

# Laboratorio di Analisi Numerica

## Lezione 10

Leonardo Robol <leonardo.robol@unipi.it>

Igor Simunec <igor.simunec@sns.it>

3 Dicembre 2021

**Quantità di esercizi:** in questa dispensa ci sono *più esercizi* di quanti uno studente medio riesca a farne durante una lezione di laboratorio, specialmente tenendo conto anche degli esercizi facoltativi. Questo è perché è pensata per “tenere impegnati” per tutta la lezione anche quegli studenti che già hanno un solido background di programmazione. Quindi fate gli esercizi che riuscite, partendo da quelli *non* segnati come facoltativi, e non preoccupatevi se non li finite tutti!

### 1 DFT in Matlab

La funzione in MATLAB che calcola la trasformata discreta di Fourier (DFT) con l'algoritmo FFT probabilmente usa una definizione leggermente diversa rispetto a quella che avete usato a lezione. La funzione `fft(a)` prende un vettore (riga o colonna) di lunghezza  $n$  contenente i numeri complessi  $[a_0 \ a_1 \ \dots \ a_{n-1}]$  e restituisce i valori assunti dal polinomio  $a(x) = \sum_{j=0}^{n-1} a_j x^j$  (occhio all'ordine dei coefficienti) nelle radici  $n$ -esime dell'unità  $\zeta^k$ ,  $k = 0, 1, \dots, n-1$  (*valutazione*). La funzione `ifft(y)` esegue l'operazione inversa: prende un vettore contenente i valori  $y_k = a(\zeta^k)$ , con  $k = 0, 1, \dots, n-1$ , e calcola i coefficienti del polinomio  $a(x) = \sum_{j=0}^{n-1} a_j x^j$  (*interpolazione*).

*Esercizio 1.* Controllare quanto scritto qui sopra provando con alcuni esempi di piccola dimensione, in cui riuscite a verificare i conti a mano. Per esempio calcolate `fft([0 0 1])`, `fft([0 1 0 0])`, `ifft([1 1 1 1])`. Notate in che ordine MATLAB utilizza le radici dell'unità; è quello che vi aspettavate?

*Esercizio 2.* Usando la DFT (IDFT), scrivete una funzione `function c=polymul(a,b)` che calcoli il prodotto di due polinomi (dati come i vettori dei loro coefficienti). Hint: quale dimensione  $n$  vi serve perché “ci sia spazio” per tutti i coefficienti del prodotto? Quando i vettori  $a$  e  $b$  sono troppo corti vanno “allungati” con un numero opportuno di zeri.

## 1.1 Matrici di Toeplitz

Una matrice si chiama *matrice di Toeplitz* se i suoi elementi  $a_{ij}$  dipendono solo dalla differenza  $i - j$ ; cioè, se sono costanti lungo le diagonali parallele a quella principale:

$$\begin{bmatrix} a_0 & a_1 & a_2 & a_3 & a_4 \\ a_{-1} & a_0 & a_1 & a_2 & a_3 \\ a_{-2} & a_{-1} & a_0 & a_1 & a_2 \\ a_{-3} & a_{-2} & a_{-1} & a_0 & a_1 \\ a_{-4} & a_{-3} & a_{-2} & a_{-1} & a_0 \end{bmatrix}.$$

*Esercizio 3.* Utilizzando la DFT scrivere una funzione che calcoli il prodotto tra una matrice di Toeplitz *triangolare inferiore* e un vettore:

$$\begin{bmatrix} a_0 & 0 & 0 & 0 & 0 \\ a_{-1} & a_0 & 0 & 0 & 0 \\ a_{-2} & a_{-1} & a_0 & 0 & 0 \\ a_{-3} & a_{-2} & a_{-1} & a_0 & 0 \\ a_{-4} & a_{-3} & a_{-2} & a_{-1} & a_0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} a_0 x_0 \\ a_{-1} x_0 + a_0 x_1 \\ a_{-2} x_0 + a_{-1} x_1 + a_0 x_2 \\ a_{-3} x_0 + a_{-2} x_1 + a_{-1} x_2 + a_0 x_3 \\ a_{-4} x_0 + a_{-3} x_1 + a_{-2} x_2 + a_{-1} x_3 + a_0 x_4 \end{bmatrix}$$

Potete rappresentare la matrice di Toeplitz passando alla funzione la sua prima colonna  $c$ : `function y=lower_toeplitz_vector(c,x)`. Hint: vedere gli elementi del vettore da calcolare come (parte dei) coefficienti di un prodotto tra polinomi.

*Esercizio 4.* Scrivete una funzione che calcoli il prodotto tra una matrice di Toeplitz *triangolare superiore* e un vettore. Potete rappresentare la matrice di Toeplitz con la sua prima riga.

*Esercizio 5.* Scrivete una funzione che calcoli il prodotto tra una matrice di Toeplitz generica e un vettore. Potete rappresentare la matrice di Toeplitz passando alla funzione la sua prima colonna e la sua prima riga: `function y=toeplitz_vector(c,r,x)`. Hint: usate le due funzioni appena scritte.

*Esercizio 6.* La somma di matrici di Toeplitz è una matrice di Toeplitz? Il prodotto? L'inversa? E se le matrici sono triangolari inferiori/superiori?

