Κεφάλαιο 1

Οι πραγματικοί και οι μιγαδικοί αριθμοί

1. 7 Τα σύμβολα $\sum$ και $\prod$

1.7.1 Παράδειγμα Αθροίσματα και γινόμενα στο Matlab.

Υπολογίστε τις ακόλουθες ποσότητες στο MATLAB:

a) $\sum_{k=1}^{100} k$

b) $\prod_{k=1}^{100} k$

c) $\sum_{k=1}^{100} k^2$

d) $\prod_{k=1}^{100} k^2$

Αθροίσματα και γινόμενα αριθμητικών ποσοτήτων μπορούν να υπολογιστούν στο MATLAB αθροίζοντας ή πολλαπλασιάζοντας αντίστοιχα τα στοιχεία διανυσμάτων που περιέχουν τις ποσότητες. Οι συναρτήσεις **sum( ) prod( )** μπορούν να χρησιμοποιηθούν για να υπολογίσουμε τις ποσότητες a-d.

```
>> clear all
>> x=1:100;
>> sum(x)

ans =

       5050

>> prod(x)

ans =

  9.3326e+157

>> sum(x.^2)

ans =

     338350

>> prod(x.^2)

ans =
```

```
  Inf
```

Παρατηρούμε ότι η τελευταία ποσότητα είναι τελικά τόσο μεγάλη που ξεπερνά τα όρια των αριθμών που χειρίζεται το MATLAB και θεωρείται άπειρη. Εναλλακτικά μπορούμε να χρησιμοποιήσουμε τις συμβολικές δυνατότητες του περιβάλλοντος. Η εντολή **symsum( )** υπολογίζει συμβολικά αθροίσματα και η **subs( )** αντικαθιστά μία συμβολική ποσότητα με μία τιμή.

```
>> syms k n
>> symsum(k^2,1,100)

ans =

338350

>> symsum(k^2,1,n)

ans =

1/3*(n+1)^3-1/2*(n+1)^2+1/6*n+1/6

>> pretty(ans)

                        3              2
                1/3 (n + 1)  - 1/2 (n + 1)  + 1/6 n +
1/6

>> subs(ans,n,100)

ans =

     338350
```

Κεφάλαιο 2

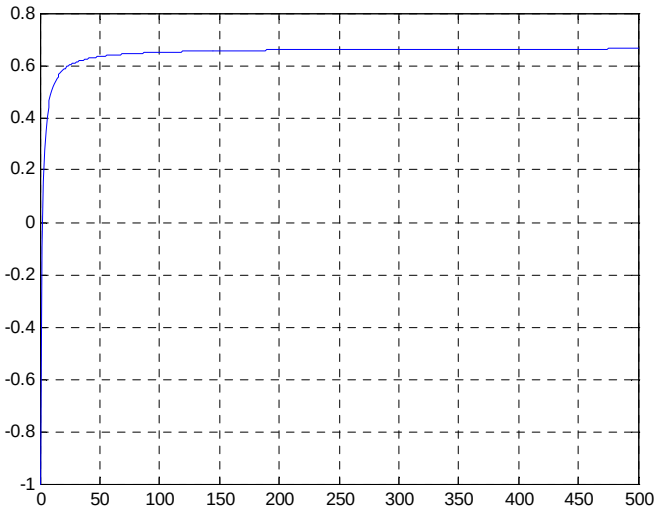
Ακολουθίες πραγματικών αριθμών

2.2 Παράδειγμα Συμπεριφορά όρων ακολουθίας στο Matlab.

Θα ορίσουμε, με τον πλέον αποδοτικό τρόπο για το MATLAB, τους 500 πρώτους όρους της ακολουθίας με τύπο $a_n = \frac{2n^2 - 5n + 1}{3n^2 - 1}$ και θα κάνουμε το γράφημά τους για να δούμε τη συμπεριφορά τους. Για να ορίσουμε τον τύπο της ακολουθίας θα χρησιμοποιήσουμε διανυσματικές πράξεις. Οι εντολές είναι οι ακόλουθες:

```
>> n=1:500;
>> an=(2*n.^2-5*n+1)./(3*n.^2-1);
>> plot(an)
>> grid on
```

και το αποτέλεσμα



μας προδιαθέτει για σύγκλιση της ακολουθίας στο 2/3.

Κεφάλαιο 3

Σειρές πραγματικών αριθμών.

3.3 Παράδειγμα Άπειρα Αθροίσματα στο Matlab.
Υπολογίστε τις σειρές

a) $\sum_{k=1}^{n}(\frac{1}{k}-\frac{1}{k+1})$
b) $\sum_{n=1}^{\infty}(\frac{1}{n}-\frac{1}{n+1})$
c) $\sum_{n=1}^{\infty}\frac{1}{k^2}$

Πάλι θα χρησιμοποιήσουμε τη συνάρτηση συμβολικών υπολογισμών **symsum( )** και οι αντίστοιχοι υπολογισμοί είναι οι ακόλουθοι.

```
>> clear all
>> syms k n
>> symsum(1/k-1/(k+1),1,n)

ans =

-1/(n+1)+1

>> symsum(1/n-1/(n+1),1,Inf)

ans =

1

>> symsum(1/n^2,1,Inf)

ans =

1/6*pi^2
```

Κεφάλαιο 4

Πραγματικές συναρτήσεις μίας Μεταβλητής

4.1.1 Παράδειγμα Ορισμός συνάρτησης στο MATLAB.

Ορίστε τις συναρτήσεις

$$f(x) = \cos(x^2 + 1)$$

$$g(x) = \frac{1}{x^2 - 1}$$

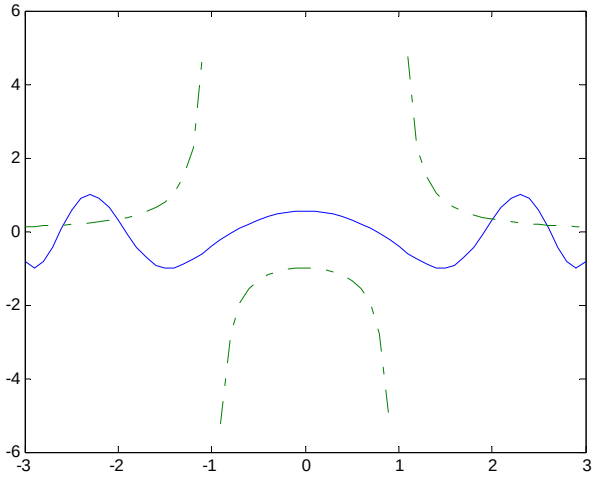
Κάντε τη γραφική τους παράσταση στο διάστημα [-π,π] . Να βρεθούν οι αντίστροφες συναρτήσεις f^{-1}και g^{-1}, οι σύνθετες συναρτήσεις $f \circ g$ και $g \circ f$ και να γίνουν τα γραφήματά τους.

