Calcolo Numerico - Laurea in Matematica, a.a. 2021-2022
Esercizi di Laboratorio del 08/11/2021

Possibili comandi utili:
`P=diag(diag(A)); d = diag(A); P=tril(A); rho=max(abs(eig(B)));legend('testo a','testo b',...);`

Siamo interessati alla risoluzione del sistema lineare

$$A\mathbf{x} = \mathbf{f}, \quad A \in \mathbb{R}^{n\times n} \tag{1}$$

La funzione Matlab `[A,f]=poisson2d(n1);` (nel sito del corso) crea $A \in \mathbb{R}^{n\times n}$, $n = \texttt{n1}^2$ ed $\mathbf{f} \in \mathbb{R}^n$ per l'approssimazione numerica della equazione differenziale $-u_{xx} - u_{yy} = f$, con $u = u(x,y)$, e $(x,y) \in (0,1)\times(0,1)$, con condizioni al bordo di Dirichlet nulle e termine noto $f \equiv 1$.

1. Implementa il metodo di Jacobi mediante la funzione

   `[x,its,resnormvec]=Jacobi(A,f,x0,maxit,tol);`

   (In output: `x` contiene il vettore soluzione approssimato, `its` il numero di iterazioni effettuate, e `resnormvec` contiene $\|r_k\|$ per $k = 0, 1, \ldots, \texttt{its}$)). Usa la funzione per risolvere il sistema lineare (1) con $\texttt{n1} = 10$, cosicchè A avrà dimensione 100×100 ($n = 100$), dando in input: `x0=zeros(n,1); maxit=5000;tol=1e-8`. In particolare:

   a) Prendi come riferimento il seguente algoritmo, dove devi aggiungere il criterio d'arresto:

   Considera lo splitting $A = P - N$, scegli $\mathbf{x}_0$, e calcola $\mathbf{r}_0$. Poi

   Per $k = 0, 1, \ldots$

   $\mathbf{x}_{k+1} = \mathbf{x}_k + P^{-1}\mathbf{r}_k$

   $\mathbf{r}_{k+1} = \mathbf{f} - A\mathbf{x}_{k+1}$

   (Nota: **Per P diagonale, l'operazione $P^{-1}r$ dev'essere fatta elemento per elemento**)

   b) Riporta su un grafico (Figura 1) la storia della convergenza, cioè $\|\mathbf{r}_k\|/\|\mathbf{r}_0\|$, al crescere di k, usando la funzione di grafica `semilogy`.

   c) Riporta su un altro grafico (Figura 2) la soluzione approssimata ottenuta, `x`, mediante il comando `mesh(reshape(x,n1,n1))`
   Questa rappresenta la soluzione approssimata della equazione differenziale considerata.

2. Fai una cosa analoga per il metodo di Gauss-Seidel e riporta la storia della convergenza sullo stesso grafico di quello di Jacobi (usando colori e/o simboli diversi). (Per la risoluzione con P triangolare inferiore, implementa la funzione matlab che risolve un sistema con matrice triangolare inferiore)

3. Crea uno script, che per $\texttt{n1} = 50 : 50 : 300$, confronta i tempi di calcolo dei due metodi mediante le funzioni Matlab `tic-toc`. Riporta con un display i tempi di CPU come segue

```
n1   Jacobi         Gauss-Seidel
 50    ....              ....
100    ....              ....
150    ....              ....
...    ....              ....
```

4. Analisi teorica: per $\texttt{n1} = 10$ calcola il raggio spettrale $\rho(B)$, dove B è la matrice di iterazione per ognuno dei due metodi, *in questo caso calcolata esplicitamente.* Confronta la velocità asintotica di convergenza dei metodi di Jacobi e Gauss-Seidel, riportando su un grafico (Figura 3) i valori di $\rho(B)^k$ per $k = 1, 2, ...$ (usa per esempio il comando `semilogy( rhoBJ.^(1:500),'r--')`) dove `rhoBJ` corrisponde a $\rho(B_J)$.