\ldots,N-1} = \left\{e^{ik\frac{2\pi}{N}}\right\}_{k=0,1,\ldots,N-1}$$



$N=6$

dal prodotto matrice-vettore

$$F_k = \sum_{j=0}^{N-1} f_j\, e^{-\frac{2\pi i}{N} kj} = \sum_{j=0}^{N-1} f_j\, \omega_N^{kj}, \qquad k=0,1,\ldots,N-1$$ forma scalare DFT

Esempio

$$\underline{\mathbf{F}} = \mathrm{DFT}\left(\underline{\mathbf{f}}\right) = \Omega_N \underline{\mathbf{f}}$$

$$\Omega_N = \left[\left(\omega_N\right)^{kj}\right]_{k,j=0,1,\ldots,N-1} \quad \text{dove} \quad \omega_N = e^{-i\frac{2\pi}{N}}$$

N=4 $\quad \omega_4 = e^{-i\frac{\pi}{2}} = -i$

$$i^0 = 1$$
$$i^1 = i$$
$$i^2 = -1$$
$$i^3 = -i$$

per la periodicità di i^p ⇒

$$\Omega_4 = \left(-i^{hk}\right)_{h,k=0,1,2,3} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & (-i)^2 & (-i)^3 \\ 1 & (-i)^2 & (-i)^4 & (-i)^6 \\ 1 & (-i)^3 & (-i)^6 & (-i)^9 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{pmatrix}$$

gli elementi di Ω_N sono le N radici $N^{\underline{\text{sime}}}$ dell'unità

ω_N è una radice 4ª primitiva

```
N=4; w=exp(sym(-2*pi*i/N))
w =
-i
k=0:N-1; W=w.^(k'*k)
W =
[ 1,   1,   1,   1]
[ 1,  -i,  -1,   i]
[ 1,  -1,   1,  -1]
[ 1,   i,  -1,  -i]
```

prodotto esterno

Ω_N è simmetrica

Principale proprietà di Ω_N: il calcolo della matrice inversa è immediato!

... e quindi non va calcolata

```
N=…; w=exp(sym(-2*pi*i/N)); k=0:N-1;
W=w.^(k'*k);
WW=simplify(W*W), D=simplify(W'*W)
```

N=2

```
WW =
[ 2, 0]
[ 0, 2]
D =
[ 2, 0]
[ 0, 2]
```

N=3

```
WW =
[ 3, 0, 0]
[ 0, 0, 3]
[ 0, 3, 0]
D =
[ 3, 0, 0]
[ 0, 3, 0]
[ 0, 0, 3]
```

N=4

```
WW =
[ 4, 0, 0, 0]
[ 0, 0, 0, 4]
[ 0, 0, 4, 0]
[ 0, 4, 0, 0]
D =
[ 4, 0, 0, 0]
[ 0, 4, 0, 0]
[ 0, 0, 4, 0]
[ 0, 0, 0, 4]
```

N=5

```
WW =
[ 5, 0, 0, 0, 0]
[ 0, 0, 0, 0, 5]
[ 0, 0, 0, 5, 0]
[ 0, 0, 5, 0, 0]
[ 0, 5, 0, 0, 0]
D =
[ 5, 0, 0, 0, 0]
[ 0, 5, 0, 0, 0]
[ 0, 0, 5, 0, 0]
[ 0, 0, 0, 5, 0]
[ 0, 0, 0, 0, 5]
```

$$\Omega_N^{-1} = \frac{1}{N}\left(\Omega_N\right)^{\mathrm{H}} = \frac{1}{N}\bar{\Omega}_N = \frac{1}{N}\left(\omega_N^{-kj}\right)_{k,j=0,1,\ldots,N-1}$$

H sta per matrice trasposta e complessa coniugata

poiché la matrice è simmetrica, Ω_N^{H} si riduce alla sola complessa coniugata

Inverse Discrete Fourier Transform (IDFT)

DEFINIZIONE: $\underline{\mathbf{f}}$ è la IDFT di $\underline{\mathbf{F}}$ $\quad \underline{\mathbf{F}} = \mathrm{IDFT}(\underline{\mathbf{f}})$

Input: $\underline{\mathbf{F}} = (F_0, F_1, F_2, \ldots, F_{N-1})^{\mathsf{T}}$ vettore reale o complesso ($\underline{\mathbf{F}} \in \mathbb{C}^N$)

Output: $\underline{\mathbf{f}} = (f_0, f_1, f_2, \ldots, f_{N-1})^{\mathsf{T}}$ ($\underline{\mathbf{f}} \in \mathbb{C}^N$)

$$\underline{\mathbf{f}} = \mathrm{IDFT}(\underline{\mathbf{F}}) = \Omega_N^{-1} \underline{\mathbf{F}}$$

forma matriciale IDFT

dove Ω_N^{-1} è l'inversa della matrice Ω_N ed è data da:

(molto facile da calcolare)

$$\Omega_N^{-1} = \frac{1}{N}(\Omega_N)^{\mathsf{H}} = \frac{1}{N}\bar{\Omega}_N = \frac{1}{N}\left(\omega_N^{-kj}\right)_{k,j=0,1,\ldots,N-1}$$

matrice complessa coniugata (poiché Ω_N è simmetrica)

$$\omega_N = e^{-i\frac{2\pi}{N}}$$

$$\bar{\omega}_N = e^{+i\frac{2\pi}{N}}$$

dal prodotto matrice-vettore:

$$f_j = \frac{1}{N}\sum_{k=0}^{N-1} F_k e^{+\frac{2\pi i}{N}kj}, \qquad j=0,1,\ldots,N-1$$

forma scalare IDFT

Esempio

$$\underline{\mathbf{f}} = \mathrm{IDFT}\left(\underline{\mathbf{F}}\right) = \Omega_N^{-1}\underline{\mathbf{F}}$$

$$\Omega_N = \left[\left(\omega_N\right)^{kj}\right]_{k,j=0,1,\ldots,N-1} \quad \text{dove} \quad \omega_N = e^{-i\frac{2\pi}{N}}$$

N=4 $\quad \omega_4 = e^{-i\frac{\pi}{2}} = -i$

$$\Omega_4 = \left(-i^{hk}\right)_{h,k=0,1,2,3} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{pmatrix} \Rightarrow \Omega_4^{-1} = \frac{1}{4}\left[\left(\overline{-i}\right)^{hk}\right]_{h,k=0,1,2,3} = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}$$