Πρώτα από όλα θα πρέπει να κάνουμε μία διευκρίνιση. Όταν λέμε συνάρτηση σε ένα περιβάλλον, όπως το MATLAB, εννοούμε έναν τύπο ο οποίος δεχόμενος τιμές, επιστρέφει τιμές. Δηλαδή, ορίζουμε μία μαθηματική σχέση και όχι μία αυστηρά μαθηματικά ορισμένη συνάρτηση. Αν ο τύπος που έχουμε ορίσει δεν ορίζεται σε κάποια τιμή που καλούμαστε να τον υπολογίσουμε τότε, το MATLAB θα επιστρέψει μία από τις απροσδιόριστες μορφές του (**Inf** ή **Nan**) και όταν ζητήσουμε τη γραφική αναπαράστασή της ποσότητας αυτής δεν θα μας εμφανίσει κάποιο σημείο.

Στην πιο απλή μορφή στο MATLAB μπορούμε να ορίσουμε ένα διάνυσμα που να περιέχει μία πυκνή διαμέριση του πεδίου στο οποίο θέλουμε να παρουσιάσουμε τη συνάρτηση και να εφαρμόσουμε σε αυτό τον τύπο της συνάρτησης. Με αυτόν τον τρόπο θα υπολογίσουμε ένα διάνυσμα που να περιέχει τις τιμές της συνάρτησης για αυτά. Στον ορισμό της έκφρασης της συνάρτησης μπορούμε να χρησιμοποιούμε τις μαθηματικές συναρτήσεις του MATLAB και θα πρέπει να χρησιμοποιούμε τους **διανυσματικούς τελεστές (vectorized) .* ,./, .^** που μας δίνουν τη δυνατότητα να εφαρμόσουμε τον μαθηματικό τύπο σε διανύσματα (στοιχείο προς στοιχείο).

Με αυτόν τον τρόπο μπορούμε να εμφανίσουμε με τη χρήση της **plot( )** τα γραφήματα μίας ή και περισσότερων συναρτήσεων μαζί. Ως ορίσματα της plot( ) βάζουμε τα ζεύγη των διανυσμάτων (πεδίο ορισμού, πεδίο τιμών) και το είδος της γραμμής (π.χ. '-.' για διακεκομμένη γραμμή).

```
>> clear all
>> clf
>> x=-3:0.1:3;
>> f=cos(x.^2+1);
>> g=1./(x.^2-1);
>> plot(x,f,'-',x,g,'-.')
```



Εναλλακτικά, μπορούμε να ορίσουμε τη συνάρτηση με τη χρήση της **inline( )**. Η συνάρτηση θα πρέπει να δοθεί με μορφή κειμένου (μέσα σε εισαγωγικά). Η συνάρτηση **vectorize( )** μπορεί να μετατρέψει την μαθηματική έκφραση της συνάρτησης σε διανυσματική μορφή, ώστε να μπορεί να εφαρμοστεί σε διανύσματα.

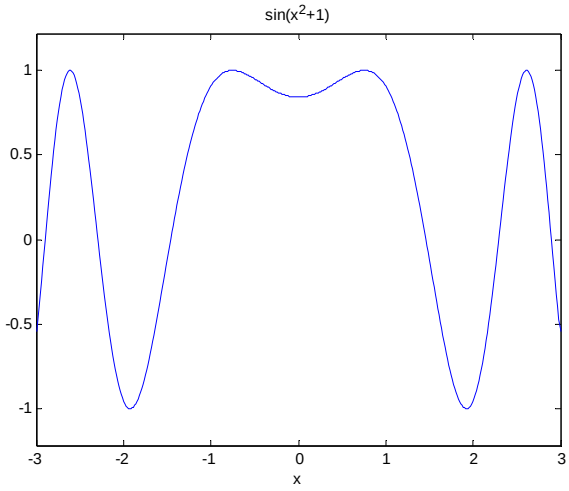
```
>> clear all
>> clf
>> f=inline(vectorize('sin(x^2+1)'))

f =

     Inline function:
     f(x) = sin(x.^2+1)

>> g=inline(vectorize('1/(x^2-1)'))

g =

     Inline function:
     g(x) = 1./(x.^2-1)

>> f([-3 0 3])

ans =

   -0.5440    0.8415   -0.5440

>> g(1)

ans =

   Inf
```

Οι συναρτήσεις **argnames( )** και **formula( )** μπορεί να μα δώσει πληροφορίες για τις συναρτήσεις που ορίζουμε.

```
>> argnames(g)

ans =

    'x'

>> formula(g)

ans =

1./(x.^2-1)
```

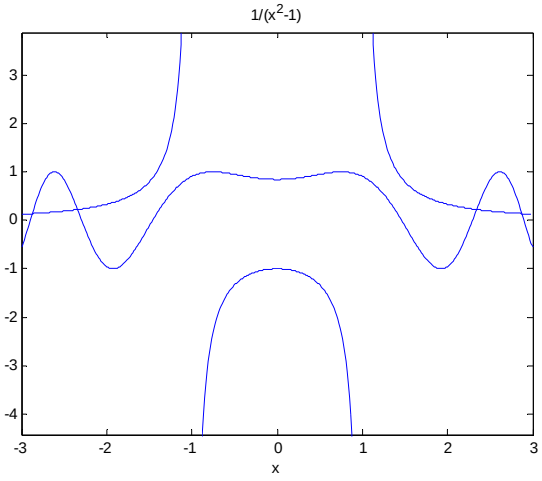
Η **ezplot( )** με παραμέτρους το όνομα και το διάστημα σχεδίασης, εμφανίζει το γράφημα μίας συνάρτησης.

```
>> ezplot(f,[-3 3])
```