Una matrice di Toeplitz è detta *circolante* se  $a_{-k} = a_h$  ogniqualvolta  $h - k = n$ . Per esempio

$$\begin{bmatrix} a_0 & a_1 & a_2 & a_3 & a_4 \\ a_4 & a_0 & a_1 & a_2 & a_3 \\ a_3 & a_4 & a_0 & a_1 & a_2 \\ a_2 & a_3 & a_4 & a_0 & a_1 \\ a_1 & a_2 & a_3 & a_4 & a_0 \end{bmatrix}.$$

*Esercizio 7* (facoltativo). Scrivete una `function y=circulant_vector(r,x)` che esegua il prodotto tra una matrice circolante con prima riga  $\mathbf{r}$  e il vettore  $\mathbf{x}$  usando solo due DFT e una IDFT. Hint: Confrontare le componenti del vettore da calcolare con i coefficienti centrali del prodotto

$$(a_1 + a_2 z + a_3 z^2 + a_4 z^3 + a_0 z^4 + a_1 z^5 + a_2 z^6 + a_3 z^7 + a_4 z^8) \cdot \sum_{i=0}^4 x_i z^{4-i}.$$

*Esercizio 8.* Il prodotto tra matrici circolanti è una matrice circolante? E l'inversa? Provare a giustificare formalmente la risposta (i.e., fornire controesempi o dimostrare che questa classe di matrici è un'algebra).

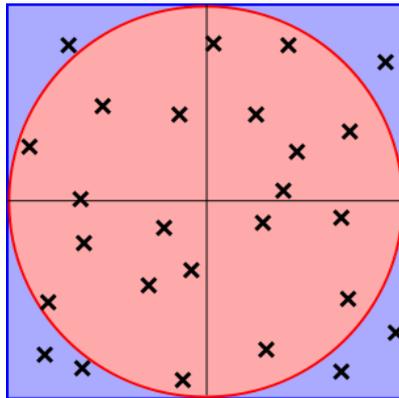
## 2 Facoltativo: calcolo di $\pi$ con il metodo Monte Carlo

Supponiamo di prendere un punto a caso nel quadrato  $[-1, 1] \times [-1, 1]$ . Qual è la probabilità che esso sia all'interno del cerchio inscritto di raggio 1? È  $\frac{\pi}{4}$ , cioè il rapporto tra l'area del cerchio e l'area del quadrato. Quindi, per la legge dei grandi numeri, se prendo  $n$  punti a caso all'interno del quadrato, il rapporto

$$\frac{\#\{\text{punti all'interno del cerchio}\}}{n}$$

tende a  $\frac{\pi}{4}$ . Questo ci suggerisce un metodo per calcolare un'approssimazione di  $\pi$  nel modo seguente:

1. Scegliere  $n$  punti a caso in  $[-1, 1] \times [-1, 1]$
2. Determinare il numero  $r$  di questi punti che stanno nel cerchio  $\{x^2 + y^2 \leq 1\}$
3. Restituire  $4\frac{r}{n}$ .



Questo metodo è noto come *Monte Carlo* perché il risultato è garantito a meno di “colpi di sfortuna” nella scelta dei numeri casuali, un po' come in un casinò...

*Esercizio 9* (facoltativo). Scrivete una funzione `function pi_appr=montecarlo(n)` che calcoli un'approssimazione di  $\pi$  in questo modo. <sup>1</sup>

<sup>1</sup>Se siete interessati a scrivere la funzione ottimizzando al massimo il numero di istruzioni eseguite, vi conviene generare tutti i punti casuali insieme con `rand(n,2)`, e poi ingegnarvi un po' con le operazioni elemento-per-elemento in modo da fare il calcolo richiesto senza cicli `for`. Può esservi utile sapere che, data una matrice `A`, l'istruzione `A<=1` restituisce una matrice che al posto  $(i, j)$  ha 1 se  $A_{ij} \leq 1$  e 0 altrimenti.

*Esercizio 10* (facoltativo). Quanto velocemente converge a  $\pi$  il valore trovato? Si calcoli `montecarlo(n)` per diversi valori di  $n$  (anche fino a  $n \approx 50000$  dovrebbe essere fattibile se scrivete il programma con attenzione) e si traccino i risultati su un grafico (funzione `plot`). Il plot più significativo si ottiene prendendo un grafico logaritmico: si calcola la funzione su  $n = 2^k$ , e poi si riportano sul grafico  $k$  sull'asse delle ascisse e  $\log |\text{montecarlo}(n) - \pi|$  sulle ordinate.

*Esercizio 11* (facoltativo). Riuscite a stimare l'ordine di convergenza del metodo dai risultati ottenuti?

*Esercizio 12* (facoltativo). Invece del numero di punti in  $\{x^2 + y^2 \leq 1\}$ , contate il numero di punti in  $\{x^2 + y^2 \leq \frac{1}{2}\}$ . A quale numero converge? Converge più o meno velocemente del metodo precedente? Sapreste spiegare il perché? Hint: Cercate su Wikipedia il Teorema limite centrale.

*Esercizio 13* (facoltativo). Calcolate in modo approssimato con lo stesso trucco l'integrale  $\int_0^1 x^2 dx$ : quanti punti scelti a caso in  $[0, 1] \times [0, 1]$  cadono nell'area sotto la parabola?