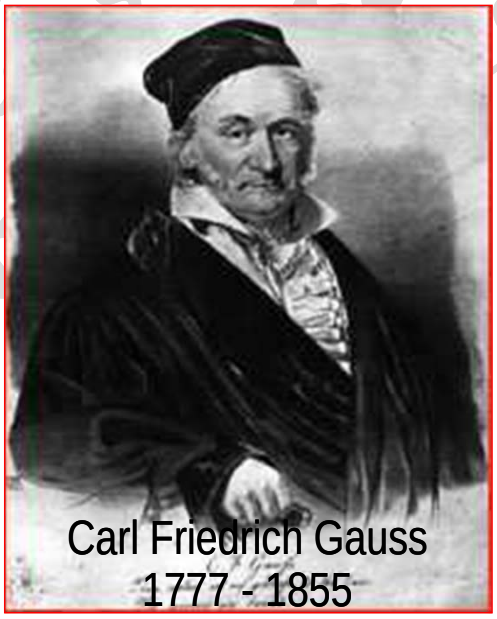
```
syms f0 f1 f2 f3; f=[f0;f1;f2;f3];
N=4; w=exp(sym(-2*pi*i/N));
k=0:N-1; W=w.^(k'*k);
disp([W   inv(W)])
[ 1,   1,  1,  1,      1/4,  1/4,   1/4,  1/4 ]
[ 1,  -i, -1,  i,      1/4,  i/4,  -1/4, -i/4 ]
[ 1,  -1,  1, -1,      1/4, -1/4,   1/4, -1/4 ]
[ 1,   i, -1, -i,      1/4, -i/4,  -1/4,  i/4 ]
F=W*f; disp((1/4*conj(W)*F-f).')
[ 0, 0, 0, 0]
```

prodotto esterno

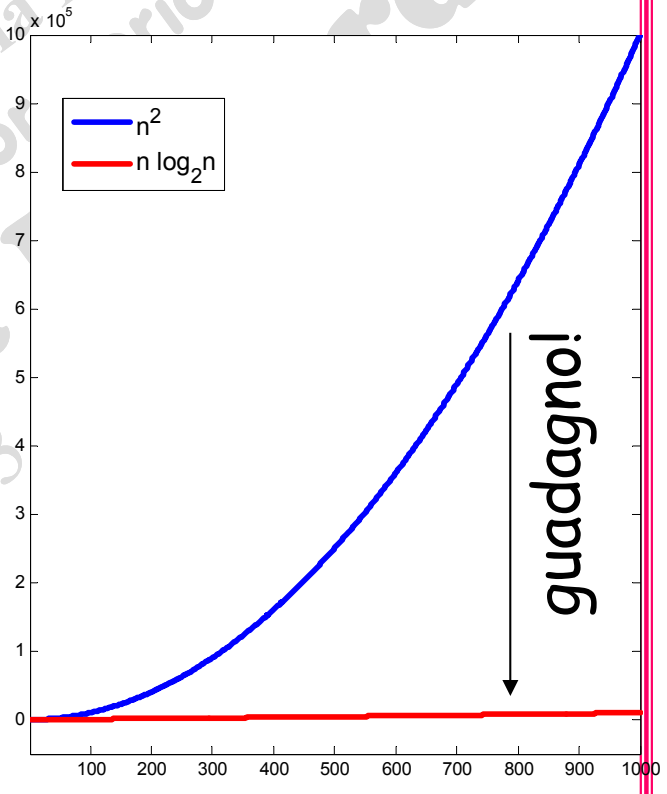
Algoritmo FFT (*Cooley-Tuckey*, 1965)
Fast Fourier Transform

Ha ridotto la complessità computazionale del calcolo di una **DFT** da n^2 a $n \cdot \log_2 n$, consentendo, in ambiente numerico, un'ampia utilizzazione dei metodi matematici basati su Fourier. Attualmente esiste una famiglia di algoritmi FFT: ciascuno applicabile a dati (ed architetture) particolari.

Jean Baptiste Joseph Fourier
1768 - 1830



Carl Friedrich Gauss
1777 - 1855



in MATLAB:
fft(...) e ifft(...)

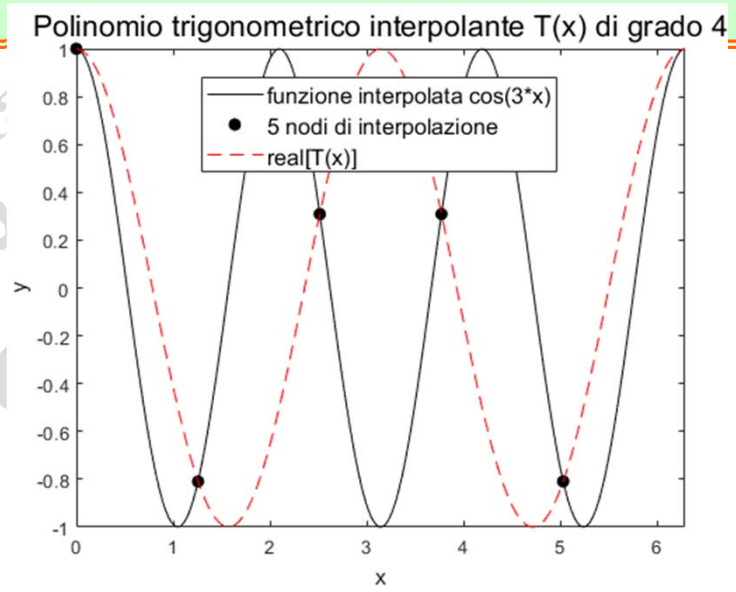
Questo algoritmo era già noto a Gauss all'incirca nel 1805.

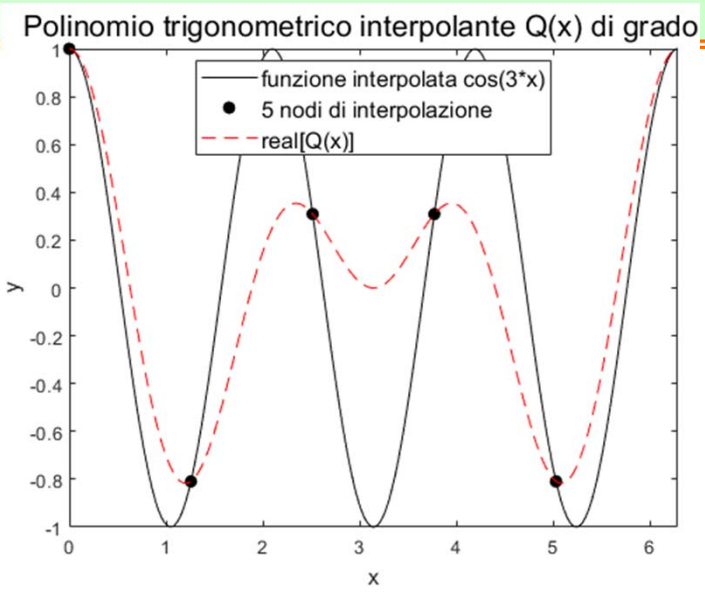
3° ALGORITMO: esempio

Per il polinomio $Q(x)$ la soluzione del sistema $A \cdot c = y$ è data direttamente da

$$c = \frac{1}{M}\Omega_M \cdot y = \frac{1}{M}\mathrm{DFT}_M(y)$$

```
myfun=@(x) cos(3*x);  x=linspace(0,2*pi,199);  y=feval(myfun,x);
M=5; N=M-1; % M num. nodi, N grado polinomio
xk=linspace(0,2*pi,M+1)';  xk(end)=[];  yk=feval(myfun,xk);
c=fft(yk)/M; % mediante DFT
Q=polyval(flipud(c),exp(i*x)); plot(x,y,'k',xk,yk,'ok',x,real(Q),'r--')
axis tight
```





