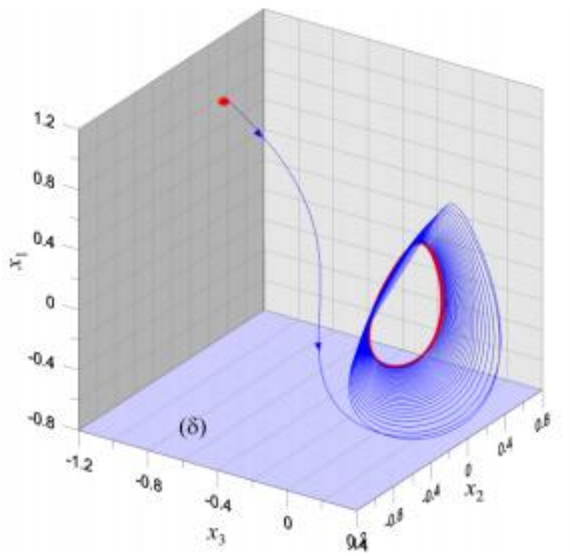
ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ Τ.Ε.

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΕΠΙΛΥΣΗ ΕΞΙΣΩΣΕΩΝ ΜΕ ΤΗ MATHEMATICA



ΣΠΟΥΔΑΣΤΕΣ: ΚΟΡΑΚΙΔΗΣ ΚΩΣΤΑΚΗΣ (Α.Μ. 6057)
ΜΑΡΟΥΔΑΣ ΣΑΒΒΑΣ (Α.Μ. 6003)

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΚΑΜΒΥΣΑΣ ΓΡΗΓΟΡΙΟΣ

ΠΑΤΡΑ 2018

ΠΡΟΛΟΓΟΣ

Το παρόν τεύχος αποτελεί την πτυχιακή εργασία που εκπονήθηκε στο Τμήμα Μηχανολόγων Μηχανικών Τ.Ε. της Σχολής Τεχνολογικών Εφαρμογών του Τεχνολογικού Εκπαιδευτικού Ιδρύματος Δυτικής Ελλάδας και έχει ως αντικείμενο την επίλυση εξισώσεων με χρήση του λογισμικού Mathematica.

Σκοπός της πτυχιακής αυτής εργασίας είναι αρχικά η εξοικείωση με τις βασικές εντολές της Mathematica και στη συνέχεια η χρήση της στην επίλυση συγκεκριμένων μαθηματικών προβλημάτων.

Θέλουμε να εκφράσουμε τις ευχαριστίες μας στον επιβλέποντα καθηγητή μας κ. Καμβύσα Γρηγόριο για την ανάθεση του θέματος και για την καθοδήγηση και το ενδιαφέρον του.

Κορακίδης Κωστάκης
Μαρούδας Σάββας

Νοέμβριος 2018

<u>Υπεύθυνη Δήλωση Σπουδαστών</u>: Οι κάτωθι υπογεγραμμένοι σπουδαστές έχουμε επίγνωση των συνεπειών του Νόμου περί λογοκλοπής και δηλώνουμε υπεύθυνα ότι είμαστε συγγραφείς αυτής της Πτυχιακής Εργασίας, αναλαμβάνοντας την ευθύνη επί ολοκλήρου του κειμένου εξ ίσου, έχουμε δε αναφέρει στην Βιβλιογραφία μας όλες τις πηγές τις οποίες χρησιμοποιήσαμε και λάβαμε ιδέες ή δεδομένα. Δηλώνουμε επίσης ότι, οποιοδήποτε στοιχείο ή κείμενο το οποίο έχουμε ενσωματώσει στην εργασία μας προερχόμενο από Βιβλία ή άλλες εργασίες ή το διαδίκτυο, γραμμένο ακριβώς ή παραφρασμένο, το έχουμε πλήρως αναγνωρίσει ως πνευματικό έργο άλλου συγγραφέα και έχουμε αναφέρει ανελλιπώς το όνομά του και την πηγή προέλευσης.

Οι σπουδαστές

Κορακίδης Κωστάκης	Μαρούδας Σάββας
……………………	……………………
(Υπογραφή)	(Υπογραφή)

ΠΕΡΙΛΗΨΗ

Στην εισαγωγή παρουσιάζονται γενικές πληροφορίες για τη Mathematica.

Στο πρώτο κεφάλαιο παρουσιάζονται όλες οι βασικές εντολές που χρησιμοποιούνται στους υπολογισμούς και στην κατασκευή γραφημάτων.

Το δεύτερο κεφάλαιο είναι αφιερωμένο στις μεθόδους επίλυσης μη γραμμικών αλγεβρικών εξισώσεων.

Στο τρίτο κεφάλαιο επιλύονται ενδεικτικά διάφορα προβλήματα γραμμικής άλγεβρας με τη Mathematica.

Το τέταρτο κεφάλαιο είναι αφιερωμένο στην παραγώγιση και ολοκλήρωση συναρτήσεων.

Στο πέμπτο κεφάλαιο παρουσιάζονται παραδείγματα επίλυσης διαφορικών εξισώσεων.

Το έκτο κεφάλαιο είναι αφιερωμένο στη μελέτη των μηχανικών ταλαντώσεων.

Στο έβδομο κεφάλαιο περιέχονται τα συμπεράσματα που προέκυψαν από την παρούσα εργασία.

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΗ

Το Mathematica είναι ένα υπολογιστικό πακέτο με πάρα πολλές δυνατότητες σε σχεδόν όλους τους τομείς των Μαθηματικών (Άλγεβρα, Θεωρία συνόλων, Ανάλυση, διαφορικές εξισώσεις, Στατιστική κ.ά.). Πρωτοεμφανίστηκε στα τέλη της δεκαετίας του 80 ως ένας πυρήνας εκτέλεσης εντολών (Kernel) ο οποίος μπορούσε να προσαρμοστεί σε κάθε λειτουργικό σύστημα όπως Unix, MacOs, Windows κ.ά.. Ο κοινός αυτός πυρήνας υπάρχει ακόμη και σήμερα αλλά οπωσδήποτε είναι βελτιωμένος και εμπλουτισμένος, ενώ η σύνδεσή του με τον χρήστη γίνεται μέσω ενός Notebook interface (περιβάλλον εργασίας) το οποίο είναι το μόνο που αλλάζει από λειτουργικό σε λειτουργικό.

Ο Stephen Wolfram είναι ο επιστήμονας ο οποίος δημιούργησε τη Mathematica. Ο Wolfram γεννήθηκε το 1959 στο Λονδίνο και πήρε το διδακτορικό δίπλωμα του στη Θεωρητική Φυσική από το πανεπιστήμιο του Caltech σε ηλικία μόλις 20 ετών. Ο Wolfram πολύ γρήγορα έγινε ένας από τους πρωταγωνιστές στον νέο κλάδο των επιστημονικών υπολογισμών στις δεκαετίες του 1970 και του 1980. Το 1979 δημιούργησε το SMP το πρώτο μοντέρνο υπολογιστικό σύστημα άλγεβρας το οποίο εμπορικά κυκλοφόρησε το 1981. Το κύριο ερευνητικό του ενδιαφέρον αφορούσε τις αρχές που διέπουν την πολυπλοκότητα που διέπει τα φυσικά φαινόμενα.

