Soluzioni I prova in itinere

12 aprile 2022

Test 1

$$x = (-1)^s\, 0.m \cdot 2^e = (-1)^0 \cdot 0.1011 \cdot 2^2 = \left(\frac{1}{2} + \frac{1}{8} + \frac{1}{16}\right) \cdot 4 = 2 + \frac{1}{2} + \frac{1}{4} = \frac{11}{4} = 2.75.$$

Test 2

La somma richiesta può essere calcolata tramite i comandi

```
n = 90;
k = 0:n;
S = 4*sum((-1).^k./(2*k+1))
```

Risulta pertanto $S = 3.1526$.

Test 3

La matrice L è triangolare inferiore, sparsa, con struttura per diagonali, e unitaria. In particolare, solo le due sottodiagonali sono non nulle (oltre alla diagonale principale). L'algoritmo `fwsub` per il sistema $L\mathbf{x} = \mathbf{b}$ si particolarizza nel seguente modo:

```
x = zeros(n,1);
x(1) = b(1);
x(2) = b(2) - L(2,1)*x(1);
for i = 3:n
    x(i) = b(i) - L(i,i-2:i-1)*x(i-2:i-1);
end
```

Il numero di flops sarà pertanto:

$$\#\text{flops} = 2 + \sum_{i=3}^{n} 4 = 2 + 4(n-2) = 4n - 6.$$

Per $n = 2000$ si hanno dunque 7994 flops.

Test 4

Ricordiamo la stima di sensitivity:

$$\frac{||x - \tilde{x}||}{||x||} \leq K(A) \frac{||b - A\tilde{x}||}{||b||}, \tag{1}$$

dove x è la soluzione esatta, $\tilde{x}$ è la soluzione approssimata ottenuta con un dato algoritmo e $K(A)$ è il condizionamento di A. Nel caso in oggetto, l'algoritmo è quello implementato con il comando \ e la norma è la norma 2. In Matlab, abbiamo:

```
n = 7;
A = hilb(n);
b = 5*ones(n,1);
x = A\b;
err_rel_estim = cond(A)*norm(b-A*x)/norm(b)
```

Si ha pertanto la stima $1.7901e-04$. Il risultato potrebbe cambiare un po' (ma non troppo) a seconda del calcolatore e della versione Matlab utilizzata.

Test 5

L'algoritmo che risolve il problema posto è il seguente.

```
x0 = [0;1];
y0 = x0/norm(x0);
x1 = A*y0;
y1 = x1/norm(x1);
lambda1 = y1'*A*y1;
```

Calcolando, "a mano", le varie quantità, otteniamo:

$$y0 = [0;1]; \quad x1 = [3;1]; \quad y1 = [3;1]/\sqrt{10}; \quad \lambda_1 = (3\gamma + 37)/10 = 0.3\gamma + 3.7.$$

Test 6

La matrice A risulta simmetrica e definita positiva, come si può evincere anche dall'espressione degli autovalori, che sono tutti reali e positivi. La condizione richiesta impone di cercare $s \in \mathbb{R}$ tale che:

$$|s - \lambda_{47}| < |s - \lambda_i|, \quad \forall i \neq 47, s \neq \lambda_{47}.$$

Trattandosi di autovalori reali e positivi e, in base alla formula fornita, ordinati in ordine decrescente, questo equivale a richiedere che

$$\frac{\lambda_{48}+\lambda_{47}}{2} < s < \frac{\lambda_{47}+\lambda_{46}}{2}, \quad s \neq \lambda_{47},$$

cioè $2.1863 < s < 2.2482$ con $s \neq 2.2173$.

