Metodi Numerici

Laboratorio 3 – Interpolazione di funzioni

a.a. 2019-20

Programma di oggi

1. `struct` in MatLab/Octave.
2. Forma di Newton del polinomio interpolatore
   - calcolo delle “differenze divise”
   - algoritmo di Neville
   - “aggiunta di un nodo”
3. esperimenti con l’interpolazione di funzioni item esercizi con il metodo dei minimi quadrati

struct in MatLab/Octave

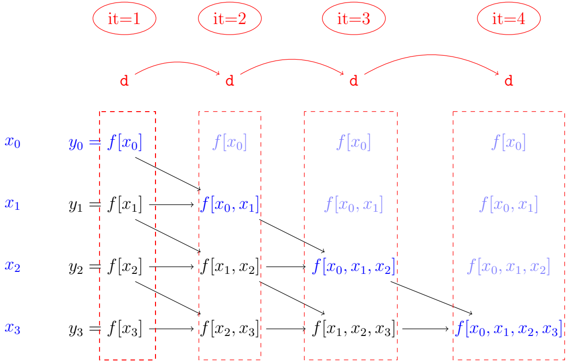
Il modo più semplice per definire una struct

```
>> S.x = [1,2,3];
>> S.y = [3,4,7];
>> S.t = "data"
S =
  scalar structure containing the fields:
    x =       1   2   3
    y =       3   4   7
    t = data
>> whos
Variables in the current scope:
   Attr Name        Size                     Bytes  Class
   ==== ====        ====                     =====  =====
        S           1x1                         52  struct
Total is 1 element using 52 bytes
>> plot(S.x, S.y);
>> title(S.t)
```

Nota: per usi più avanzati, usate il comando struct

Calcolo delle differenze divise

.m Completate la function `diffDiv.m` che, dati in input un elenco di nodi di interpolazione e di valori da interpolare, restituisce i coefficienti della forma di Newton del polinomio interpolatore (le differenze divise).
Il disegno qui sotto suggerisce come implementare il calcolo usando un solo vettore `d` della stessa lunghezza dei dati.



Debugging di `diffDiv.m`

- Calcolate a mano la tabella delle differenze divise per i dati

x	0	1	2	4
y	1	-1	3	5

  e verificate che l'output della function `diffDiv` sia corretto

- deducete che il polinomio interpolatore dei dati del punto precedente è

$$\begin{aligned}\pi_3(x) &= 1 - 2x + 3x(x-1) - x(x-1)(x-2) \\ &= -x^3 + 6x^2 - 7x + 1\end{aligned}$$

Algoritmo di Neville

$$\pi_N(\hat{x}) = f[x_0] + (\hat{x} - x_0)\Big\{ f[x_0, x1] + (\hat{x} - x_1)\Big\{ \dots$$

$$\Big\{ f[x_0, \dots, x_{N-1}] + (\hat{x} - x_{N-1}) f[x_0, \dots, x_N] \Big\} \Big\} \Big\}$$

L'algoritmo di Neville valuta il polinomio nel punto x "partendo dall'interno" nella formula precedente, ovvero da $f[x_0, \dots, x_N]$ ed eseguendo ad ogni passo la moltiplicazione per un monomio e la somma di una differenza divisa di ordine via via più basso.

Completate la function `neville.m` che, data in input una struct resitutita da `diffDiv` e un vettore di punti in cui valutare il polinomio, restituisce i valori assunti dal polinomio nei punti indicati.
Nota: la function deve operare correttamente anche se `x̂` in input è un vettore; a tal fine, usate l'operazione vettoriale `.*` e non un ciclo `for`.

Debugging di `neville.m`

- Disegnate sullo stesso grafico
  - i punti

x	0	1	2	4
y	1	-1	3	5

    con (circoletti rossi)
  - calcolate con `diffDiv` le differenze divise e valutate con `neville` il polinomio interpolatore su almeno 100 punti nell'intervallo $[0, 4]$
  - disegnate il grafico del polinomio: deve interpolare i dati!
- sovrapponete il grafico del polinomio

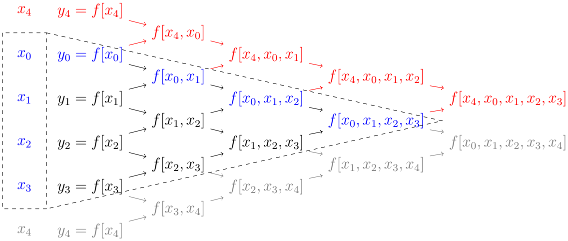
$$\pi_3(x) = -x^3 + 6x^2 - 7x + 1 \tag{1}$$

