

ΑΠΑΝΤΗΣΕΙΣ ΕΡΓΑΣΤΗΡΙΟΥ 1

3. Μεταβλητές εισόδου τα a, tol , εξόδου $x, iter$.
4. $xdiff$ είναι μεταβλητή που υλοποιεί το κριτήριο διακοπής της επαναληπτικής διαδικασίας. σε κάθε βήμα υπολογίζει την απόλυτη σχετική μεταβολή της τιμής που υπολογίστηκε σε σχέση με την προσέγγιση της προηγούμενης επανάληψης. Όταν αυτή γίνει μικρότερη του TOL τότε η επαναληπτική διαδικασία διακόπτεται.

5. Στο Command Window εκτελούμε:

```
>> help error
```

```
>> help break
```

7. Η εντολή θα σας εμφανίσει τα σχόλια που υπάρχουν στη `mysqrt`.

8. Στο Command Window εκτελούμε:

```
>> root1=mysqrt(65,1e-4)
```

```
>> root2=mysqrt(65)
```

Τα αποτελέσματα καταχωρούνται στις μεταβλητές `root1, root2`. Στη δεύτερη με την απουσία δεύτερης παραμέτρου στην κλήση η `nargin` είναι 1 και η `tol` καθορίζεται από την εντολή `if nargin < 2, tol = eps`. Το `eps` είναι της τάξης του 10^{-16} .

Αν θέλουμε να δούμε και τον αριθμό επαναλήψεων η κλήση θα είναι η ακόλουθη:

```
>> [root1,iter1]=mysqrt(65,1e-4)
```

```
>> [root2,iter2]=mysqrt(65)
```

9. Η συνάρτηση θα είναι τώρα (οι αλλαγές υπογραμμίζονται):

```
function [x,xlist,iter] = mysqrt1(a,tol)
```

```
% Square root of a scalar, a is assumed to be >= 0.
```

```
% TOL is a convergence tolerance (default EPS).
```

```
% returns also the number of iterations ITER for convergence.
```

```
if nargin < 2, tol = eps; end
```

```
x = a;
```

```
iter = 0;
```

```
xdiff = inf;
```

```
xlist=x;
```

```
while xdiff > tol
```

```
    iter = iter + 1;
```

```
    xold = x;
```

```
    x = (x + a/x)/2;
```

```
    xdiff = abs(x-xold)/abs(x);
```

```
    if iter > 50
```

```
        error('Not converged after 50 iterations.')
```

```
    end
```

```
    xlist=[xlist;x];
```

```
end
```

10. Στο Command Window εκτελούμε:

```
>> [root1,sec1,iter1]=mysqrt(13,1e-4)
```

```
>> [root2,sec2,iter2]=mysqrt(13,1e-8)
```

12. Η εντολή στο Mathematica είναι:

```
N[Sqrt[2], 18]
```

Το αποτέλεσμα

1.41421356237309505

13. Το script γίνεται:

```
clear all;
format long e;
clf;
toler=input('Give tolerance :');
xtrue=1.41421356237309505;
[x, xs, it]=mysqrt1(2, toler);
xerr=abs(xs-xtrue);
xabserr=abs(xs-xtrue)./abs(xtrue);
subplot(2,2,1);plot(xerr);
subplot(2,2,2);plot(xabserr);
acc=-log10(xerr);
pre=-log10(xabserr);
subplot(2,2,3);plot(acc);
subplot(2,2,4);plot(pre);
```

Η εκτέλεση του από το Command Window

>> **sqrscr1**

Δείτε το παράθυρο των γραφημάτων.

14. Το ./ υλοποιεί τη διαίρεση στοιχείο προς στοιχείο δύο διανυσμάτων.

15. Το script είναι:

```
clear all;
format long e;
clf;
xtrue=1.41421356237309505;
[x1, xs1, it1]=mysqrt1(2, 1e-3);
[x2, xs2, it2]=mysqrt1(2);
xerr1=abs(xs1-xtrue);
xabserr1=abs(xs1-xtrue)./abs(xtrue);
acc1=-log10(xerr1);
pre1=-log10(xabserr1);
xerr2=abs(xs2-xtrue);
xabserr2=abs(xs2-xtrue)./abs(xtrue);
acc2=-log10(xerr2);
pre2=-log10(xabserr2);
subplot(2,2,1);plot(acc1, '*');
subplot(2,2,2);plot(pre1, 'x');
subplot(2,2,3);plot(acc2, '+');
subplot(2,2,4);plot(pre2, '.');
```

Η εκτέλεση του από το Command Window

>> **sqrscr2**

Δείτε το παράθυρο των γραφημάτων.