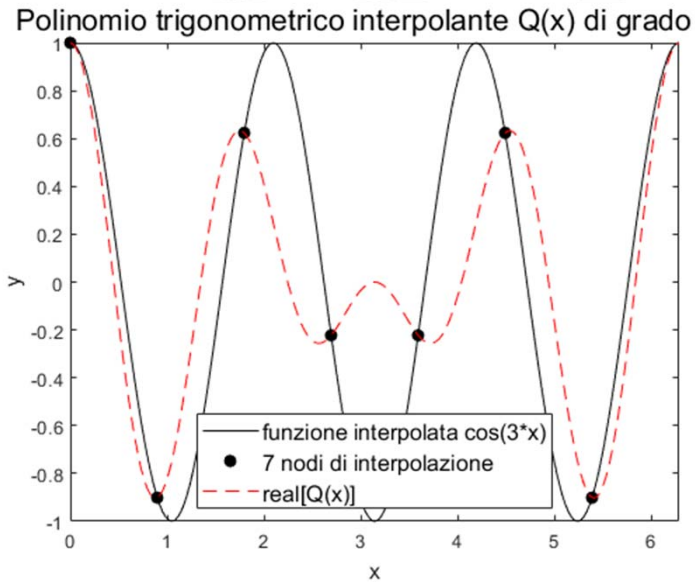
3° ALGORITMO

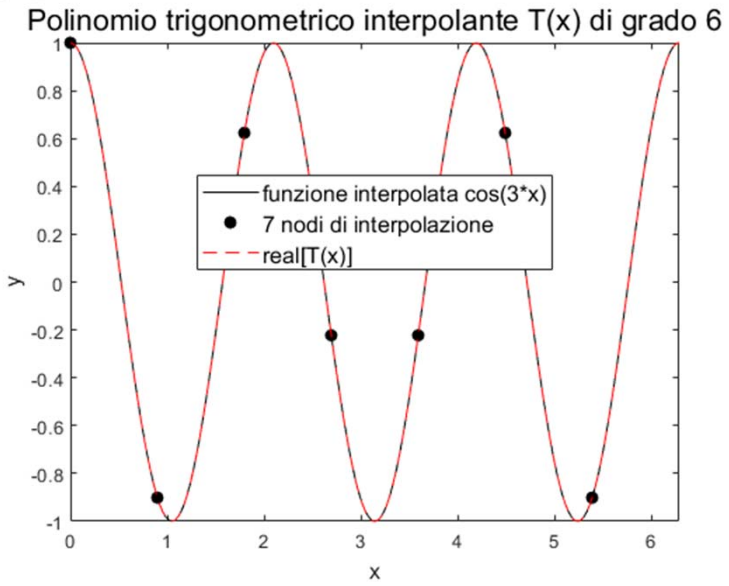
Per il polinomio $Q(x)$ la soluzione del sistema $A \cdot c = y$ è data direttamente da

$$c = \frac{1}{M}\Omega_M \cdot y = \frac{1}{M}\mathrm{DFT}_M(y)$$

con soli 2 nodi in più

```
myfun=@(x) cos(3*x);  x=linspace(0,2*pi,199);  y=feval(myfun,x);
M=7; N=M-1; % M num. nodi, N grado polinomio
xk=linspace(0,2*pi,M+1)';  xk(end)=[];  yk=feval(myfun,xk);
c=fft(yk)/M; % mediante DFT
Q=polyval(flipud(c),exp(i*x)); plot(x,y,'k',xk,yk,'ok',x,real(Q),'r--')
axis tight
```





I coefficienti del polinomio trigonometrico interpolante $Q(x)$ sugli $M=N+1$ punti (x_k,y_k), con i nodi x_k equispaziati, sono dati da (analogamente per il polinomio $T(x)$):

- Algoritmo 1 (metodo di Gauss): sistema $A \cdot c = y$

  con complessità $O(M^3/3)$

- Algoritmo 2 (prodotto mat-vet): $c = A^{-1} \cdot y = \frac{1}{M} \bar{A} \cdot y$

  con complessità $O(M^2)$

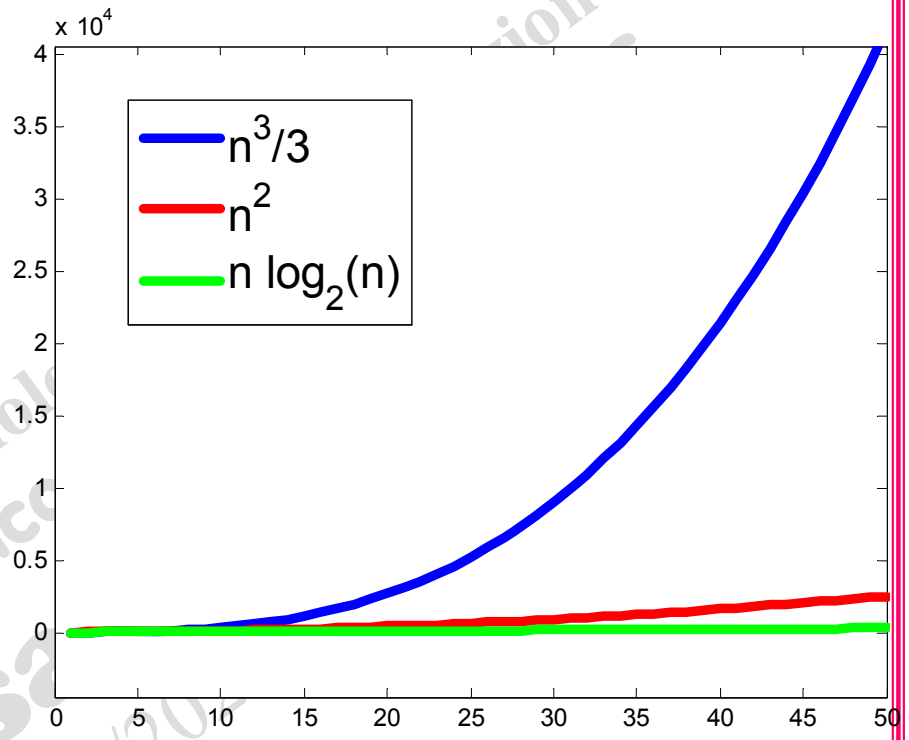
- Algoritmo 3 (DFT):

  $$c = \frac{1}{M} \mathrm{DFT}_M(y)$$

  con complessità $O(M \log_2 M)$

il più efficiente

3 algoritmi per determinare i coefficienti del polinomio trigonometrico interpolante su n nodi:

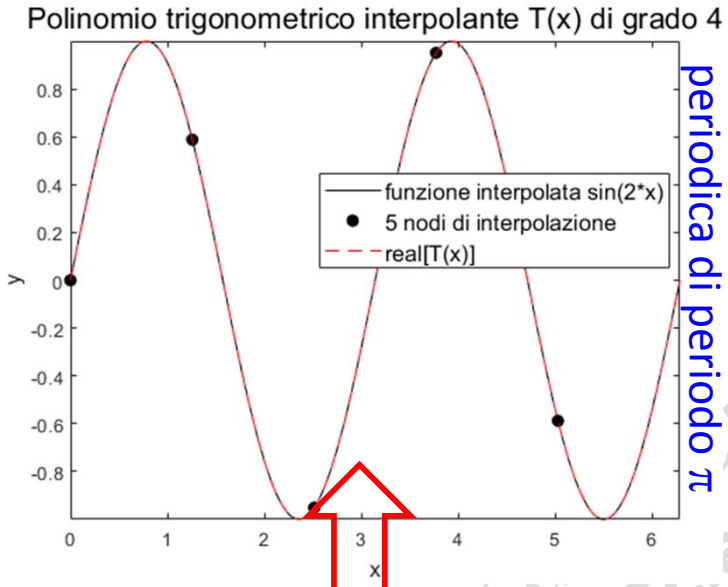


1. [per n nodi qualsiasi] Si risolve il sistema lineare con l'algoritmo di Gauss: allora la complessità computazionale è $O(n^3/3)$;

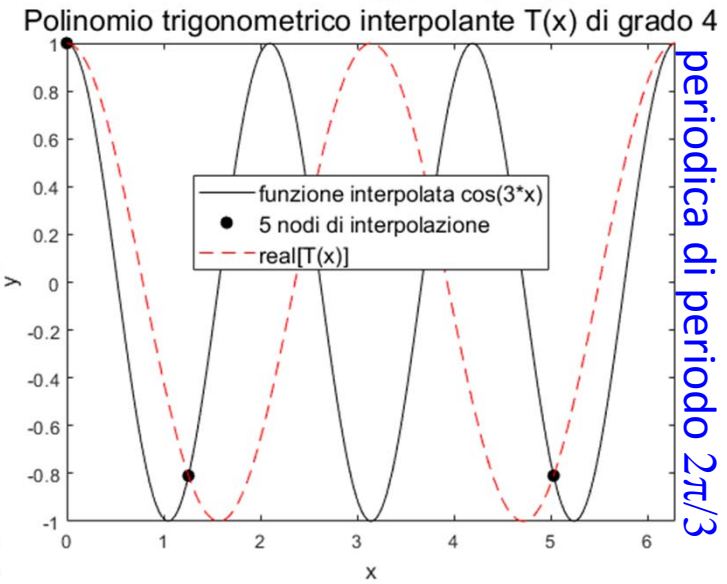
2. [per n nodi equispaziati] Conoscendo l'inversa della matrice del sistema, si esegue il prodotto matrice×vettore: la complessità computazionale è $O(n^2)$;

3. [per n nodi equispaziati] La complessità computazionale si riduce a $O(n \cdot \log_2 n)$ ricorrendo all'algoritmo FFT per il calcolo di una DFT.