Dovrebbe coincidere con quello disegnato usando `neville`

Nota: nella stringa di formato (terzo argomento di `plot`) potete concatenare un colore (`r,g,b,m,c,y,k`), un tipo linea (`-,--,.-`) ed un simbolo (`+,o,*,.,d` etc). Non è obbligatorio indicare tutti e 3 gli elementi (si veda l'help di plot)

Aggiunta di un nodo di interpolazione

Come dimostra il disegno qui sotto, volendo aggiungere un nuovo nodo (x_4, y_4) dopo aver già calcolato le differenze divise in blu, onde non dover ricalcolare l'intera tabella (grigio), conviene mettere il nodo in cima alla tabella (rosso).



Completate la function `diffDivAggNodo.m` che prende in ingresso una struct `DD` come quelle restituite da `diffDiv` ed una nuova coppia di dati `xnew` e `ynew`. La function deve restituire una struct di differenze divise che rapprsenti un polinomio interpolante il dato (`xnew`, `ynew`) oltre ai dati già intepolati da `DD`.

Debugging di `diffDivAggNodo.m`

Test. Il polinomio interpolante i dati

x	0	1	2	4
y	1	-1	3	5

deve essere indipendente dall'ordine in cui consideriamo i nodi.

- calcolate le differenze divise di tutti i dati con `diffDiv` e disegnate il polinomio interpolatore
- calcolate le differenze divise dei primi tre dati con `diffDiv`, aggiungete il quarto con `diffDivAggNodo` e disegnate il polinomio interpolatore. Deve coincidere col precedente!
- calcolate le differenze divise degli ultimi tre dati con `diffDiv`, aggiungete il primo con `diffDivAggNodo` e disegnate il polinomio interpolatore. Deve coincidere col precedente!
- calcolate le differenze divise di due dati con `diffDiv`, aggiungetene un terzo e poi un quarto con `diffDivAggNodo`. Il polinomio Deve coincidere col precedente!

Test finali - interpolazione

- Calcolate e disegnate il polinomio interpolatore di $f(x) = \sin(x)$ su $[0, 2\pi]$ con 8 nodi equispaziati.
- Valutate la distanza fra $\sin(x)$ e il suo polinomio interpolatore su $[0, 2\pi]$ con 8 nodi equispaziati. Per approssimare il valore massimo di tale distanza, calcolatela in almeno 1000 punti e calcolatene il massimo con la funzione `max`.
- (fenomeno di Runge) Interpolate la funzione $f(x) = 1/(1 + 25 * x^2)$
  - con $3, 5, 7, 11$ nodi equispaziati nell'intervallo $[-5, 5]$
  - con $3, 5, 7, 11$ nodi nell'intervallo $[-5, 5]$ assegnati da

$$x_k = -5\cos(\pi k/N), \quad k = 0, \dots, N$$

  e sovrapponete in un grafico $f(x)$, e i polinomi interpolatori.
- (fenomeno di Gibbs) Interpolate la funzione $f(x) = \mathrm{sign}(x)$ e sovrapponete in un grafico $f(x)$, e i polinomi interpolatori.

Minimi quadrati

- È stato misurato un tratto di strada.

Assumiamo che siano state fatte 5 misurazioni e si siano ottenuti i seguenti valori

$AD = 89m,\ AC = 67m,\ BD = 53m,\ AB = 35m,\ CD = 20m.$

Si determinino la lunghezza dei segmenti $x_1 = AB, x_2 = BC$ e $x_3 = CD$.

Minimi quadrati

Scrivere uno script che, assegnati i due vettori di dati riportati nella seguente tabella,

x	y
6.5	17.769
7.0	24.001
8.0	25.961
8.5	34.336
9.0	29.036
9.1	33.417

i) trovi i polinomi approssimanti di grado 1, 2, 3, 4 ai minimi quadrati, prima usando il comando `polyfit` e, successivamente, la matrice di Vandermonde e li disegni nella stessa figura;

ii) calcoli il polinomio interpolante i dati mediante l'istruzione `polyfit` e lo disegni, in un unica figura, con i polinomi approssimanti ottenuti al punto i).