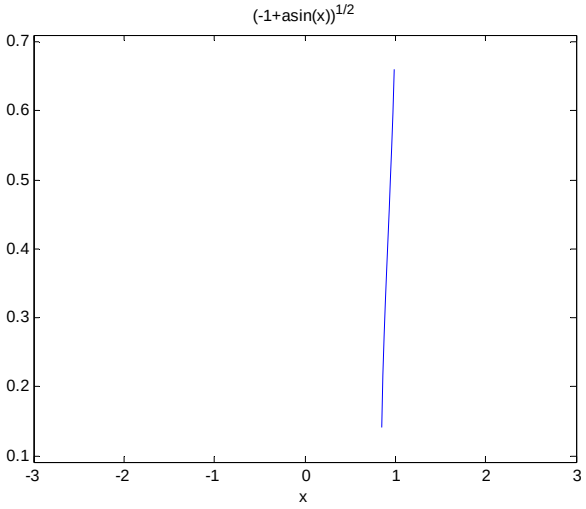
Όμως για ταυτόχρονη αναπαράσταση παραπάνω από μία συναρτήσεις θα πρέπει να κάνουμε εφαρμογή της **hold**.

```
>> hold
Current plot held
>> ezplot(g,[-3 3])
```



Η χρήση των συμβολικών δυνατοτήτων του MATLAB μας επιτρέπει να βρούμε την αντίστροφη συνάρτησης με την **finverse( )** ή τον τύπο της σύνθετης συνάρτησης με τη χρήση της **compose( )**. Αφού οριστεί η μεταβλητή x ως συμβολική στην **inline( )** ο τύπος της συνάρτησης δε χρειάζεται εισαγωγικά. Και σε αυτήν την περίπτωση μπορούμε να χρησιμοποιήσουμε την **ezplot( )**.

```
>> clf
>> syms x
>> f=inline(vectorize(sin(x^2+1)))

f =

     Inline function:
     f(x) = sin(x.^2+1)

>> g=inline(vectorize(1/(x^2-1)))

g =

     Inline function:
     g(x) = 1./(x.^2-1)

>> finverse(f(x))

ans =

(-1+asin(x))^(1/2)

>> pretty(ans)

                                  1/2
                    (-1 + asin(x))

>> ezplot(finverse(f(x)),[-3,3])
```




```
>> finverse(g(x))
```

ans =

1/x*(x*(x+1))^(1/2)

>> pretty(ans)

```
                  1/2
(x (x + 1))
--------------
      x
```

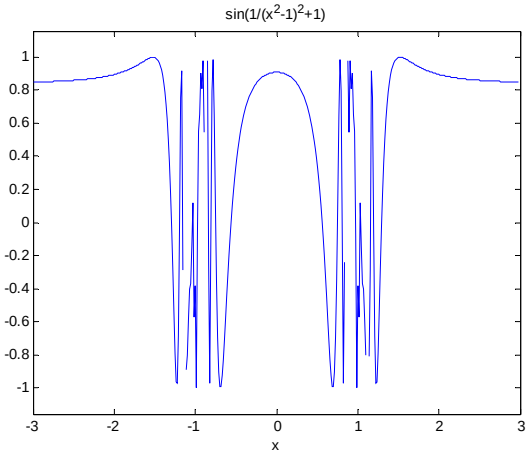
>> z=compose(f(x),g(x))

z =
sin(1/(x^2-1)^2+1)

>> pretty(z)

```
          1
sin(--------- + 1)
      2     2
   (x  - 1)
```

>> ezplot(z,[-3,3])



>> h=compose(g(x),f(x))

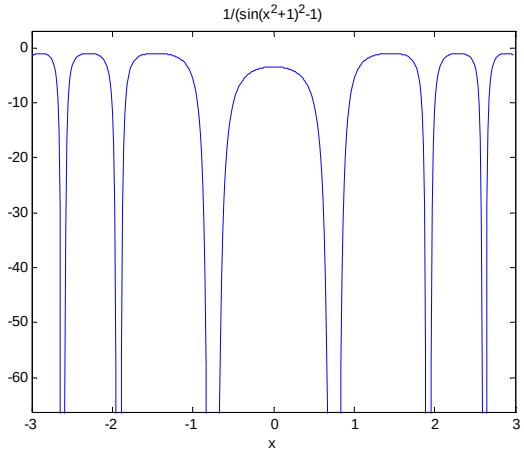
h =

1/(sin(x^2+1)^2-1)

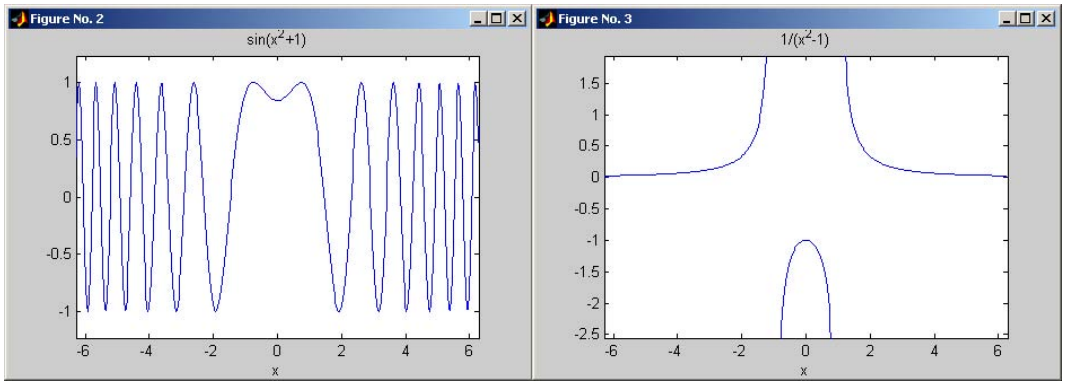
>> pretty(h)