Algoritmi 1,2,3 su nodi equispaziati in $[0,2\pi[$

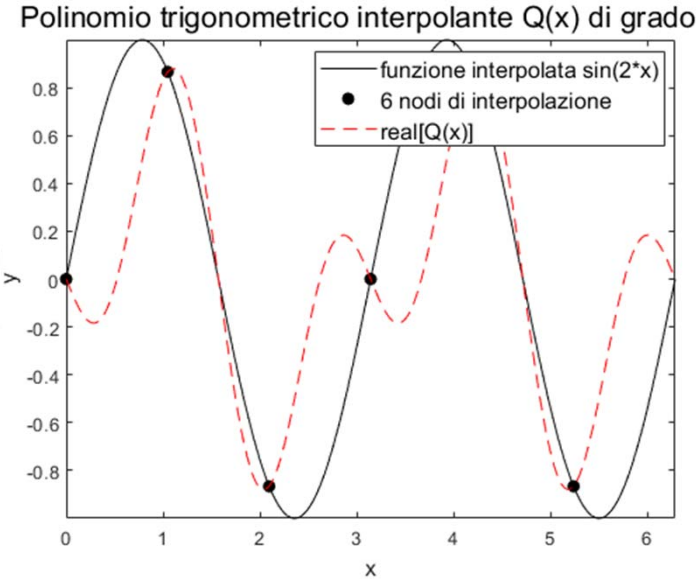


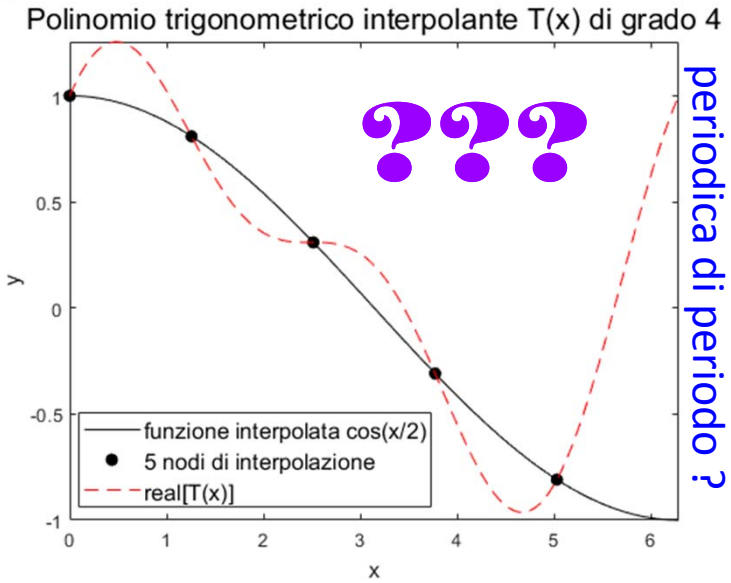
periodica di periodo π



periodica di periodo $2\pi/3$

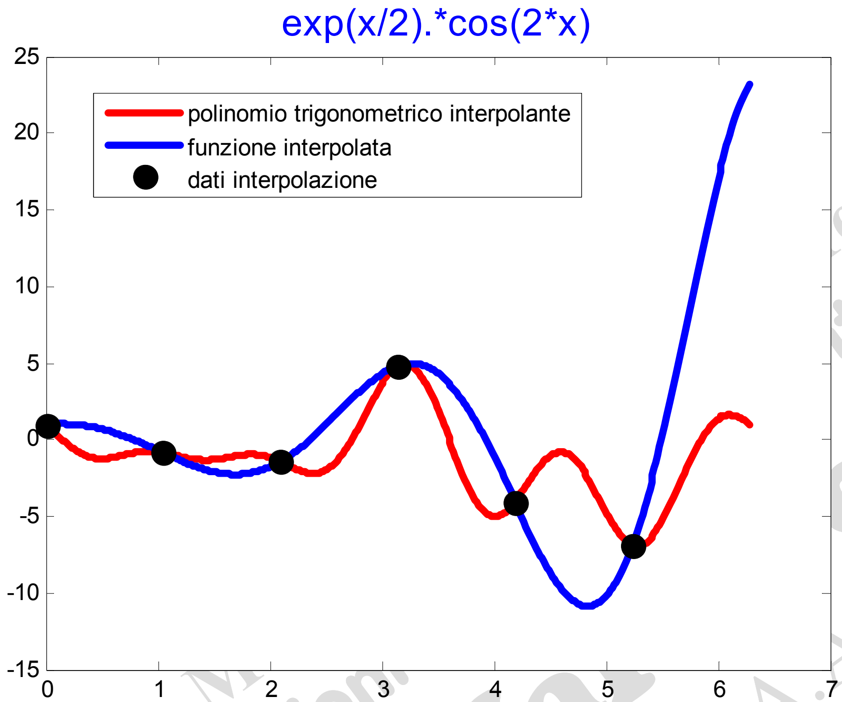
I risultati sono gli stessi per i 3 algoritmi;
l'unica cosa che cambia è il tempo

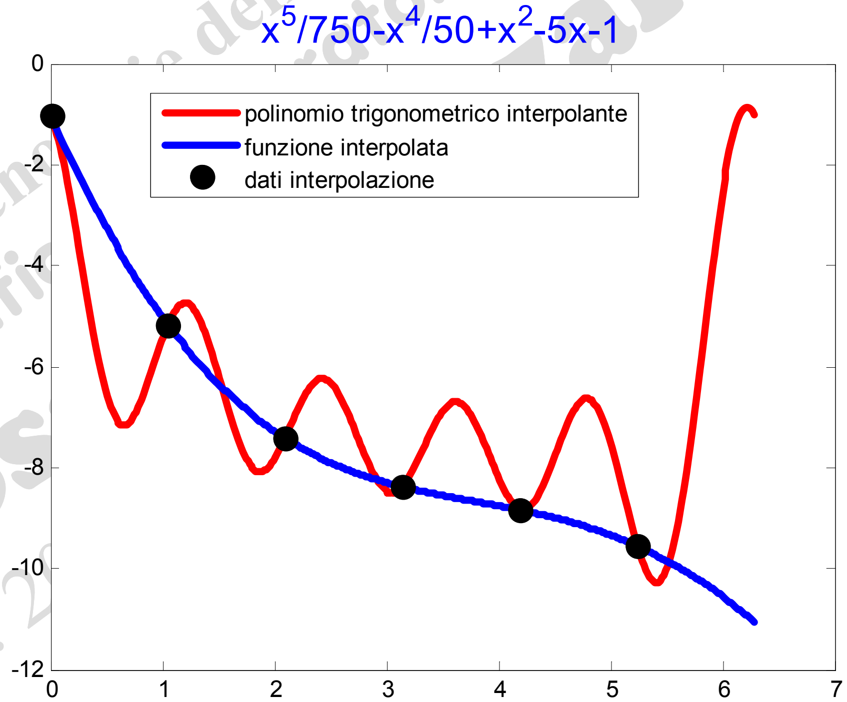




periodica di periodo ?

… e se i dati di interpolazione provengono da una funzione non periodica …





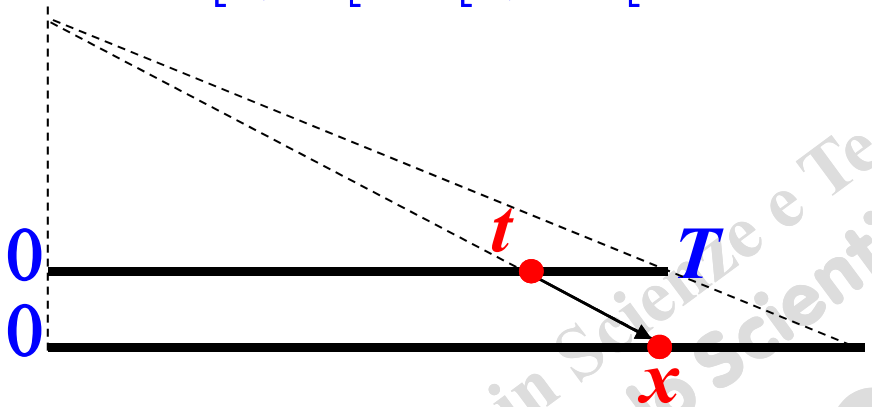
???

???

Non bisogna usare l'interpolazione trigonometrica!!!

