

CALCOLO NUMERICO  
Corso di Laurea in Ingegneria Biomedica  
A.A. 2018/2019 – Correzione Appello 13/09/2019

---

NOME

COGNOME

MATRICOLA

---

**Esercizio 1**

1. Per separazione grafica si ha  $f(x) = 0$  se e solo se  $\sqrt{x} = 1 - x^2$ .
2. Si ottiene  $g'(x) = -\frac{1}{4\sqrt{1-\sqrt{x}}\sqrt{x}}$  da cui  $|g'(\alpha)| = \frac{1}{4\alpha\sqrt{\alpha}}$ . Da  $\alpha > 1/2$  segue  $4\alpha\sqrt{\alpha} > 1$  e quindi  $|g'(\alpha)| < 1$ . La convergenza locale segue dal corollario del teorema del punto fisso.
3. Si ha  $f \in C^2[1/2, +\infty]$ ,  $f'(x) = 2x+1/(2\sqrt{x}) > 0$  per ogni  $x > 0$  e  $f''(x) = 2-1/(4x\sqrt{x}) > 0$  per ogni  $x \geq 1/2$ . La convergenza per ogni punto iniziale  $x_0 \geq 1/2$  segue dal teorema di convergenza su intervalli.

4. `function [x,k] = es1_13092019_1(x0)`

```
g=@(x)sqrt(1-sqrt(x));
```

```
err=inf;
```

```
k=0;
```

```
while (err>=1.0e-12)
```

```
    x=g(x0);
```

```
    err=abs(x-x0);
```

```
    k=k+1;
```

```
    x0=x;
```

```
end
```

```
end
```

```
function [x,k] = es1_13092019_2(x0)
```

```
f=@(x)x^2 +sqrt(x)-1;
```

```
f1=@(x)2*x + 1/(2*sqrt(x));
```

```
err=inf;
```

```
k=0;
```

```
while (err>=1.0e-12)
```

```
    x=x0-f(x0)/f1(x0);
```

```
    err=abs(x-x0);
```

```
    k=k+1;
```

```
    x0=x;
```

```
end
```

```
end
```

Per  $x_0 = 0.6$  la prima funzione restituisce  $(5.248885986567915e-01, 62)$  mentre la seconda restituisce  $(5.248885986564048e-01, 5)$ .

**Esercizio 2**

1. La somma dei moduli degli elementi fuori dalla diagonale è massima nella(e) riga centrale.  
Per ogni  $k \geq 1$  ha  $2 \sum_{i=1}^k (1/4)^i < 2 \sum_{i=1}^{\infty} (1/4)^i = 2 \frac{1}{4} \frac{4}{3} = 2/3 < 1$ .

```
2. function [y] = mat_prod(x)
n=length(x);
y=zeros(n);
y(n-1)=x(n)/4;
for k=n-2:-1:1
    y(k)=(x(k+1) +y(k+1))/4;
end
end
```

```
3. function [xnew,it] = es2_13092019(b)
n=length(b);
it=0;
xold=zeros(n,1);
xnew=zeros(n,1);
yu=zeros(n,1);
yl=zeros(n,1);
err=inf;
while(err>1.0e-12)
yu(n-1)=xold(n)/4;
for k=n-2:-1:1
    yu(k)=(xold(k+1) +yu(k+1))/4;
end
yl(2)=xold(1)/4;
for k=3:n
    yl(k)=(xold(k-1)+yl(k-1))/4;
end
xnew=b-yu-yl;
err=norm(xnew-xold, inf);
it=it+1;
xold=xnew;
end
end
```

Per  $n = 64$  si ottiene  $k = 70$ . Per  $n = 1024$  si ottiene  $k = 70$ .