Αφού ολοκλήρωσε μία ιδιαίτερα επιτυχή ακαδημαϊκή καριέρα, ο Wolfram ίδρυσε το 1986 την εταιρεία Wolfram η οποία διέθεσε εμπορικά την πρώτη έκδοση του Mathematica στις 23 Ιουνίου του 1988. Η έκδοση αυτή σημείωσε σημαντική επιτυχία και καθιέρωσε την εταιρεία Wolfram ως μια από τις πρώτες εταιρείες σε παγκόσμια κατάταξη στην παραγωγή λογισμικού. Το 1991 κυκλοφόρησε η 2η έκδοση του Mathematica, ενώ ακολούθησαν οι εκδόσεις 3, 4 και 5 το 1996, το 1999 και το 2003 αντίστοιχα.

Σήμερα υπάρχουν περίπου 2.000.000 άνθρωποι που χρησιμοποιούν το Mathematica σε παγκόσμιο επίπεδο. Οι χρήστες του Mathematica ανήκουν σε διάφορους επιστημονικούς κλάδους όπως μαθηματικοί, φυσικοί, μηχανολόγοι, οικονομολόγοι και άλλοι επιστήμονες. Σε όλους αυτούς τους επιστήμονες τα μαθηματικά είναι ένα απαραίτητο εργαλείο

Παρόλα τα πλεονεκτήματα της ύπαρξης ενός κοινού πυρήνα στη Mathematica, δυστυχώς υπάρχουν και ορισμένα μειονεκτήματα. Τα σημαντικότερα από αυτά είναι η αρκετά χαμηλή ταχύτητα επεξεργασίας σε σύγκριση με τις καθαρές γλώσσες προγραμματισμού, η αστάθεια του προγράμματος και οι αυξημένες απαιτήσεις για μνήμη.

Η χαμηλή ταχύτητα γίνεται εμφανής κυρίως όταν ζητείται η επαναλαμβανόμενη εκτέλεση μιας ομάδας εντολών μέσω εντολών επανάληψης. Η κύρια αίτια της μειωμένης ταχύτητας επεξεργασίας είναι ότι η Mathematica χρησιμοποιεί interpreter και όχι compiler όπως οι κλασσικές γλώσσες προγραμματισμού (π.χ. Fortran, C, Pascal).

1. ΕΙΣΑΓΩΓΗ ΣΤΗ MATHEMATICA

1.1 ΓΕΝΙΚΑ

Το λογισμικό Mathematica χρησιμοποιώντας εκατοντάδες εντολές και συναρτήσεις έχει την δυνατότητα να πραγματοποιεί βασικά τρεις κατηγορίες εντολών:

1) *Αριθμητικούς υπολογισμούς*.

Πρόκειται *για* απλούς υπολογισμούς με τις συνήθεις πράξεις: +, − , *, / ,^ κ.τ.λ. καθώς επίσης και σύνθετους υπολογισμούς όπως για παράδειγμα τον αριθμητικό υπολογισμό των ριζών μιας εξίσωσης και τον αριθμητικό υπολογισμό ενός ορισμένου ολοκληρώματος.

2) Συμβολικό λογισμό.

Την εύρεση ή απλοποίηση εκφράσεων που περιέχουν μαθηματικές μεταβλητές. Για παράδειγμα την εύρεση της παραγώγου μιας συνάρτησης, την εύρεση ενός αορίστου ολοκληρώματος, την απλοποίηση ενός σύνθετου κλάσματος κ.ά.

3) Κατασκευή διαγραμμάτων.

Γραφικές παραστάσεις συναρτήσεων σε μία ή περισσότερες διαστάσεις.

Επίσης περιέχονται και πολλές εντολές χειρισμού δομών δεδομένων (λίστες, πίνακες), επανάληψης (Do, While), λογικές (If), εμφάνισης αποτελεσμάτων (Print) κ.ά. Με τις εντολές αυτές μπορούν να κατασκευαστούν ολοκληρωμένα προγράμματα.

Η Mathematica επίσης προσφέρει τη δυνατότητα να φορτωθούν επιπλέον πακέτα εντολών (packages) τα οποία περιέχουν εξειδικευμένες εντολές. Για παράδειγμα υπάρχουν πολλά Statistical Packages που περιέχουν επιπλέον εντολές και συναρτήσεις για στατιστική επεξεργασία δεδομένων.

Οι εντολές και οι συναρτήσεις του Mathematica είναι πολλές εκατοντάδες. Στη συνέχεια θα παρουσιαστούν πιο αναλυτικά οι πιο βασικές εντολές καθώς επίσης και ορισμένες εντολές που θα χρησιμοποιηθούν στις εφαρμογές που πραγματεύεται η παρούσα εργασία.

1.2 ΒΑΣΙΚΕΣ ΠΡΑΞΕΙΣ

Οι βασικές πράξεις που υποστηρίζει η *Mathematica*, καθώς και οι αντίστοιχοι τελεστές (τα σύμβολα) που χρησιμοποιούνται για την εκτέλεση τους, παρουσιάζονται στον πίνακα 1.1.

Πίνακας 1.1 Αριθμητικοί και συγκριτικοί τελεστές της Mathematica [2]

Αριθμητικοί τελεστές	Πράξη	Τελεστές σύγκρισης	Πράξη
+	συν	==	Ίσον (σύγκριση)
-	πλην	<	μικρότερο
*	επί	< =	μικρότερο ή ίσον
/	δια	>	μεγαλύτερο
^	δύναμη	> =	μεγαλύτερο ή ίσον
=	Ίσον (αντικατάσταση)	!=	όχι ίσο
!	παραγοντικό		

Τα ορίσματα των συναρτήσεων που υποστηρίζει η Mathematica γράφονται μέσα σε αγκύλες ενώ το πρώτο γράμμα κάθε συνάρτησης είναι κεφαλαίο. Οι κλασσικές μαθηματικές συναρτήσεις είναι οι παρακάτω:

Sqrt [x]
Exp [x]
Log [x]
Log [b,x] (λογάριθμος του x με βάση το b)
Sin [x]
Cos [x]
Tan [x]
ArcSin [x]
ArcCos [x]
ArcTan [x]
n!
Abs [x]
Round [x] (πλησιέστερος ακέραιος)
Floor [x] (ακέραιο μέρος)
Mod [n,m] (υπόλοιπο της διαίρεσης του n με το m)
Π.χ. η εντολή Mod[17,3] δίνει σαν αποτέλεσμα το 2.
Random [] (ψευδοτυχαίος αριθμός μεταξύ 0 και 1)
Random [{a,b}] (ψευδοτυχαίος αριθμός μεταξύ a και b.
Στην εντολή Random πρέπει να ορίσουμε αν θέλουμε ο τυχαίος αριθμός να είναι ακέραιος, πραγματικός ή μιγαδικός.
Max [x,y,…]
Min [x,y,…]
GCD[m,n] (επιστρέφει το μέγιστο κοινό διαιρέτη των m και n)
LCM[m,n] (επιστρέφει το ελάχιστο κοινό πολλαπλάσιο των m και n)
Οι συναρτήσεις GCD και LCM μπορούν να έχουν και περισσότερα από δύο ορίσματα. Για παράδειγμα αν γράψουμε: GCD[24,20,40] θα πάρουμε ως αποτέλεσμα 4 και αν γράψουμε LCM[24,20,40] θα πάρουμε 120.

```
Untitled-3 *

In[227]:=
        Round[2.7]
Out[227]= 3

In[228]:=
        Floor[2.7]
Out[228]= 2

In[229]:=
        Mod[17, 3]
Out[229]= 2

In[230]:=
        RandomReal[{0, 5}]
Out[230]= 0.102025

In[231]:= RandomInteger[{0, 1000}]
Out[231]= 232

In[232]:=
        RandomReal[]
Out[232]= 0.0630973

In[233]:=
        Max[7, 5.2, 15, 8.4]
Out[233]= 15
```

Σχήμα 1.1 Διάφοροι υπολογισμοί στη Mathematica.

```
Untitled-2 *

In[173]:=
        GCD[24, 20, 40]
Out[173]= 4

In[174]:=
        LCM[24, 20, 40]
Out[174]= 120
```

Σχήμα 1.2 Μέγιστος κοινός διαιρέτης και ελάχιστο κοινό πολλαπλάσιο.

1.3 ΣΥΜΒΟΛΙΚΟΣ ΛΟΓΙΣΜΟΣ

Όπως αναφέρθηκε ήδη το Mathematica έχει τη δυνατότητα εκτέλεσης πράξεων με σύμβολα. Αυτό επιτυγχάνεται με τις εντολές:
Expand [έκφραση] (ανάπτυγμα παράστασης σε άθροισμα απλών όρων)
Factor [έκφραση] (παραγοντοποίηση, αντίθετη της Expand)
Collect [έκφραση, x] (ομαδοποίηση δυνάμεων του x)

Untitled-1 *

In[165]:=
```
Apart[5 / (x^2 + 3 x + 2)]
```
Out[165]= $\frac{5}{1+x} - \frac{5}{2+x}$

In[166]:=
```
Together[3 x / (x - 2) + 4 / (x - 2)]
```
Out[166]= $\frac{4+3x}{-2+x}$

In[167]:=
```
Numerator[(3 x^3 + 2) / (7 x + 2)]
```
Out[167]= $2 + 3 x^3$

In[168]:=
```
Collect[3 x + 2 - 2 x^3 - 8 x + 5 x^2 - 9 x^3 - 2 + 4 x^2 - 6 x^3 - 2, x]
```
Out[168]= $-2 - 5 x + 9 x^2 - 17 x^3$

In[169]:=
```
Factor[x^2 + x - 12]
```
Out[169]= $(-3 + x)(4 + x)$

In[170]:=
```
Expand[(x - 3) * (x + 4)]
```
Out[170]= $-12 + x + x^2$

Σχήμα 1.3 Διάφοροι υπολογισμοί με τη Mathematica.

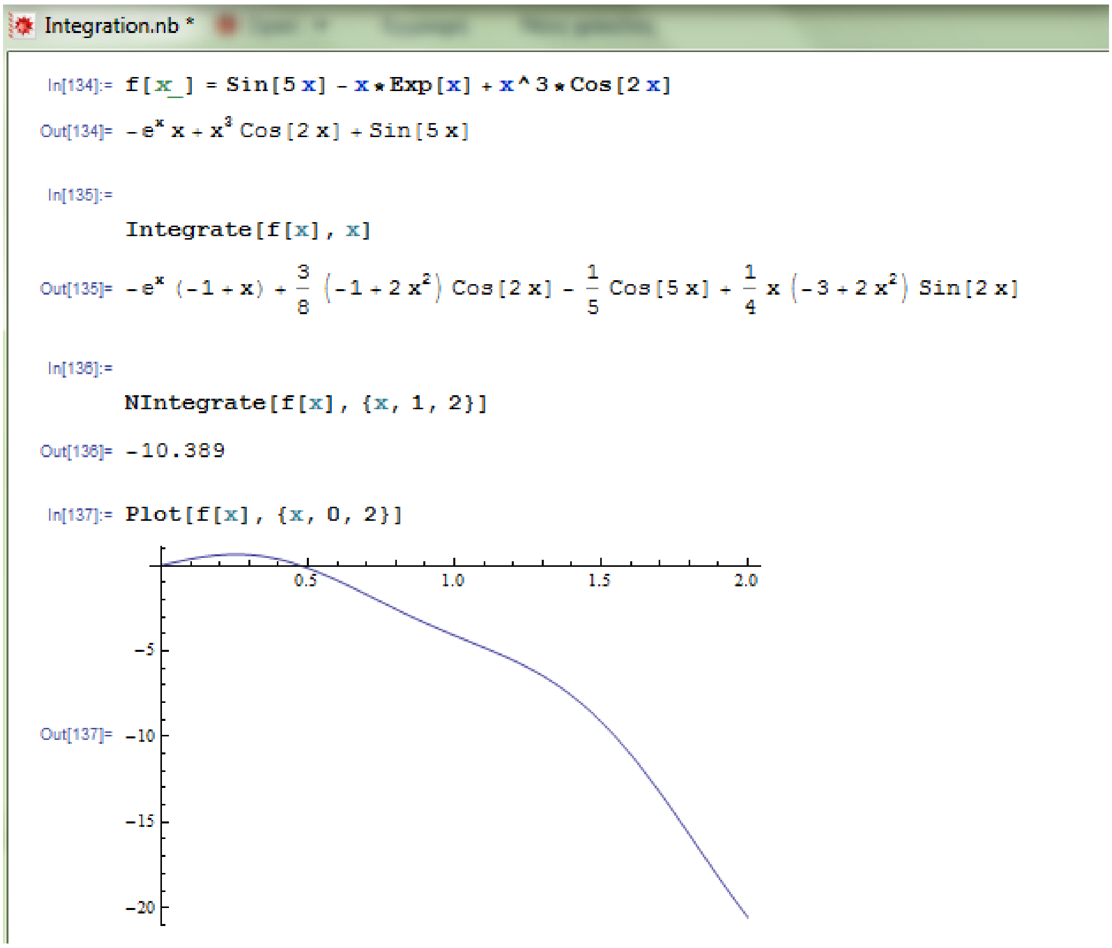
Simplify [έκφραση] (απλοποίηση έκφρασης με αλγεβρικούς μετασχηματισμούς)
FullSimplify [έκφραση] (απλοποίηση έκφρασης όχι μόνο με αλγεβρικούς μετασχηματισμούς)
Simplify [έκφραση, assumptions] (απλοποίηση έκφρασης με αλγεβρικούς μετασχηματισμούς υπό περιορισμούς)
Together [έκφραση] (πρόσθεση κλασμάτων που έχουν κοινό παρονομαστή)
Apart [έκφραση] (ανάλυση σε απλά κλάσματα)
Cancel [έκφραση] (απλοποίηση αριθμητή και παρονομαστή κλάσματος που περιέχουν κοινούς παράγοντες)
Numerator [έκφραση] (επιστρέφει τον αριθμητή μιας κλασματικής παράστασης)
Denominator [έκφραση] (επιστρέφει τον παρονομαστή μιας κλασματικής παράστασης)

```
Paragwgos 2.nb *

In[137]:= f[x_] := Cos[5*x] + Tan[x^2] - Exp[3 x - 2]

In[138]:= D[f[x], x]

Out[138]= -3 e^(-2+3 x) + 2 x Sec[x^2]^2 - 5 Sin[5 x]

In[139]:=
          D[f[x], {x, 2}]

Out[139]= -9 e^(-2+3 x) - 25 Cos[5 x] + 2 Sec[x^2]^2 + 8 x^2 Sec[x^2]^2 Tan[x^2]

In[140]:= N[f'[1]]

Out[140]= 3.49081

In[141]:=
          N[f''[1]]

Out[141]= 17.9744
```