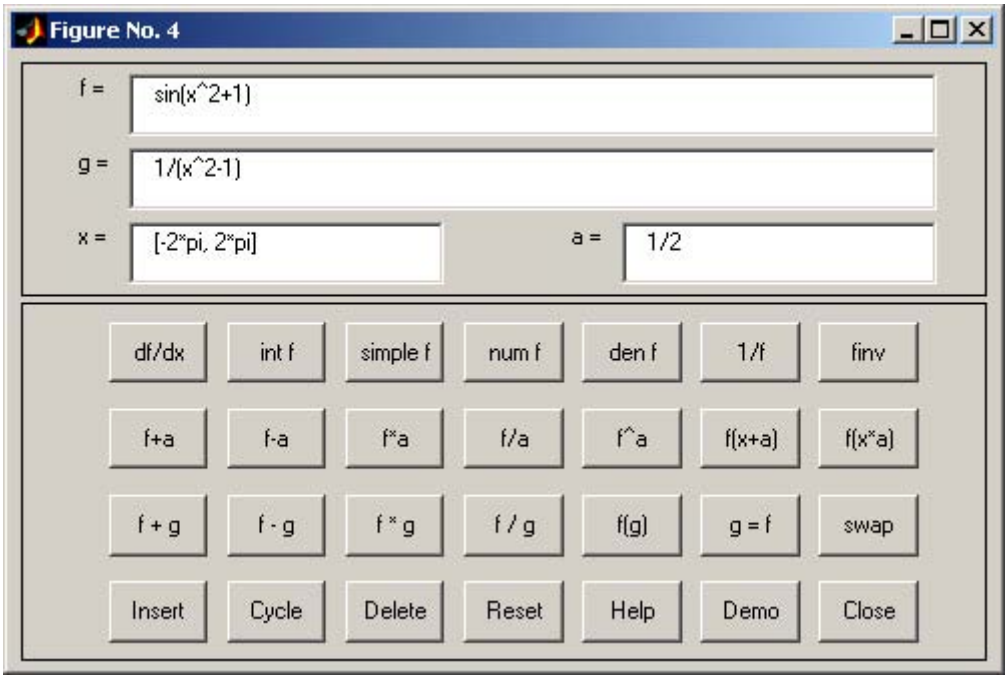
```
          1
---------------
     2     2
sin(x  + 1)  - 1
```

>> ezplot(h,[-3,3])



Το εργαλείο Symbolic Math Toolbox που μας δίνει μία ένα απλό γραφικό περιβάλλον, το **funtool**, με το οποίο μπορούμε να διερευνήσουμε τα όσα είδαμε παραπάνω με τη χρήση πλήκτρων. Η χρήση του είναι ιδιαίτερα απλή.





Κεφάλαιο 5

Όρια Πραγματικών συναρτήσεων

5.1 Παράδειγμα Χειρισμός ορίων στο MATLAB.

Υπολογίστε τα ακόλουθα όρια.

α) $\lim_{x\to 0}\frac{\sin(x)}{x}$

β) $\lim_{x\to -2}\frac{x^3-3x+2}{x^2-2x+8}$

και μελετήστε τη συμπεριφορά της

$$f(x)=\frac{1}{x}$$

Για να υπολογίσουμε όρια στο Matlab θα πρέπει να βασιστούμε στις συμβολικές δυνατότητες του περιβάλλοντος. Αυτό σημαίνει ότι θα πρέπει αρχικά να δηλώσουμε με τη χρήση της εντολής **syms** τις συμβολικές ποσότητες που θα περιέχονται στα όρια μας. Η εντολή με τη χρήση της οποίας υπολογίζουμε όρια είναι η **limit**. Στον πίνακα που ακολουθεί παρουσιάζεται η σύνταξή της.

Μαθηματική Έκφραση	**Υλοποίηση στο Matlab**
$\lim_{x\to 0} f(x)$	**limit(f,x)**
$\lim_{x\to a} f(x)$	**limit(f,x,a)**
$\lim_{x\to a-} f(x)$	**limit(f,x,a, 'left')**
$\lim_{x\to a+} f(x)$	**limit(f,x,a, 'right')**

Στον παραπάνω πίνακα το x είναι η μεταβλητή. Η f είναι η συνάρτηση, ως προς x. Η συνάρτηση μπορεί να γραφεί απευθείας μέσα στη limit ή να έχει ορισθεί πριν ως μία συμβολική ποσότητα ή με τη χρήση της inline. Για όρια στο άπειρο το a αντικαθιστάται με το inf. Η limit μπορεί να εφαρμοστεί και σε διάνυσμα συναρτήσεων. Κατά τη χρήση της limit χωρίς την εμφάνιση της παραμέτρου x το Matlab θεωρεί ότι μεταβλητή είναι η x και υπολογίζει το όριο.

Αρχικά ορίζουμε τις συμβολικές ποσότητες x,h,n που θα χειριστούμε.

```
>> clear all
>> syms x h n
```

Το όριο γνωστό $\lim_{x\to 0}\frac{\sin(x)}{x}$

```
>> limit(sin(x)/x)
 ans =
1
```

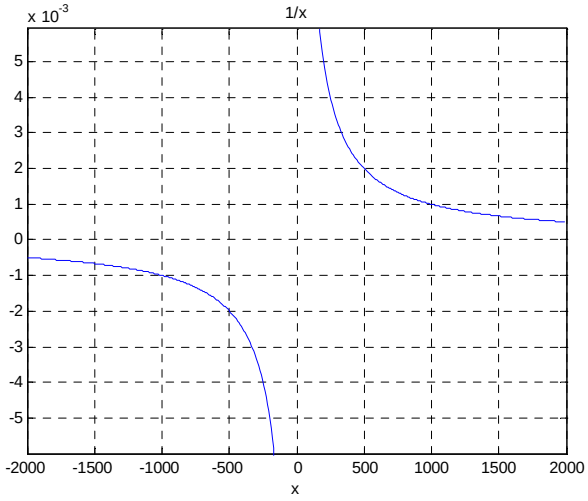
Το όριο $\lim_{x\to -2}\frac{x^3-3x+2}{x^2-2x+8}$

```
>> p=(x^3-3*x+2)/(x^2-2*x-8)
p =
(x^3-3*x+2)/(x^2-2*x-8)
>> pretty(p)
```

$$\frac{x^3-3x+2}{x^2-2x-8}$$

```
>> limit(p,-2)
ans =
-3/2
```

Η συμπεριφορά της $f(x)=\frac{1}{x}$




```
>> limit(1/x,x,0)
ans =
NaN
>> limit(1/x,x,0,'right')
ans =
inf
>> limit(1/x,x,0,'left')
ans =
-inf
>> limit(1/x,x,inf)
ans =
0
```

Κεφάλαιο 6

Παράγωγος

6.1 Παράδειγμα Ο υπολογισμός της παραγώγου στο MATLAB με βάση τον ορισμό της.

Υπολογίστε την παράγωγο της $f(x) = \cos(x)$ με τον ορισμό της παραγώγου.