Interpolazione Trigonometrica di periodo $T=b-a$

L'intervallo $[0, 2\pi[$ **non** è una limitazione: se si interpolano dati periodici nell'intervallo $[0, T[$ è sufficiente "proiettare" $[0, T[$ in $[0, 2\pi[$:



$$t \in [0,T[\longrightarrow x = \frac{2\pi}{T} t \in [0,2\pi[$$

Se più in generale si interpolano dati periodici in un qualsiasi intervallo $[a,b[$: prima si trasla l'origine in a e poi si proietta $[0, b-a[$ in $[0, 2\pi[$

$$t \in [a,b[\longrightarrow x = \frac{2\pi}{b-a}(t-a) \in [0,2\pi[$$

… è una affinità

Problema di Interpolazione Trigonometrica
(intervallo $[a, b[$, periodo $T = b-a$)

Assegnati M nodi distinti t_k equispaziati in $[a, b[$ ed M valori y_k, il polinomio trigonometrico interpolante, di grado $N=M-1$, del tipo:

$$Q(x) = c_0 + c_1 e^{ix} + c_2 e^{i2x} + c_3 e^{i3x} + \cdots + c_N e^{iNx} \qquad (N \text{ qualsiasi})$$

$$T_N(x) = \gamma_{-\frac{N}{2}} e^{-i\frac{N}{2}x} + \cdots + \gamma_{-1} e^{-ix} + \gamma_0 + \gamma_{+1} e^{ix} + \cdots + \gamma_{+\frac{N}{2}} e^{i\frac{N}{2}x} \qquad (N \text{ pari})$$

si ottiene con la sostituzione: $x = \dfrac{2\pi}{b-a}(t-a)$

$$Q(t) = c_0 + c_1 e^{i\frac{2\pi}{b-a}(t-a)} + c_2 e^{2i\frac{2\pi}{b-a}(t-a)} + c_3 e^{3i\frac{2\pi}{b-a}(t-a)} + \cdots + c_N e^{iN\frac{2\pi}{b-a}(t-a)}$$

$$T_N(t) = \gamma_{-\frac{N}{2}} e^{-i\frac{N}{2}\frac{2\pi}{b-a}(t-a)} + \cdots + \gamma_{-1} e^{-i\frac{2\pi}{b-a}(t-a)} + \gamma_0 + \gamma_{+1} e^{i\frac{2\pi}{b-a}(t-a)} + \cdots + \gamma_{+\frac{N}{2}} e^{i\frac{N}{2}\frac{2\pi}{b-a}(t-a)}$$

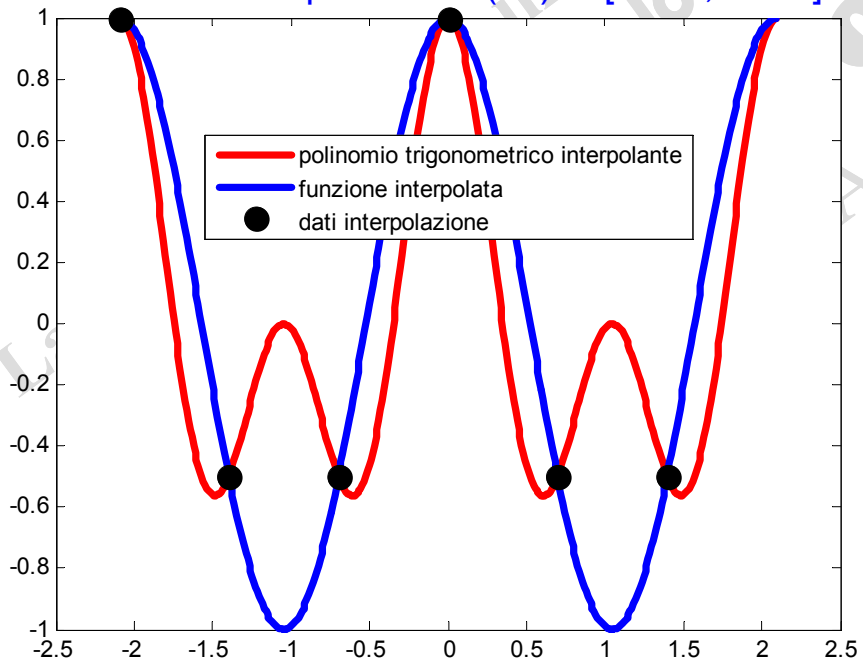
Interpolazione Trigonometrica in $[a,b[$ di periodo $T=b-a$

```
xk=linspace(a,b-(b-a)/M, M)';
```

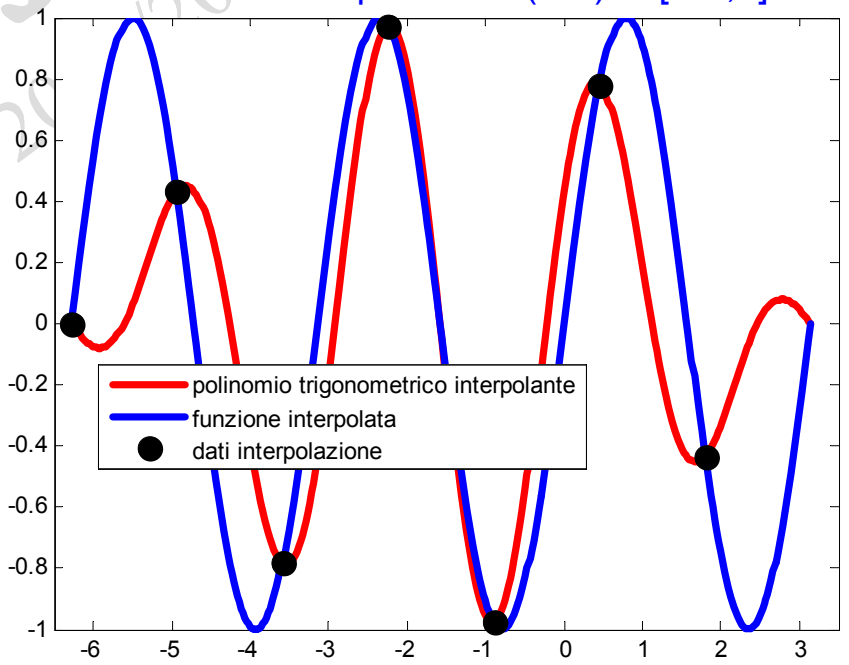
```
fun=str2func(['@(x)' input('stringa funzione: f(x) = ','s')]);
a=input('a = '); b=input('b = '); x=linspace(a,b,401); y=feval(fun,x);
M=input('numero nodi '); xk=linspace(a,b,M+1)'; xk(end)=[ ];
yk=feval(fun,xk);
c = fft(yk)/M;
Q = polyval(flipud(c), exp(i*2*pi*(x-a)/(b-a)));
plot(x,y,'b',xk,yk,'ok',x,real(Q),'r')
```

unica modifica

funzione interpolata cos(3x) in [-2/3π,+2/3π]