Σχήμα 1.4 Υπολογισμός της πρώτης και της δεύτερης παραγώγου μιας συνάρτησης καθώς και των τιμών αυτών στο x=1.

D [f, x] (μερική παράγωγος της συνάρτησης f ως προς x)
D [f, x_1, …, x_n] (πολλαπλή μερική παράγωγος της συνάρτησης f ως προς x_1,…, x_n)
D [f, {x,n}] (μερική παράγωγος τάξης n της συνάρτησης f ως προς x)

Integrate [f, x] (επιστρέφει το αόριστο ολοκλήρωμα της συνάρτησης f)
Integrate [f, {x, xmin, xmax}] (υπολογίζει το ορισμένο ολοκλήρωμα της συνάρτησης f με όρια xmin και xmax)
Integrate [f, {x, xmin, xmax}, {y, ymin,ymax}] (υπολογίζει το διπλό ορισμένο ολοκλήρωμα της συνάρτησης f)



Σχήμα 1.5 Υπολογισμός ολοκληρώματος και σχεδίαση γραφικής παράστασης.

Series[f,{x,x0,order}] (αναπτύσσει τη συνάρτηση f σε δυναμοσειρά γύρω από το σημείο x0 έως την τάξη order)

Normal{Series] (αποκόπτει από τη σειρά τον όρο o(x^{order}) ώστε να είναι σε κανονική μορφή συνάρτησης)

SeriesCoefficient[series,k] (επιστρέφει το συντελεστή του x^k στη δοσμένη σειρά)

Sum[f,{i,imin,imax}] (επιστρέφει το άθροισμα της f από imin έως imax)

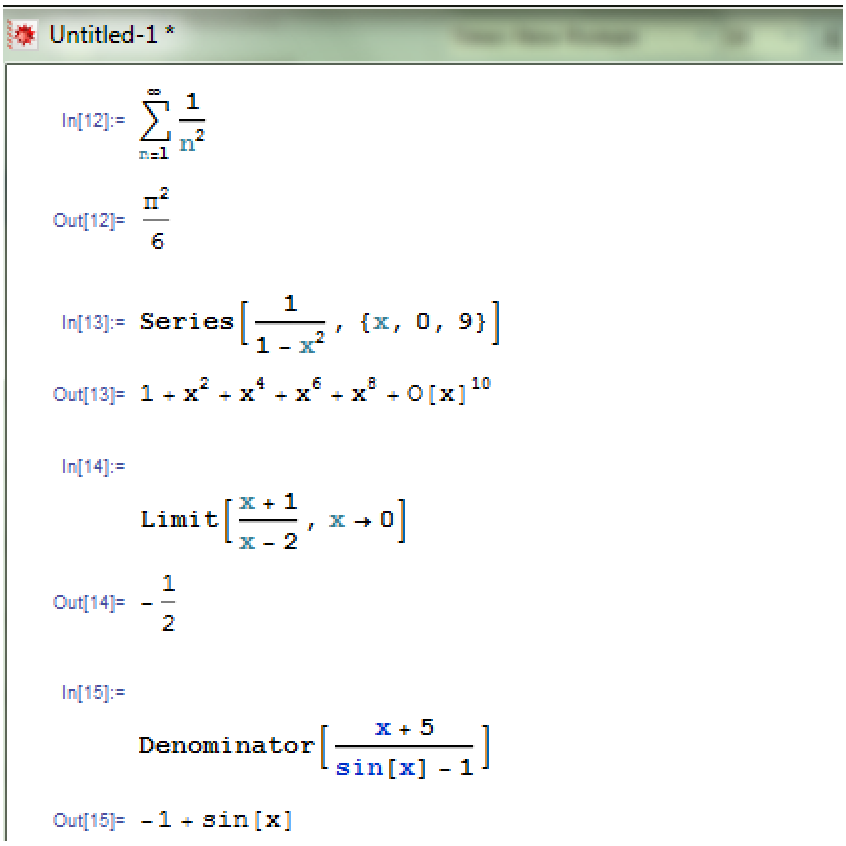
Product[f,{i,imin,imax}] (επιστρέφει το γινόμενο της f από imin έως imax)

Solve[f(x)==g(x),x] (επίλυση εξίσωσης ως προς x)

Solve[{f1=g1,f2=g2,...}, {x,y,...}] (επίλυση συστήματος εξισώσεων)