Ο υπολογισμός της παραγώγου της $\cos(x)$ με τη χρήση του ορισμού μπορεί να γίνει με τις ακόλουθες εντολές.

```
>> g=inline('cos(x)','x')
g =
     Inline function:
     g(x) = cos(x)
>> limit((g(x+h)-g(x))/h,h,0)
ans =
 -sin(x)
```

6.2 Παράδειγμα Ο υπολογισμός της παραγώγου στο MATLAB.

Υπολογίστε την πρώτη και τη δεύτερη παράγωγο της συνάρτησης

$$f(x) = \cos(x^2 + 1)$$

Για να υπολογίσουμε παραγώγους συναρτήσεων στο Matlab μπορούμε πάλι να βασιστούμε στις συμβολικές δυνατότητες του εργαλείου Symbolic Math Toolbox του περιβάλλοντος. Αυτό σημαίνει ότι θα πρέπει αρχικά να δηλώσουμε με τη χρήση της εντολής **syms** τις συμβολικές ποσότητες που θα περιέχονται στα όρια μας. Η εντολή με τη χρήση της οποίας υπολογίζουμε όρια είναι η **diff**. Στον πίνακα που ακολουθεί παρουσιάζεται η σύνταξή της.

Μαθηματική Έκφραση	**Υλοποίηση στο Matlab**
$\frac{df(x)}{dx}$	**diff(f(x),x)**
$\frac{d^k f(x)}{dx^k}$	**diff(f(x),x,k)**

Στον παραπάνω πίνακα το x είναι η μεταβλητή. Η f είναι η συνάρτηση, ως προς x. Η συνάρτηση μπορεί να γραφεί απευθείας μέσα στη diff( ) ή να έχει ορισθεί πριν ως μία συμβολική ποσότητα ή με τη χρήση της inline( ). Η diff( ) μπορεί να εφαρμοστεί και σε διάνυσμα συναρτήσεων. Κατά τη χρήση της diff χωρίς την εμφάνιση της παραμέτρου x το Matlab θεωρεί ότι μεταβλητή είναι η x και υπολογίζει το όριο.

Αρχικά ορίζουμε τις συμβολικές ποσότητες που θα χειριστούμε και στη συνέχεια υπολογίζουμε την παράγωγο.

```
>> clear all
>> syms x n
>> f=inline(vectorize(sin(x^2+1)))

f =

    Inline function:
    f(x) = sin(x.^2+1)

>> diff(f(x),x)

ans =

2*cos(x^2+1)*x

>> pretty(ans)

                    2
2 cos(x  + 1) x
>> diff(f(x),2)

ans =

-4*sin(x^2+1)*x^2+2*cos(x^2+1)

>> pretty(ans)

                2       2          2
-4 sin(x  + 1) x  + 2 cos(x  + 1)
```

Κεφάλαιο 7

Βασικά Θεωρήματα Διαφορικού Λογισμού

7.1 Παράδειγμα Εύρεση ακρότατων και σημείων καμπής.

Θα μελετήσουμε ως προς τα ακρότατα και τα σημεία καμπής τη συνάρτηση

$$f(x) = (x+1)^2(x-1)^2$$

Αρχικά ορίζουμε το x ως συμβολική ποσότητα και τη συνάρτηση *f(x)*.

```
>> clear all
>> clf
>> syms x
>> f=(x-1)^2*(x+1)^2

f =

(x-1)^2*(x+1)^2
```

Θα μας χρειαστεί η πρώτη, η δεύτερη και η τρίτη παράγωγος της συνάρτησης, την οποία υπολογίζουμε με την diff.

```
>> df=diff(f,x)

df =

2*(x-1)*(x+1)^2+2*(x-1)^2*(x+1)

>> ddf=diff(f,x,2)

ddf =

2*(x+1)^2+8*(x-1)*(x+1)+2*(x-1)^2

>> dddf=diff(f,x,3)

dddf =

24*x
```

Βρίσκοντας τα κρίσιμα σημεία δηλαδή τις ρίζες της πρώτης παραγώγου και αντικαθιστώντας τις στη δεύτερη παράγωγο βρίσκουμε ότι το 0 είναι σημείο ολικού μεγίστου και τα 1 και -1 ολικού ελαχίστου.

```
>> s=solve(df,x)

s =

[  0]
[  1]
[ -1]
```

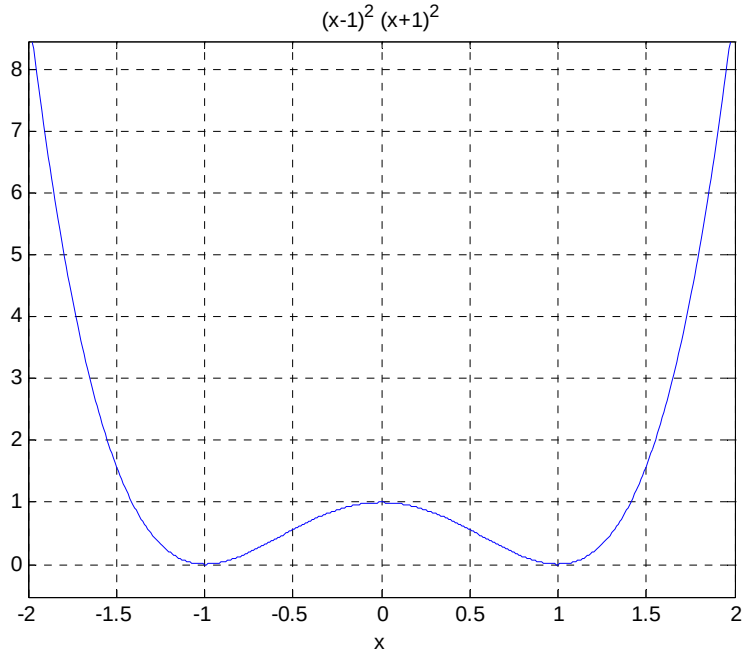
```
>> subs(ddf,s(1))

ans =

-4

>> subs(ddf,s(2))

ans =

8

>> subs(ddf,s(3))

ans =

8
```

Βρίσκοντας τις ρίζες της δεύτερης παραγώγου και αντικαθιστώντας τις στην τρίτη παράγωγο βρίσκουμε ότι τα σημεία $\pm\frac{\sqrt{3}}{3}$ αποτελούν σημεία καμπής.

```
>> ds=solve(ddf,x)

ds =

[  1/3*3^(1/2)]
[ -1/3*3^(1/2)]

>> subs(dddf,ds(1))

ans =

8*3^(1/2)

>> subs(dddf,ds(2))

ans =

-8*3^(1/2)
```

Όλα τα παραπάνω συμπεράσματα πιστοποιούνται και από το γράφημα της συνάρτησης.

```
>> ezplot(f,-2,2)
>> grid on
>>
```



7.2 Παράδειγμα Ένα πρόβλημα ελαχίστου.