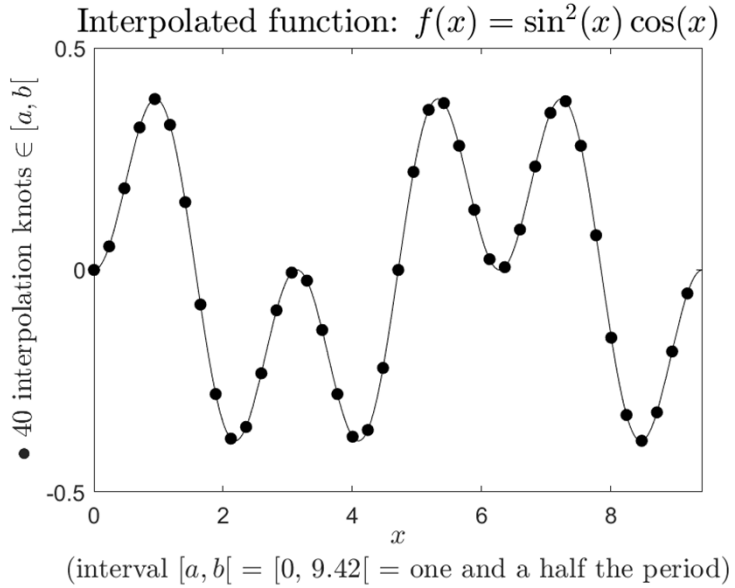
funzione interpolata sin(2*x) in [-2π,π]



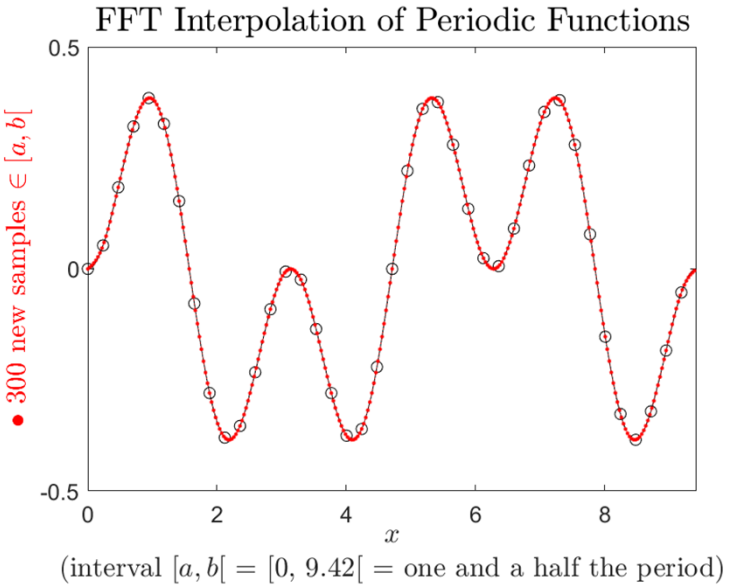
Interpolazione trigonometrica in MATLAB

`y=interpft(fj,N)` interpolazione mediante *Trasformata di Fourier*. Usa la *Trasformata di Fourier* dei valori **`fj`**, in **`n`** nodi equispaziati, per produrre **`N`** valori in punti equispaziati. Se **`fj`** è una matrice agisce su ogni colonna, altrimenti usare **`y=interpft(fj,N,dim)`** per scegliere la dimensione su cui agire.

```
pf=@(t) sin(t).^2.*cos(t);      % funzione interpolata
str='$f(x)=\sin^2(x)\cos(x)$'; % formula LaTeX per i titoli
a=0; b=3*pi; L=b-a; n=40; dx=L/n; xj=a+dx*(0:(n-1))'; fj=pf(xj); % nodi di interp.
figure; fplot(pf,[a b],'Color','k'); hold on
plot(xj,fj,'ok','MarkerFaceColor','k'); xlabel(…); ylabel(…); title(…)
N=300;  Xj=a+L/N*(0:(N-1))';  Fj=interpft(fj,N);
figure; fplot(pf,[a b],'LineStyle',':','Color','k'); hold on
plot(xj,fj,'ok',Xj,Fj,'.r'); xlabel(…); ylabel(…); title(…)
```



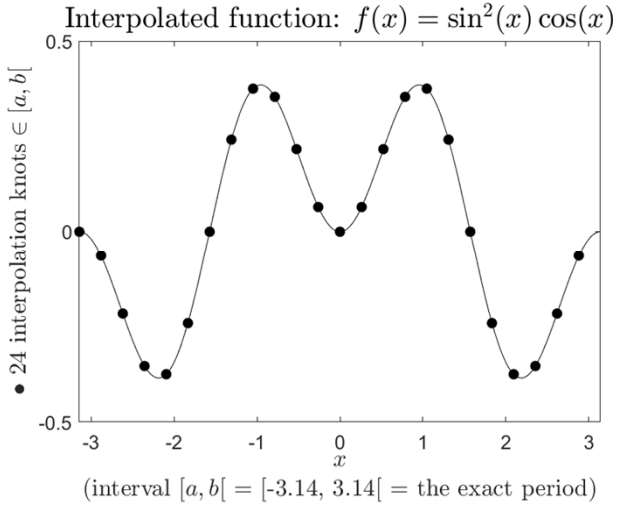
(interval $[a, b[= [0, 9.42[$ = one and a half the period)



(interval $[a, b[= [0, 9.42[$ = one and a half the period)

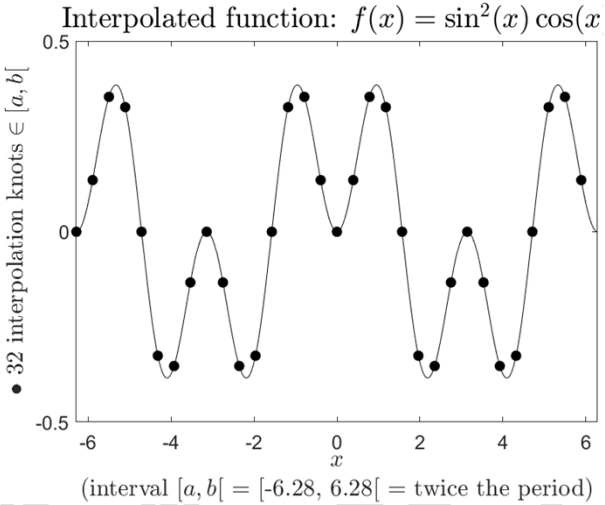
Interpolazione trigonometrica in MATLAB

in input: n nodi di interpolazione equispaziati in $[a, b[$



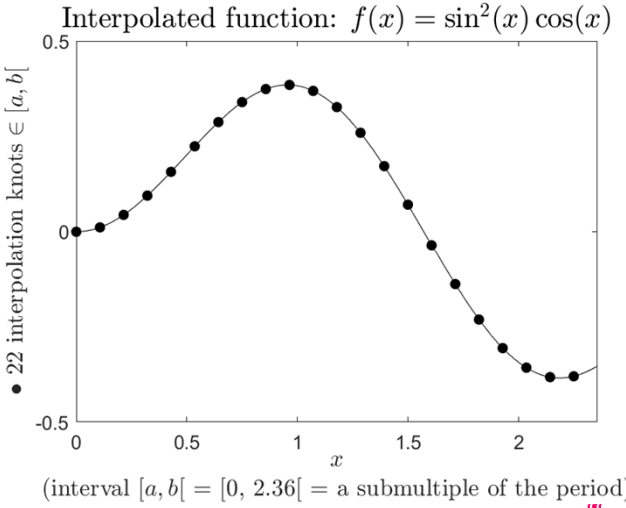
(interval $[a, b[$ = [-3.14, 3.14[= the exact period)

```
a=-pi; b=+pi;
n=24; N=100;
```