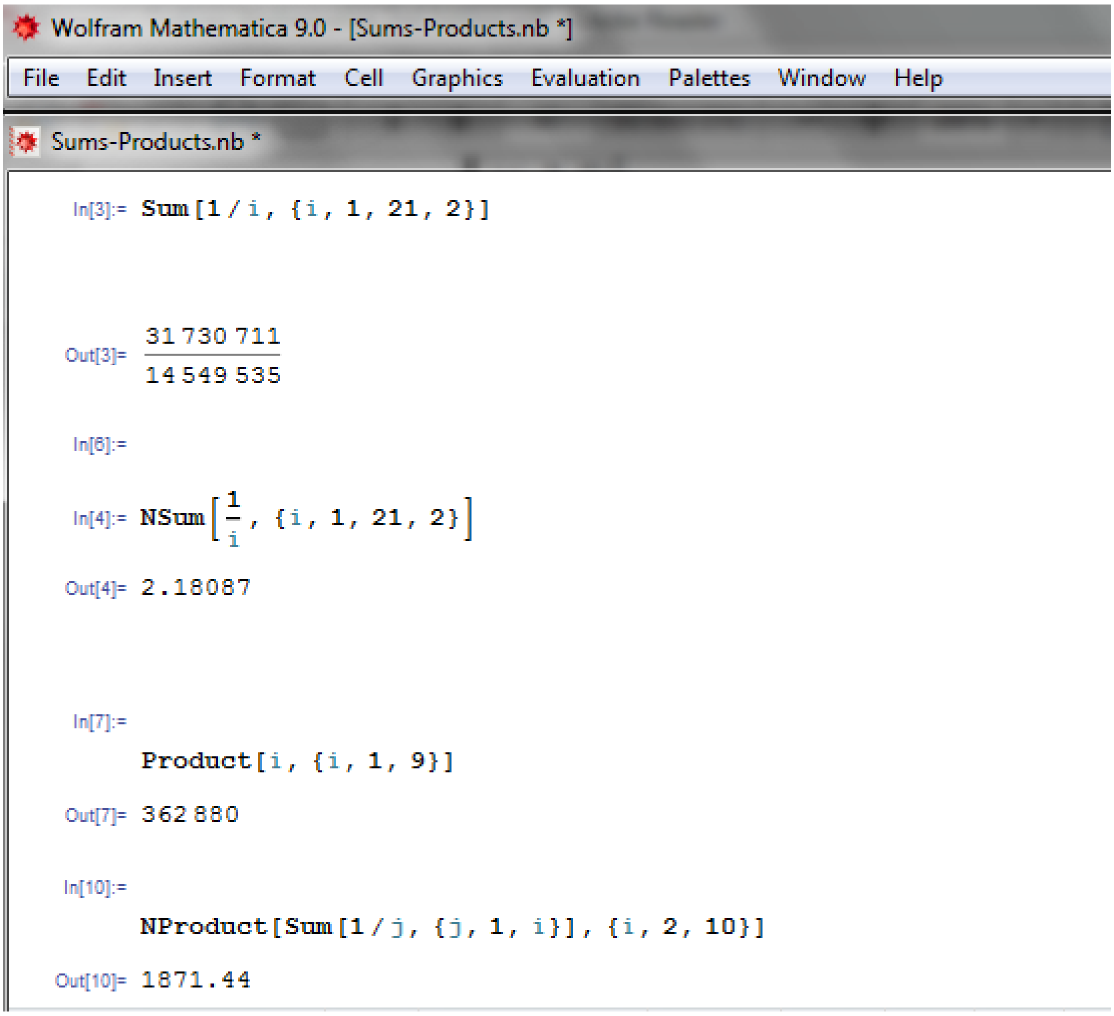
Limit[f,x->x0] (επιστρέφει το όριο της f όταν το x τείνει στο x0)

Dsolve[equation,y[x],x] (επίλυση διαφορικής εξίσωσης και εύρεση της συνάρτησης y της ανεξάρτητης μεταβλητής x, η οποία είναι η λύση της διαφορικής εξίσωσης)



Σχήμα 1.6 Παραδείγματα υπολογισμών με τη Mathematica.

Στο παράδειγμα που ακολουθεί υπολογίζονται: το άθροισμα $\sum_{1}^{21} n$ n=περιττός με την εντολή Sum, η ποσότητα 9! Με την εντολή product καθώς και η ποσότητα $\prod_{i=2}^{10}\sum_{j=1}^{i}\frac{1}{j}$ με χρήση και των δύο παραπάνω εντολών. Οι εντολές NSum και NProduct έχουν την ίδια σύνταξη και εργάζονται με παρόμοιο τρόπο με τις εντολές Sum και Product για να δώσουν αριθμητικές προσεγγίσεις των αποτελεσμάτων.



Σχήμα 1.7 Υπολογισμοί με τις εντολές Sum και Product.

1.4 ΕΝΤΟΛΕΣ ΕΛΕΓΧΟΥ

If[p, then, else] (εκτέλεση εντολών ανάλογα με το αν η πρόταση p είναι αληθής ή ψευδής)

Για τη διατύπωση της πρότασης *p* μπορούν να χρησιμοποιηθούν οι τελεστές: ==**,** != (≠)**,** >**,** >=**,** <=**,** !p (not)**,** p&&q (*and*)**,** p||q (or)

1.5 ΕΝΤΟΛΕΣ ΕΠΑΝΑΛΗΨΗΣ

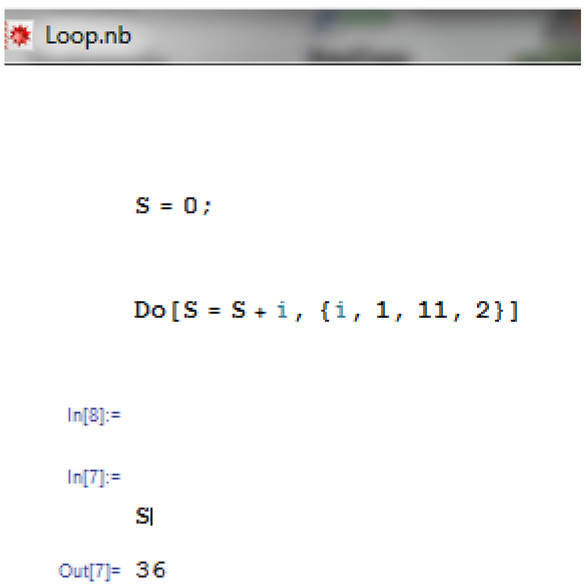
Οι πιο συχνά χρησιμοποιούμενες εντολές επανάληψης που διαθέτει η Mathematica είναι οι εξής:

Do[ομάδα εντολών, {i,imin,imax,di}] (εκτέλεση της ομάδας εντολών για τιμές του i από imin έως imax με βήμα di)

Do[ομάδα εντολών,{i,imin,imax,di},{j,jmin,jmax,dj},...] (πολλαπλή επανάληψη για όλες τις τιμές των i, j, …

While[p,ομάδα εντολών] (η ομάδα εντολών εκτελείται όσο η πρόταση p είναι αληθής)

Στο παράδειγμα που ακολουθεί υπολογίζεται το άθροισμα $\sum_{1}^{11} n$, n=περιττός με χρήση της εντολής Do.



Σχήμα 1.8 Υπολογισμός αθροίσματος με την εντολή Do.