Ένα κλειστό κυλινδρικό δοχείο με κυκλική βάση έχει χωρητικότητα 64 cm^3. Θα βρούμε τις διαστάσεις του ώστε το ποσό του μετάλλου που χρειάζεται για τα τοιχώματά του να είναι ελάχιστο.

Για να ελαχιστοποιήσουμε το ποσό του μετάλλου αρκεί να ελαχιστοποιήσουμε την επιφάνεια του κυλινδρικού δοχείου.
Έστω r, h, V, E η ακτίνα της βάσης, το ύψος, ο όγκος και το εμβαδό της επιφάνειας του δοχείου αντίστοιχα.
Έχουμε:

$$V = \pi r^2 h \Leftrightarrow \pi r^2 h = 64 \Leftrightarrow h = \frac{64}{\pi r^2}$$

$$E = 2\pi rh + 2\pi r^2 = 2\pi r \frac{64}{\pi r^2} + 2\pi r^2 = \frac{128}{r} + 2\pi r^2, r > 0$$

Το κρίσιμο σημείο της συνάρτησης του εμβαδού βρίσκεται Λύνοντας την:

$$E'(r) = 0$$

Στο MATLAB υπολογίζουμε την παράγωγο και λύνουμε την εξίσωση.

```
>> clear all
>> clf
>> syms x
>> clear all
>> syms r
>> E=128/r+2*pi*r^2

E =

128/r+2*pi*r^2
```

```
>> DE=diff(E,r)

DE =

-128/r^2+4*pi*r

>> s=solve(DE,r)

s =

[                              2/pi*4^(1/3)*(pi^2)^(1/3)]
[ -1/pi*4^(1/3)*(pi^2)^(1/3)+i*3^(1/2)/pi*4^(1/3)*(pi^2)^(1/3)]
[ -1/pi*4^(1/3)*(pi^2)^(1/3)-i*3^(1/2)/pi*4^(1/3)*(pi^2)^(1/3)]

>> s=simplify(s)

s =

[            2/pi^(1/3)*2^(2/3)]
[ 2^(2/3)*(-1+i*3^(1/2))/pi^(1/3)]
[ -2^(2/3)*(1+i*3^(1/2))/pi^(1/3)]
```

Οι μιγαδικές ρίζες δεν έχουν νόημα μιας και δεν μπορούμε να έχουμε μιγαδικές διαστάσεις. Θα εξετάσουμε τώρα, μελετώντας την παράγωγο $2^{ης}$ τάξης, αν το κρίσιμο σημείο που βρήκαμε είναι θέση ολικού ελαχίστου.

```
>> DDE=diff(E,r,2)

DDE =

256/r^3+4*pi
```

Παρατηρούμε ότι $E''(r) > 0, \forall r > 0$. Επομένως $r = 2\sqrt[3]{\frac{4}{\pi}}$ είναι θέση ολικού ελαχίστου. Για την τιμή αυτή της ακτίνας το ύψος είναι

```
>> h=64/(pi*s(1)^2)

h =

4/pi^(1/3)*2^(2/3)
```

Κεφάλαιο 8

Το ορισμένο ολοκλήρωμα

8.1 Παράδειγμα Υπολογισμός του ορισμένου ολοκληρώματος στο MATLAB

Υπολογίστε το $\int_0^{\pi} \sin(x)dx$

Ο υπολογισμός του ορισμένου ολοκληρώματος

$$\int_a^b f(x)dx$$

μπορεί να γίνει με δύο τρόπους. Ο πρώτος είναι χρησιμοποιώντας τις συμβολικές δυνατότητες του MATLAB και τη συνάρτηση συμβολικού (ακριβούς) υπολογισμού ολοκληρώματος **int( )**. Στην εντολή αυτή ως ορίσματα έχουμε την προς ολοκλήρωση συνάρτηση, τη μεταβλητή ολοκλήρωσης και τα άκρα. Οι δυνατότητες υπολογισμού ολοκληρωμάτων με αυτήν τη συνάρτηση δεν είναι απεριόριστες.

```
>> syms x
>> clear all
>> syms x
>> f=sin(x)
f =
sin(x)
>> int(f,x,0,pi)
ans =
2
```

Εναλλακτικά, μπορούμε να υπολογίσουμε το ολοκλήρωμα με τη συνάρτηση **quad( )**. Η συνάρτηση αυτή υπολογίζει με προσεγγιστικές μεθόδους το ολοκλήρωμα και δέχεται ως παραμέτρους τη συνάρτηση που έχει οριστεί με inline και τα άκρα της ολοκλήρωσης. Το αποτέλεσμα αποτελεί προσέγγιση της τιμής του ολοκληρώματος.

```
>> clear all
>> f=inline('sin(x)')
f =
     Inline function:
     f(x) = sin(x)
>> quad(f,0,pi)
ans =
   1.999999996398431e+000
```

8.2 Παράδειγμα Υλοποίηση του κανόνα του τραπεζίου και του κανόνα του Simpson στο MATLAB

Στις συναρτήσεις (functions) που ακολουθούν υλοποιούμε τον αριθμητικό υπολογισμό ορισμένου ολοκληρώματος με τη χρήση του κανόνα του τραπεζίου για την **trapez( )** και με τη χρήση του κανόνα του Simpson στην **simpson( )**. Ως ορίσματα οι functions, δέχονται την προς ολοκλήρωση συνάρτηση, τα άκρα του διαστήματος και τον αριθμό των υποδιαστημάτων στο οποίο θα εφαρμόσουμε τον κανόνα που υλοποιεί η function.