(interval $[a, b[$ = [-6.28, 6.28[= twice the period)

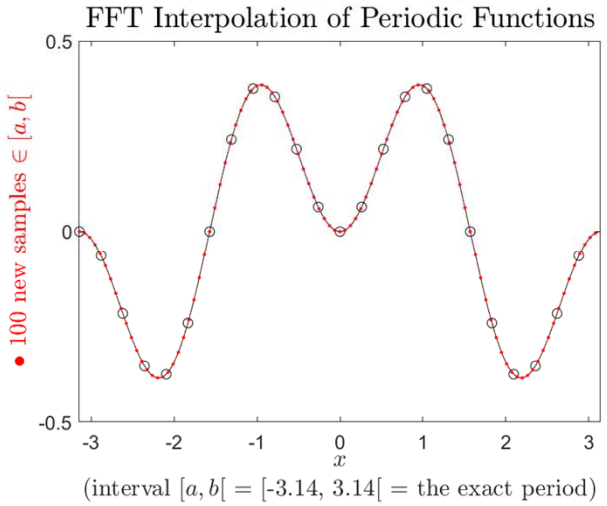
```
a=-2*pi; b=+2*pi;
n=32; N=200;
```



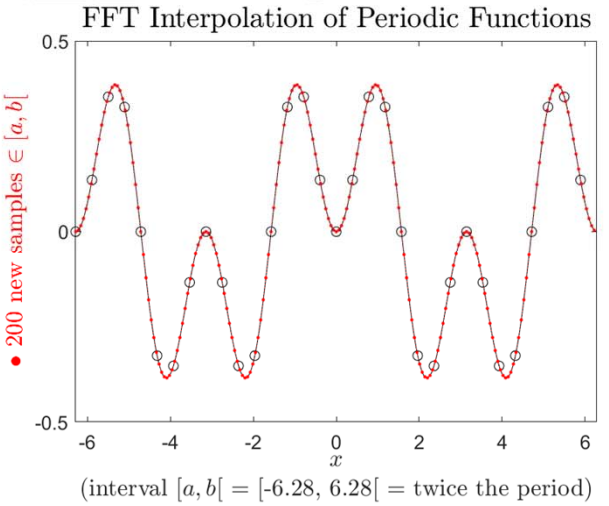
(interval $[a, b[$ = [0, 2.36[= a submultiple of the period)

```
a=0; b=3/4*pi;
n=22; N=100;
```

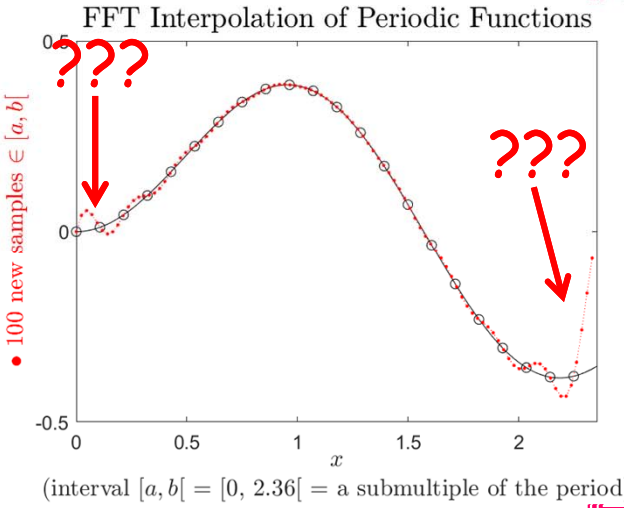
in output: N valori in nodi equispaziati in $[a, b[$



(interval $[a, b[$ = [-3.14, 3.14[= the exact period)



(interval $[a, b[$ = [-6.28, 6.28[= twice the period)



(interval $[a, b[$ = [0, 2.36[= a submultiple of the period)

Interpolazione trigonometrica in MATLAB

Idea dell'algoritmo*

* l'algoritmo di **interpft** è leggermente diverso; usare **edit interpft** nella Command Window per vederne il codice

Y=interpft(fj,N) trasforma, mediante la *Trasformata di Fourier*, il vettore degli **n** campioni **fj** dallo spazio del tempo a quello delle frequenze e, introducendo degli zeri, ne aumenta il numero a **N**, con **N>n**. Poi inverte la nuova *Trasformata di Fourier*.

per semplicità l'esempio suppone **n** e **N** pari

```
a=0; b=3*pi; n=32; xj=linspace(a,b,n)'; fj=pf(xj);
N=300; Y=interpft(fj,N);  % per confronto
if rem(n,2) == 1 % n dispari
    fj=[mean(fj([1 end])); fj(2:end-1)];
else                  % n pari
    fj(1)=mean(fj([1 end])); fj(end)=fj(1);
end
a=fftshift(fft(fj)); a=[a; a(1)];
nyqst=floor(n/2); nu=(-nyqst:nyqst)';
figure; stem(nu,abs(a),'ob'); AX=axis; axis([-nyqst-1 nyqst+1 -0.1 AX(4)])
hold on; plot([-nyqst nyqst],abs(a([1 end])),'or')
b=[zeros((N-n)/2,1); a; zeros((N-n)/2,1)]; % allunga il vettore
figure; stem((-N/2:N/2)',abs(b),'.c')
AX=axis; axis([-N/2-1 N/2+1 -0.1 AX(4)])
hold on; stem(nu,abs(a),'ob-.','MarkerSize',4)
mid=N/2+1; plot([-nyqst nyqst],abs(b([mid-nyqst mid+nyqst])),'.r')
plot([-nyqst nyqst],abs(a([1 end])),'or','MarkerSize',4)
b=b(1:N); y=ifft(fftshift(b)); y=y*N/n;
if isreal(fj),  y=real(y);  end
fprintf('\nY=interpft(fj,N) e y da Fourier Transform:')
fprintf('\nmax(abs(Y - y)) = %g\n', max(abs(Y-y)))
Y=interpft(fj,N)  e y da Fourier Transform:
max(abs(Y - y)) = 1.11022e-16
```

FT

IFT

Errore assoluto