1.6 ΠΡΑΞΕΙΣ ΜΕ ΠΙΝΑΚΕΣ

Οι κυριότερες εντολές χειρισμού πινάκων αναφέρονται στη συνέχεια. Στο παρακάτω σχήμα παρουσιάζονται ορισμένα παραδείγματα των εντολών που διαθέτει η Mathematica για τις πράξεις των πινάκων. Περισσότερα παραδείγματα παρουσιάζονται στο κεφάλαιο 3.

```
Untitled-3 *

In[320]:=
        t = DiagonalMatrix[{1, 2, 3}]

Out[320]= {{1, 0, 0}, {0, 2, 0}, {0, 0, 3}}

In[321]:=
        TableForm[t]

Out[321]//TableForm=
        1    0    0
        0    2    0
        0    0    3

In[322]:=
        t[[2, 2]]

Out[322]= 2

In[323]:=
        Transpose[t]

Out[323]= {{1, 0, 0}, {0, 2, 0}, {0, 0, 3}}

In[324]:=
        t1 = Inverse[t]

Out[324]= {{1, 0, 0}, {0, 1/2, 0}, {0, 0, 1/3}}

In[325]:=
        TableForm[t1]

Out[325]//TableForm=
        1    0    0
        0    1/2  0
        0    0    1/3

In[326]:=
        Det[t]

Out[326]= 6
```

Σχήμα 1.9 Παραδείγματα εντολών χειρισμού πινάκων.

t=Table[expr, {i,imin,imax,di},{..},..] (δημιουργία πίνακα μιας ή περισσότερων διαστάσεων)
a[[i,j]] (επιστρέφει το στοιχείο του πίνακα που βρίσκεται στην i-οστή γραμμή και την j-οστή στήλη)
TableForm[t] (παρουσιάζει τον πίνακα σε κανονική μορφή και όχι σε μορφή λίστας)
Inverse[T] (επιστρέφει τον αντίστροφο του πίνακα)
Det[T] (επιστρέφει την ορίζουσα του πίνακα)
Transpose[T] (επιστρέφει τον ανάστροφο του πίνακα)
MatrixPower[T,n] (υψώνει τον πίνακα στη δύναμη n)
Eigenvalues[T] (υπολογίζει τις ιδιοτιμές του πίνακα)
Eigenvectors[T] (υπολογίζει τα ιδιοδιανύσματα του πίνακα)
T1.T2 (γινόμενο των πινάκων T1 και T2
IdentityMatrix[n] (επιστρέφει τον μοναδιαίο πίνακα nxn)
DiagonalMatrix[λίστα] (επιστρέφει το διαγώνιο πίνακα με διαγώνια στοιχεία αυτά που περιέχονται στη λίστα)

1.7 ΛΙΣΤΕΣ

Σε πολλές περιπτώσεις είναι χρήσιμο να έχουμε τη δυνατότητα να θεωρούμε μία συλλογή από στοιχεία ως μία μεταβλητή. Τέτοιου είδους μεταβλητές υποστηρίζονται από τη Mathematica και ονομάζονται λίστες (lists). Το μέγεθος των λιστών μπορεί να είναι οσοδήποτε. Μοναδικός περιορισμός είναι φυσικά η μνήμη του συστήματος που έχουμε στη διάθεσή μας. Τα στοιχεία μιας λίστας μπορεί να είναι αριθμοί, αλγεβρικές παραστάσεις, γραφήματα κ.ά., ή ακόμα και άλλες λίστες.

Για παράδειγμα η λίστα L που περιέχει τους αριθμούς 1, 3, 5 μπορεί να οριστεί με την εντολή: L{1,3,5}. Η εντολή L[[n]] επιστρέφει το n-οστό στοιχείο της λίστας. Επίσης είναι δυνατό να γίνουν πράξεις ανάμεσα σε λίστες.

1.8 ΟΡΙΣΜΟΣ ΜΕΤΑΒΛΗΤΗΣ

Η γενική μεταβλητή με όνομα x ορίζεται με την εντολή: x = τιμή και καταργείται με την εντολή x = . ή με την εντολή Clear[x]. Το όνομα της μεταβλητής μπορεί να είναι οποιοσδήποτε συνδυασμός χαρακτήρων. Αφού οριστούν οι γενικές μεταβλητές (global variables) εξακολουθούν να ισχύουν έως ότου τις καταργήσουμε, ή κλείσουμε το πρόγραμμα ή τον πυρήνα.

Η Mathematica δίνει τη δυνατότητα χρήσης τοπικών μεταβλητών οι οποίες έχουν ισχύ μόνο σε μία εντολή. Μετά την εκτέλεση της εντολής αυτής παύουν να ισχύουν. Οι τοπικές μεταβλητές ορίζονται με την εντολή: έκφραση/.x->value ή με την εντολή: expr/.{x->valuex,y->... } για περισσότερες από μία.

1.9 ΓΡΑΦΗΜΑΤΑ

Για την κατασκευή διδιάστατων γραφημάτων είναι χρήσιμες οι εντολές:

Plot[f,{x,xmin,xmax}] (εμφανίζει τη γραφική παράσταση της συνάρτησης f (*x*) για τιμές του x μεταξύ του xmin και του xmax)

Plot[{f1,f2,...},{x,xmin,xmax}] (εμφανίζει τις γραφικές παραστάσεις δύο συναρτήσεων σε κοινό σύστημα αξόνων)

Plot[Evaluate[f],{x,xmin,xmax}] (εμφανίζει τη γραφική παράσταση της f αφού πρώτα εκτελέσει τυχόν εντολές μέσα στην f)

Show[p1,...,pn, option->value] (εμφανίζει μαζί τα γραφήματα με ονόματα p1,…, pn.

1.10 ΔΙΑΦΟΡΕΣ ΕΝΤΟΛΕΣ

Ορισμός συνάρτησης

Ο ορισμός της συνάρτησης με όνομα f και όρισμα x πραγματοποιείται μέσω της εντολής: f[x_]:= έκφραση.

Παράδειγμα: f[x_]:= x^3.Η εντολή ?f, : εμφανίζει τον τύπο της συνάρτησης f. Η εντολή f[2] : επιστρέφει την τιμή της f για x=2, ενώ η εντολή Clear[f]: καταργεί τη συνάρτηση f.

Εμφάνιση αποτελεσμάτων

Η εμφάνιση αποτελεσμάτων γίνεται με την εντολή: Print[όνομα μεταβλητής,"μήνυμα",όνομα μεταβλητής,…].

2. ΕΠΙΛΥΣΗ ΑΛΓΕΒΡΙΚΩΝ ΕΞΙΣΩΣΕΩΝ

2.1 ΓΕΝΙΚΑ

Στο κεφάλαιο αυτό θα ασχοληθούμε με αριθμητικές μεθόδους για την εύρεση των ριζών μη-γραμμικών αλγεβρικών εξισώσεων. Θα μελετηθούν οι παρακάτω μέθοδοι:

Μέθοδος Διχοτόμησης

Μέθοδος των Διαδοχικών Προσεγγίσεων

Μέθοδος Newton-Raphson

Μέθοδος Τέμνουσας

και θα παρουσιαστούν παραδείγματα στη Mathematica.