Κανόνας Τραπεζίου

```
function result = trapez(fname,a,b,n)
% The Trapezodial Rule is applied on each
% of n equal subintervals of [a,b].
% In
%   fname    The function f(x) that is defined on [a,b]. f should
%            return a column vector if x is a column vector.
%     a,b    real scalars
%       n    positive integer
% Working
% Out
%    the integral Approximation.
%
  m=2;
  coef = [1 1]'/2;
  d = (b-a)/n;
  h = d/(m-1);
  x = a+h*(0:(n*(m-1)))';
  x = linspace(a,b,n*(m-1)+1)';
  f = feval(fname,x);
  numI = 0;
  first = 1;
  last = m;
  result=0;
  for i=1:n
     %Add in the inner product for the i-th subintegral.
     result = result + coef'*f(first:last);
     first = last;
     last = last+m-1;
  end
  result = h*result;
```

Κανόνας Simpson

```
function result = simpson(fname,a,b,n)
% The Simpson Rule is applied on each
% of n equal subintervals of [a,b].
% In
%   fname    The function f(x) that is defined on [a,b]. f should
```

```
%           return a column vector if x is a column vector.
%    a,b    real scalars
%      n    positive integer
% Working
% Out
%   the integral Approximation.
%
  m=3;
  coef = [1 4 1]'/3;
  d = (b-a)/n;
  h = d/(m-1);
  x = a+h*(0:(n*(m-1)))';
  x = linspace(a,b,n*(m-1)+1)';
  f = feval(fname,x);
  numI = 0;
  first = 1;
  last = m;
  result=0;
  for i=1:n
    %Add in the inner product for the i-th subintegral.
    result = result + coef'*f(first:last);
    first = last;
    last = last+m-1;
  end
  result = h*result;
```

8.2 Παράδειγμα Σύγκριση του κανόνα του τραπεζίου και του κανόνα του Simpson στο MATLAB

Θα υπολογίσουμε προσεγγιστικά το $\int_0^{\pi} \sin(x)dx = 2$ τόσο με τον κανόνα του τραπεζίου όσο και με τον κανόνα του Simpson όταν εφαρμόζονται σε ένα έως πέντε υποδιαστήματα του $[0,\pi]$. Για καθεμία από τις προσεγγίσεις θα υπολογίσουμε το απόλυτο σφάλμα τους και θα τα εμφανίσουμε σε δύο πίνακες.

Στο MATLAB φτιάχνουμε το ακόλουθο script με όνομα testintegr.m

```
clear all
format long e
f=inline('sin(x)');
true=2;
for k=1:5,
   tr(k)=trapez(f,0,pi,k);
   errtr(k)=abs(tr(k)-true);
   sim(k)=simpson(f,0,pi,k);
   errsim(k)=abs(sim(k)-true);
end
[errtr' , errsim']
```

Εκτελώντας το έχουμε τα ακόλουθα αποτελέσματα:

```
>> testintegr

ans =

   2.000000000000000e+000   9.439510239319526e-002
   4.292036732051034e-001   4.559754984420739e-003
   1.862006357657822e-001   8.631896735358247e-004
   1.038811020629602e-001   2.691699483876597e-004
   6.623440190719498e-002   1.095173150043038e-004
```

Στην πρώτη στήλη έχουμε τα απόλυτα σφάλματα του κανόνα του τραπεζίου και στη δεύτερη τα απόλυτα σφάλματα του κανόνα του Simpson. Είναι φανερό ότι καλύτερη προσέγγιση δίνει ο κανόνας του Simpson.

Κεφάλαιο 9

Το αόριστο ολοκλήρωμα

9.1 Παράδειγμα Υπολογισμός του ορισμένου ολοκληρώματος στο MATLAB

Υπολογίστε τα $\int \sin(x)dx$, $\int \frac{x^2}{\sqrt{4-x^2}}dx$, $\int \frac{x^3}{\sqrt{x^2+25}}dx$.

Ο υπολογισμός του ορισμένου ολοκληρώματος

$$\int f(x)dx$$

μπορεί να γίνει μόνο χρησιμοποιώντας τις συμβολικές δυνατότητες του MATLAB και τη συνάρτηση συμβολικού (ακριβούς) υπολογισμού ολοκληρώματος **int( )**. Οι δυνατότητες υπολογισμού ολοκληρωμάτων με αυτήν τη συνάρτηση δεν είναι απεριόριστες.

```
>> clear all
>> syms x
>> int(sin(x),x)
ans =
-cos(x)
>> int(x^2/sqrt(4-x^2),x)
 ans =
-1/2*x*(4-x^2)^(1/2)+2*asin(1/2*x)
 >> int(x^3/sqrt(x^2+25),x)
 ans =
1/3*x^2*(x^2+25)^(1/2)-50/3*(x^2+25)^(1/2)
```

Κεφάλαιο 10

Εφαρμογές των ολοκληρωμάτων

10.1 Παράδειγμα Γενικευμένα ολοκληρώματα στο MATLAB

Υπολογίστε με την εντολή **int( )** τα γενικευμένα ολοκληρώματα $\int_0^{\infty} \frac{1}{x^2+1} dx$ και $\int_{-\infty}^{\infty} \frac{1}{x^2+2x+2} dx$.

```
>> syms x
>> int(1/(x^2+1),x,0,inf)
 ans =
 1/2*pi
 >> int(1/(x^2+2*x+2),x,-inf,inf)
 ans =
 pi
```

Κεφάλαιο 11

Σειρές Taylor Δυναμοσειρές

11.1 Παράδειγμα Οι σειρές στο MATLAB.

Αναπτύξτε σε σειρά Taylor την $f(x) = \sin(x)$

Για να προσεγγίσουμε μία συνάρτηση με ανάπτυγμα Taylor ή Maclaurin στο Matlab θα πρέπει να βασιστούμε στις συμβολικές δυνατότητες του περιβάλλοντος. Αυτό σημαίνει ότι θα πρέπει αρχικά να δηλώσουμε ως συμβολικές, με τη χρήση της εντολής **syms( ),** τις μεταβλητές ή τη συνάρτηση. Η εντολή με τη χρήση της οποίας υπολογίζουμε τα αναπτύγματα είναι η **Taylor( )**. Το αποτέλεσμα της Taylor είναι ένα πολυώνυμο και όχι μία άπειρη σειρά. Στον πίνακα που ακολουθεί παρουσιάζεται η σύνταξή της. Όταν η μεταβλητή της συνάρτησης είναι το x, η παράμετρος x στην κλίση της μπορεί να παραληφθεί. Η παράμετρος a καθορίζει το σημείο ως προς το οποίο αναπτύσσουμε. Όταν το παραλείπουμε το αποτέλεσμα είναι πολυώνυμο Maclaurin.