Test 7

Il valore $\alpha = 0$ è uno zero semplice di f, essendo $f(0) = 0$ e $f'(0) = 3 \neq 0$. Il metodo di Newton converge pertanto quadraticamente, pur di scegliere $x^{(0)}$ sufficientemente vicino ad α (ipotesi che è garantita dal testo del problema). Vale pertanto la seguente stima dell'errore (esatta asintoticamente):

$$|e^{(k+1)}| \simeq \frac{|f''(\alpha)|}{2|f'(\alpha)|}\,(e^{(k)})^2.$$

Calcolando $f''(0) = 9$ e sostituendo, si ottiene pertanto:

$$|e^{(k+1)}| \simeq \frac{9}{6}(e^{(k)})^2 = 1.5\,(10^{-2})^2 = 1.5 \cdot 10^{-4}.$$

Allo stesso risultato si può pervenire, con qualche calcolo in più, usando il teorema di convergenza di ordine p per i metodi di punto fisso, che fornisce

$$|e^{(k+1)}| \simeq \frac{\Phi''(\alpha)}{2}\,(e^{(k)})^2,$$

applicato alla funzione di iterazione di Newton,

$$\Phi(x) = \Phi_N(x) = x - \frac{f(x)}{f'(x)}.$$

Test 8

Il metodo di Steffensen richiesto si può implementare nel seguente modo, osservando che è un metodo di punto fisso (anche se non è necessario accorgersene):

```
f = @(x) exp(-4*x) - 2*x;
PhiS = @(x) x - f(x)^2 / (f(x + f(x)) - f(x));
x = 1;
tol = 1e-2;
err = 1 + tol;
```

```
k = 0;
while err > tol
    x = PhiS(x);
    err = abs(f(x));
    k = k + 1;
end
k, x
```

Si ha pertanto $N = 9$ e $x^{(N)} = 0.2135$.

Test 9

Un possibile script che permette di ottenere la soluzione è il seguente:

```
phi = @(x) x - 9/2*log(x/3);
x = 2;
for i = 1:4, x = phi(x); end, x
```

Si ottiene $x^{(4)} = 2.9291$.

Test 10

Si deve applicare il teorema di esistenza e unicità, e convergenza globale. Lo script seguente produce il grafico riportato in Figura 1

```
 phi = @(x) x - 140/11*(exp(x/7-1) - 1);
dphi = @(x) 1 -  20/11* exp(x/7-1);
x = linspace(4,8,1000);
plot(x,x,x,phi(x),x,abs(dphi(x)),x,ones(size(x)))
grid
```

Cominciamo con il verificare su quale intervallo del tipo $[a, b]$ si ha $\phi(x) \in [a, b]$ per ogni $x \in [a, b]$ con $a \geq 4$ e $b \leq 8$. Dal grafico di ϕ, si evince che la condizione è verificata per

$$a \geq 5.4013, \tag{2}$$

dove il valore numerico può essere ottenuto zoomando nell'intorno del valore corrispondente a $\phi = 8$.

In seguito, verifichiamo su quale intervallo si ha $|\phi'| < 1$. Il grafico di ϕ' mostra che la condizione è soddisfatta per $x < 7.6672$, che può essere determinato graficamente, zoomando, o risolvendo l'equazione

$$\frac{20}{11} e^{x/7-1} - 1 = 1,$$

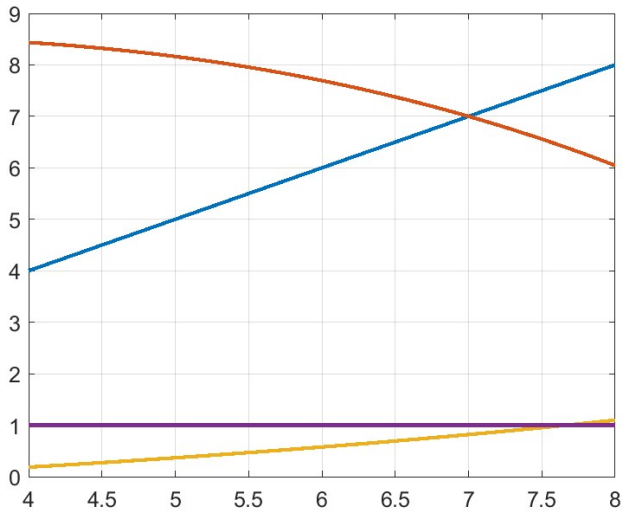

Figura 1: Grafico di ϕ (rosso), ϕ' (giallo), della bisettrice e della funzione unitaria.

dopo avere osservato che $\phi' < 0$ sull'intervallo $[4, 8]$, che fornisce $x = 7(1 + \log \frac{11}{10}) \simeq 7.6672$. Quindi si ha la condizione

$$b < 7.6672. \tag{3}$$

In conclusione, la soluzione al test è fornita dalla coppia di condizioni (2) e (3).