Όπως θα δούμε, οι περισσότερες μέθοδοι εύρεσης ρίζας χρειάζονται μια αρχική προσέγγιση της λύσης (ή και περισσότερες), την οποία βελτιώνουν σε κάθε στάδιο της επίλυσης. Η αριθμητική τους ευστάθεια προσδιορίζεται από τη συμπεριφορά τους σε μεταβολές αυτής της αρχικής τιμής. Μια μέθοδος είναι ευσταθής αν οποιαδήποτε κατάλληλα μικρή μεταβολή της αρχικής τιμής δεν επηρεάζει την εύρεση της ρίζας, ενώ είναι ασταθής αν μια μικρή μεταβολή της αρχικής προσέγγισης οδηγεί μακριά από τη ρίζα.

2.2 ΜΕΘΟΔΟΣ ΔΙΧΟΤΟΜΗΣΗΣ

Η μέθοδος της διχοτόμησης (bisection method) βασίζεται στο γνωστό από την Ανάλυση θεώρημα του Bolzano:

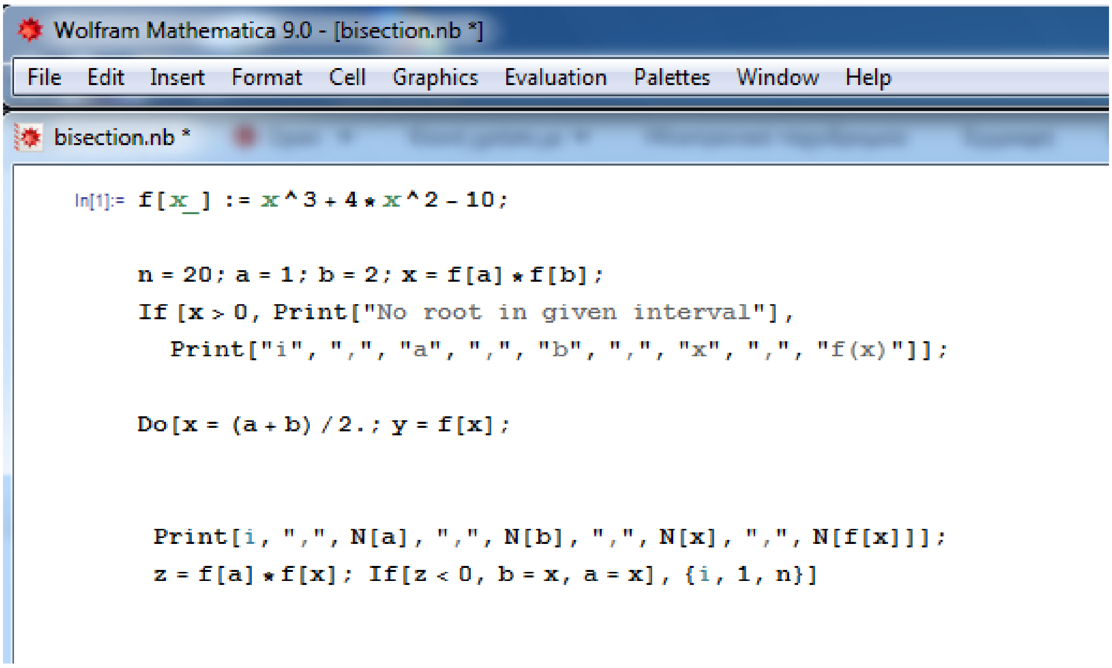
Αν η *f*(*x*) είναι συνεχής στο [a,b] και έχουμε f(a)·f(b) < 0 , τότε, υπάρχει τουλάχιστον μία ρίζα της *f*(*x*) στο (a,b).

Η μέθοδος διχοτομεί το διάστημα [a,b], εντοπίζει τη ρίζα σε ένα από τα δύο υποδιαστήματα και επαναλαμβάνεται στο επιλεγμένο υποδιάστημα.

Στη συνέχεια παρουσιάζεται η επίλυση στη Mathematica της εξίσωσης με τη μέθοδο της διχοτόμησης:

$$x^3 + 4x^2 - 10 = 0$$

Γίνεται αναζήτηση της ρίζας στο διάστημα [1,2]. Ισχύει f(1)=-5 και f(2)=14 και συνεπώς ισχύει το θεώρημα Bolzano για το διάστημα αυτό.



```
i,a,b,x,f(x)
1,1.,2.,1.5,2.375
2,1.,1.5,1.25,-1.79688
3,1.25,1.5,1.375,0.162109
4,1.25,1.375,1.3125,-0.848389
5,1.3125,1.375,1.34375,-0.350983
6,1.34375,1.375,1.35938,-0.0964088
7,1.35938,1.375,1.36719,0.0323558
8,1.35938,1.36719,1.36328,-0.03215
9,1.36328,1.36719,1.36523,0.0000720248
10,1.36328,1.36523,1.36426,-0.0160467
11,1.36426,1.36523,1.36475,-0.00798926
12,1.36475,1.36523,1.36499,-0.0039591
13,1.36499,1.36523,1.36511,-0.00194366
14,1.36511,1.36523,1.36517,-0.000935847
15,1.36517,1.36523,1.3652,-0.000431919
16,1.3652,1.36523,1.36522,-0.000179949
17,1.36522,1.36523,1.36523,-0.0000539625
18,1.36523,1.36523,1.36523,9.03099×10^-6
19,1.36523,1.36523,1.36523,-0.0000224658
20,1.36523,1.36523,1.36523,-6.71741×10^-6
```

Σχήμα 2.1 Παράδειγμα εφαρμογής της μεθόδου της διχοτόμησης στη Mathematica.

Εκτελούνται n=20 επαναλήψεις. Με την εντολή Do συνεχώς υποδιαιρούμε το διάστημα στο οποίο βρίσκεται η ζητούμενη ρίζα. Με τη μεταβλητή z, και με βάση το θεώρημα Bolzano, ελέγχουμε σε ποιο υποδιάστημα βρίσκεται η ρίζα κάθε φορά.

2.3 ΜΕΘΟΔΟΣ NEWTON-RAPHSON

Η μέθοδος Newton–Raphson είναι επαναληπτική μέθοδος της μορφής $x = g(x)$. Η επιλογή της συνάρτησης $g(x)$ γίνεται ως εξής:
Έστω ότι αναζητούμε τη ρίζα της συνεχούς και διαφορίσιμης, σε διάστημα [a, b], συνάρτησης $f(x)$. Αν γνωρίζουμε την τιμή αυτής και των παραγώγων της σε κάποιο σημείο $x_0 \in$ [a, b], το θεώρημα του Taylor, μας εξασφαλίζει ότι στη ρίζα, $\rho \in$ [a, b], ισχύει:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Πρέπει να παρατηρήσουμε ότι σε κάθε επανάληψη της μεθόδου πρέπει να υπολογίσουμε τις τιμές δύο συναρτήσεων.