Μαθηματική Έκφραση	**Υλοποίηση στο Matlab**
$\sum_{k=0}^{n}(x-a)^k \frac{f^{(k)}(a)}{k!}$	**Taylor(f,n,x,a) ή Taylor(f,n,a)**
$\sum_{k=0}^{n}x^k \frac{f^{(k)}(0)}{k!}$	**Taylor(f,n,x) ή Taylor(f,n)**
$\sum_{k=0}^{5}x^k \frac{f^{(k)}(0)}{k!}$	**Taylor(f,x) ή Taylor(f)**

Θα αναπτύξουμε την $\sin(x)$

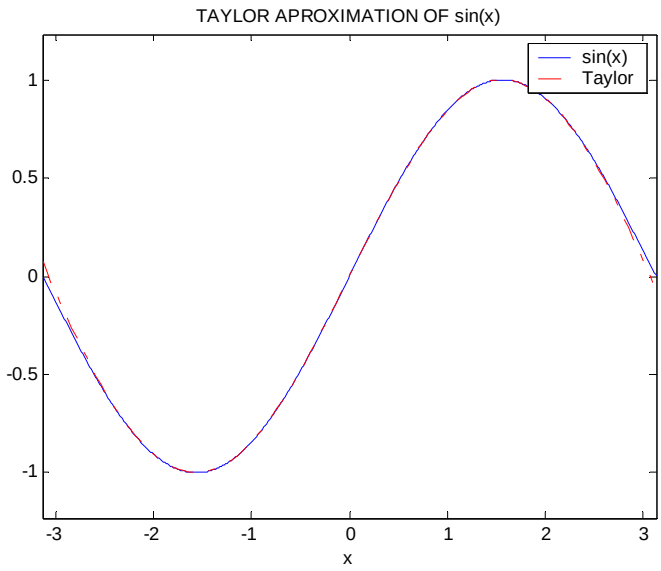
```
>> clear all
>> syms x
>> p=Taylor(sin(x),x,7,1)
p =
sin(1)+cos(1)*(x-1)-1/2*sin(1)*(x-1)^2-1/6*cos(1)*(x-1)^3+1/24*sin(1)*(x-
1)^4+1/120*cos(1)*(x-1)^5-1/720*sin(1)*(x-1)^6
>> pretty(p)
                                                2
3
 sin(1) + cos(1) (x - 1) - 1/2 sin(1) (x - 1)   - 1/6
cos(1) (x - 1)

                                 4                          5
        + 1/24 sin(1) (x - 1)  + 1/120 cos(1) (x - 1)
```

```
                                   6
          - 1/720 sin(1) (x - 1)
```

```
>> p=Taylor(sin(x),x,7)
p =
x-1/6*x^3+1/120*x^5
>> pretty(p)
                                        3          5
                                  x - 1/6 x  + 1/120 x
>> p=Taylor(sin(x),x,8)
 p =
 x-1/6*x^3+1/120*x^5-1/5040*x^7
 >> pretty(p)
                                      3          5           7
                             x - 1/6 x  + 1/120 x  - 1/5040 x
>> pretty(Taylor(sin(x)))
                                        3          5
                                  x - 1/6 x  + 1/120 x
```

Τα αποτελέσματα της προσέγγισης μπορούν να φανούν εκτελώντας το ακόλουθο πρόγραμμα εντολών Matlab (script).

```
clg;
syms x
f = sin(x);
p = taylor(f,8);
xd = -pi:0.05:pi; yd = subs(p,x,xd);
ezplot(f, [-pi,pi]); hold on;
plot(xd, yd, 'r-.')
title('TAYLOR APROXIMATION OF sin(x)');
legend('sin(x)','Taylor')
```



11.2 Παράδειγμα Ένα παράδειγμα Συνδυασμού στο MATLAB.

Αναπτύξτε σε σειρά Taylor δυνάμεων του $x-1$ την συνάρτηση $\ln x$ και υπολογίστε το $\lim_{x\to 1}\frac{\ln x}{x-1}$

```
>> clear all
>> syms x
```
Το όριο υπολογισμένο συμβολικά.
```
>> limit(log(x)/(x-1),x,1)
 ans =
 1
```
Το ανάπτυγμα Taylor.
```
 >> p=Taylor(log(x),7,1)
 p =
 x-1-1/2*(x-1)^2+1/3*(x-1)^3-1/4*(x-1)^4+1/5*(x-1)^5-1/6*(x-1)^6
 >> q=simple(p/(x-1))
 q =
 -1/6*x^5+31/30*x^4-163/60*x^3+79/20*x^2-71/20*x+49/20
 >> pretty(q)
                                  5    31   4    163   3    79   2    71
49
                      - 1/6 x   + -- x   - --- x   + -- x   - -- x
+ --
                                    30        60         20        20
20
```
Η τιμή και το όριο του αναπτύγματος στο x=1.
```
>> subs(q,x,1)
ans =
   1.0000
>> limit(q,x,1)
```

```
ans =
1
```

Κεφάλαιο 13

Εισαγωγή στις διαφορικές εξισώσεις

8.1 Παράδειγμα Λύση διαφορικών εξισώσεων στο MATLAB

Η λύση διαφορικών εξισώσεων και συστημάτων διαφορικών εξισώσεων στο MATLAB μπορεί να γίνει χρησιμοποιώντας τις συμβολικές δυνατότητες του MATLAB και τη συνάρτηση συμβολικού (ακριβούς) υπολογισμού λύσεων **dsolve( )**.

Χρησιμοποιώντας τη συνάρτηση **dsolve( )** θα υπολογίσουμε τη γενική λύση της διαφορικής εξίσωσης $x' = 2 - a \cdot x$.

```
>> clear all
>> syms x y a
>> dsolve('Dx=2-a*x')

ans =

2/a+exp(-a*t)*C1
```

Επίσης μπορούμε να λύσουμε και συστήματα διαφορικών εξισώσεων, όπως το

$$x' = y$$
$$y' = x$$

```
>> d=dsolve('Dx=y','Dy=x')

d =

    x: [1x1 sym]
    y: [1x1 sym]

>> d.x

ans =

1/2*C1*exp(-t)+1/2*C1*exp(t)+1/2*C2*exp(t)-1/2*C2*exp(-t)

>> d.y

ans =

1/2*C1*exp(t)-1/2*C1*exp(-t)+1/2*C2*exp(-t)+1/2*C2*exp(t)
```