Esercizio - Punto 1

Si può verificare che la matrice A è simmetrica e definita positiva, ma non a dominanza diagonale stretta (né per righe, né per colonne). Si può pertanto già concludere che il metodo di Gauss-Seidel è convergente. Nulla si può dire ancora sul metodo di Jacobi, essendo la condizione di dominanza diagonale stretta solo sufficiente. Per dirimere la questione per il metodo di Jacobi, applichiamo la condizione necessaria e sufficiente, $\rho(B_J) < 1$, e calcoliamo anche $\rho(B_{GS})$ per il metodo di Gauss-Seidel, al fine di caratterizzarne la velocità di convergenza. Lo script seguente

```
n = 100;
A = 20*eye(n) + diag(ones(n-2,1),-2) ...
    - 11*diag(ones(n-1,1),-1) - 11*diag(ones(n-1,1),1) ...
    + diag(ones(n-2,1),2);
D = diag(diag(A));
T = tril(A);
```

```
BJ  = eye(n) - D\A;
BGS = eye(n) - T\A;
rhoJ  = max(abs(eig(BJ )))
rhoGS = max(abs(eig(BGS)))
```

fornisce $\rho(B_J) = 1.1993$ e $\rho(B_{GS}) = 0.9993$. Si evince pertanto che il metodo di Jacobi non converge, mentre viene confermata la convergenza del metodo di Gauss-Seidel, ma con una bassissima velocità di convergenza, essendo $\rho(B_{GS})$ molto prossimo a 1.
Si faccia attenzione che, non essendo la matrice A tridiagonale, ma pentadiagonale, non è possibile stabilire alcun collegamento fra la convergenza di un metodo e quella dell'altro.

Esercizio - Punto 2

Lo script seguente

```
b = 5*ones(n,1);
x0 = b;
toll = 1e-2;
nmax = 1e4;
[x,k] = gs(A,b,x0,toll,nmax);
k, x(1), norm(b-A*x)/norm(b)
```

permette di trovare i valori richiesti: $N = 6607$, $x_1^{(N)} = 31.3231$ e $r_{norm}^{(N)} = 0.0100$.

Esercizio - Punto 3

Occorre applicare una condizione simile alla (1), in particolare

$$e_{rel}^{(N)} = \frac{||x - x^{(N)}||}{||x||} \leq K(A) \frac{||b - Ax^{(N)}||}{||b||}.$$

Calcolando il termine di destra della stima tramite lo script

```
K = cond(A)
K*norm(b-A*x)/norm(b)
```

dove x è il vettore ottenuto al Punto 2, si ottiene $e_{rel}^{(N)} \leq 64.5901$. Dato che quest'ultimo valore è molto maggiore della tolleranza imposta, 10^{-2}, se ne deduce che il criterio basato sul residuo normalizzato non è affidabile. Questo è causato dal mal condizionamento della matrice, che risulta $K(A) = 6.4617e + 03$.

Esercizio - Punto 4

Affinché il metodo proposto sia fortemente consistente, è necessario determinare la matrice di iterazione, B_ω e il termine noto $\mathbf{g}_\omega$, in modo tale da soddisfare la condizione di splitting, $A = P_\omega - (P_\omega - A)$. Questo fornisce:

$$B_\omega = I - {P_\omega}^{-1}A = I - \omega T^{-1}A, \quad \mathbf{g}_\omega = {P_\omega}^{-1}\mathbf{b} = \omega T^{-1}\mathbf{b}.$$

