Calcolo Numerico - Laurea in Matematica, a.a. 2021-2022
Esercizi di Laboratorio del 09/03/2022

**Interpolazione polinomiale.** (Suggerimento: scrivi uno script con tutti i comandi)

## (A PAG.2 SI TROVANO INDICAZIONI SULL'USO DI ALCUNE FUNZIONI)

Considera la funzione $f(x) = x^2 \sin(x)\cos(x)$, per $x \in [a, b]$ con $a = -\pi$, $b = \pi$.

1. Definisci l'**handle** `f = @(x)(x.^2.*sin(x).*cos(x));` e fai il grafico della funzione: `fplot(f,[-pi,pi],'r--')`

2. Per $n = 6, 10, 14$, determina il polinomio interpolante $p_n(x)$ come segue:

   i) Determina e plotta i nodi equispaziati interpolanti $(x_i, f(x_i))$:

   ```
   x=linspace(-pi,pi,n+1)';
   hold on
   plot(x,f(x),'*')
   ```

   ii) Siano `y=f(x)` e `a` il vettore contenente i coeff del polinomio interpolante. Crea la funzione `a=get_polyn(x,y)` contenente le seguenti operazioni:

   * Creazione della matrice di Vandermonde di $n+1$ nodi equispaziati $x_i$ in $[a, b]$.

   * Calcolo dei coefficienti del polinomio interpolatore risolvendo il sistema

   $$V_n \mathbf{a} = \mathbf{y}, \qquad V_n = \begin{bmatrix} x_0^n & x_0^{n-1} & \dots & 1 \\ \vdots & \vdots & & \vdots \\ x_n^n & x_n^{n-1} & \dots & 1 \end{bmatrix}.$$

   **Nota:** Per ogni $n$ includi il calcolo di `cond(V_n)` e commenta.

   **Suggerimento:** Dato il vettore di nodi `x`, usa i seguenti comandi per costruire $V_n$:

   ```
   V(:,n+1) = ones(n+1,1);
   for j = n:-1:1,
     V(:,j) = (x).*V(:,j+1);
   end
   ```

   iii) Per calcolare il valore di un polinomio $p_n$ in un punto, puoi usare la funzione Matlab `yp=polyval(a,t)`, che calcola il valore del polinomio di coefficienti $a \in \mathbb{R}^{n+1}$ nei punti $t \in \mathbb{R}^r$. (**Attenzione**, la funzione Matlab `polyval` usa i coefficienti nell'ordine $a(1) = a_n, a(2) = a_{n-1}, ...$, coerente con la definizione di $V_n$ fatta sopra)

   Per ogni $n$, riporta il grafico del polinomio $p_n(x)$ sulla stessa figura di $f$ mediante i seguenti comandi:

   ```
   t=linspace(-pi,pi,100);          % valori t_i in ascissa per fare il grafico
   yp=polyval(a,t);                 % valuta p_n(t_i)
   plot(t,yp,'k');                  % fa il grafico del polinomio usando i punti (t_i, yp_i)
   ```

3. Determina l'errore $E_n(f) := \max_{x\in[a,b]} |f(x) - p_n(x)|$ per le varie scelte di $n$ e commenta;

   (suggerimento: $E_n(f) \approx$ `max(abs(f(t)-yp))`, se $t \in \mathbb{R}^d$ con le componenti di $t$ fittamente distribuite in $[a, b]$ (eventualmente usare più di $d = 100$ valori per $t$, ed in generale sempre diversi dai nodi di interpolazione, per i quali l'errore e' zero!)

4. (Facoltativo) Approssimazione mediante minimi quadrati:

   i) Modifica la funzione `get_polyn.m` in modo da dare in input il grado $m$ del polinomio richiesto e risolvere il problema dei minimi quadrati associato, se $m < n$, con `a = V\f` (es. $n = 6, 10, 14$ ed $m = n - 2$, ma prova anche altri valori di $m < n$).

   ii) Riporta sulla figura il grafico del polinomio $p_m$ risultante, come nel punto (iii) sopra.

## INDICAZIONI SULL'USO DI ALCUNE FUNZIONI

- Costrutto per funzioni in riga (in gergo, `handle`):

```
f = @(x)(x.^2.*sin(x).*cos(x));
```

  1. Il comando definisce la funzione `f` nella variabile `x`, corrispondente a $f(x) = x^2 \sin(x) \cos(x)$. La valutazione è per componente (uso di "`.*`"), per cui `x` può essere un vettore. Il comando

```
y = f(x);
```

  determina `y`, per componente, corrispondente ad ogni componente in `x`.

  Lo handle può essere chiamato come si vuole, non necessariamente con la lettera `f` (es. `mio_cos = @(x)(cos(x))`)

  2. Per permettere l'inserimento di più parametri, si può usare per esempio

```
f = @(x,a)(x.^a.*sin(x).*cos(x));
```

  In altre parole, se si scrive `@(x,a)(....)`, vuol dire che entrambe `x` e `a` sono date in input.

  3. È anche possibile usare un parametro, senza usarlo come variabile. Per esempio

```
a=3; f = @(x)(x.^a.*sin(x).*cos(x));
```

- Il comando `fplot` rappresenta una variante di `plot` per il grafico di funzioni. In input richiede la funzione (come handle), e l'intervallo per il quale fare il grafico. `fplot` in modo autonomo decide quanti punti mettere per la migliore visualizzazione.