Per determinare il valore di ω per cui si ha la convergenza più rapida conviene calcolare il raggio spettrale della matrice di iterazione (da cui dipende la velocità asintotica di convergenza) e verificare per quale valore è minimo (e minore di 1). Lo script seguente

```
Omega = [1.45, 1.55, 1.65, 1.75, 1.85];
T = tril(A);
nomega = length(Omega);
rho = zeros(1,nomega);
for i = 1:nomega
    omega = Omega(i);
    B = eye(n) - omega*(T\A);
    rho(i) = max(abs(eig(B)));
end
format long e
rho
```

permette di calcolare i diversi valori. Per quanto tutti i valori siano prossimi ad 1, il minimo si ha in corrispondenza di $\omega = 1.65$ e vale $9.988769880117021e-01$ (è opportuno usare il `format long e` per discriminare questo valore da quello corrispondente ad $\omega = 1.55$). Addirittura, per gli ultimi due valori di ω, non si ha convergenza.

Esercizio - Punto 5

Per determinare quale sia il migliore precondizionatore, fra $P_1 = I$ (non precondizionato) e $P_2 = \text{tridiag}(-1, 2-1)$, è necessario cacolare il condizionamento spettrale delle matrici precondizionate, $P_i^{-1}A$, $i = 1, 2$, per stabilire per quale delle due è inferiore. Lo script seguente

```
P2 = 2*eye(n) - diag(ones(n-1,1),-1) - diag(ones(n-1,1),1);
e1 = eig(A);
e2 = eig(P2\A);
Ksp1 = max(abs(e1))/min(abs(e1))
Ksp2 = max(abs(e2))/min(abs(e2))
```

fornisce $K_{sp}(A) = 6.4617e+03 > K_{sp}(P_2^{-1}A) = 1.5704$, ed indica che il precondizinatore P_2 è il più efficace, garantendo che il metodo così ottenuto sia più rapido rispetto al metodo non precondizionato. Si faccia attenzione che, affinché il confronto possa essere eseguito sulla base della stima di convergenza, è necessario che anche le matrici di precondizionamento P_i (oltre ad A) devono essere simmetriche e definite positive. Tale proprietà è garantita, come si può facilmente verificare. Sotto tale ipotesi, il comando `abs` nello script può anche essere omesso.
Il fattore di abbattimento richiesto si può calcolare tramite la formula

$$\left(\frac{K_{sp}(P_2^{-1}A) - 1}{K_{sp}(P_2^{-1}A) + 1}\right)^{10}$$

e fornisce il valore $2.8976e-07$. Tale valore, molto piccolo, indica che il metodo è molto rapido (come conseguenza del fatto che $K_{sp}(P_2^{-1}A)) \simeq 1$).

Esercizio - Punto 6

Fatta la scelta $P = P_2$ in base al punto precedente, il fattore di abbattimento richiesto si può calcolare tramite lo script

```
K = sqrt(Ksp2);
c = (K-1)/(K+1);
k = 10;
2*c^k/(1+c^(2*k))
```

che fornisce il valore $6.4159e-10$. Il valore così ottenuto, di circa tre ordini di grandezza inferiore a quello ottenuto al Punto 5, indica che il metodo del gradiente coniugato precondizionato con P_2 converge ad un tasso maggiore di quello del gradiente precondizionato con la stessa matrice (in linea con quanto ci si attende dalla teoria).

Esercizio - Punto 7

La function richiesta può essere implementata tramite il seguente script

```
x0 = ones(n,1);
ymax = x0/norm(x0);
ymin = ymax;
N = 100;
K = zeros(1,N+1);
K(1) = 1;
for k = 1:N
```

```
    xmax = A*ymax;
    ymax = xmax/norm(xmax);
    xmin = A\ymin;
    ymin = xmin/norm(xmin);
    K(k+1) = (ymax'*A*ymax)/(ymin'*A*ymin);
end
K([2,3,101])
```

I valori richiesti per le approssimazioni del numero di condizionamento spettrale di A sono: $K^{(1)} = 3.2157 \cdot 10^3$, $K^{(2)} = 4.5359 \cdot 10^3$ e $K^{(100)} = 6.4156 \cdot 10^3$.