Ακολουθεί η εύρεση μέσω του Mathematica της τετραγωνικής ρίζας του. Με άλλα λόγια, επιλύεται η εξίσωση:

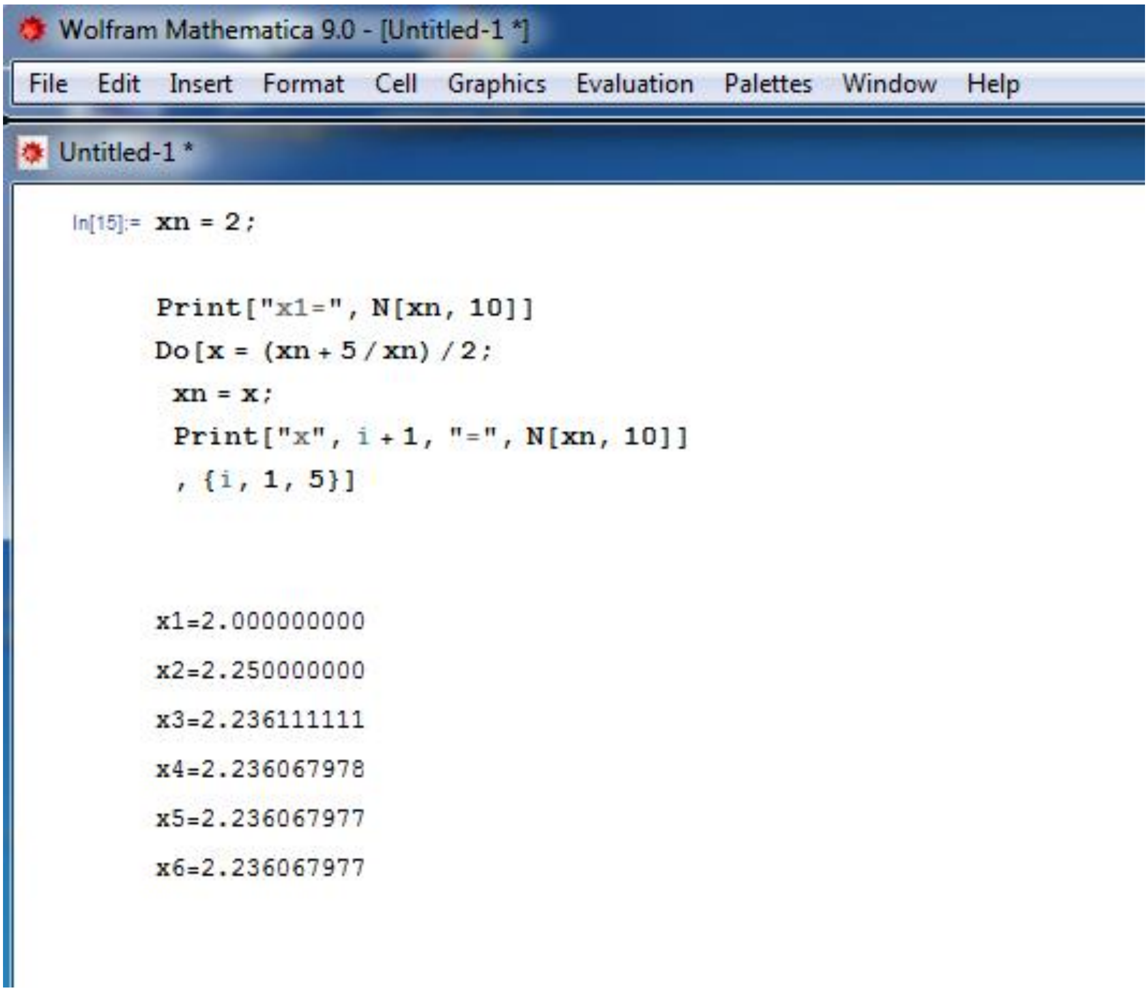
$$f(x) = x^2 - 5 = 0$$

Η παράγωγος είναι: $f'(x) = 2x$ και ο τύπος της μεθόδου Newton-Raphson γράφεται:

$$x_{n+1} = x_n - \frac{{x_n}^2 - 5}{2x_n}$$

Η εξίσωση αυτή γράφεται στην πιο βολική μορφή:

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{5}{x_n}\right)$$



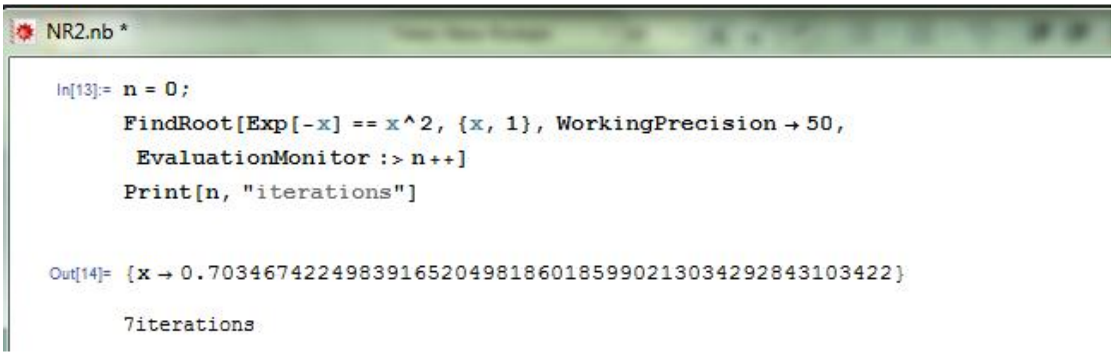
Σχήμα 2.2 παράδειγμα εφαρμογής της μεθόδου Newton-Raphson στη Mathematica.

Στο παραπάνω παράδειγμα το x υπολογίζεται από την προηγούμενη αναδρομική σχέση και στη συνέχεια το xn παίρνει την τιμή του x ώστε ν υπολογισθεί η επόμενη προσέγγιση κ.ο.κ. Με την εντολή N[xn,10] μέσα στην εντολή Print τυπώνεται η αριθμητική του xn με 10 σημαντικά ψηφία.

Στο προηγούμενο παράδειγμα ορίσαμε εξαρχής ότι θα γίνουν 5 επαναλήψεις της μεθόδου Newton-Raphson, αδιαφορώντας για την ακρίβεια της λύσης. Στη συνέχεια θα χρησιμοποιήσουμε την εντολή FindRoot της Mathematica. Η εντολή αυτή χρησιμοποιεί τη μέθοδο Newton-Raphson όταν ο χρήστης δίνει ένα αρχικό σημείο. Μας δίνει επίσης τη δυνατότητα να καθορίσουμε εμείς την ακρίβεια που επιθυμούμε στον υπολογισμό της ρίζας μιας αλγεβρικής εξίσωσης. Η προκαθορισμένη αυτή ακρίβεια αποτελεί και το κριτήριο τερματισμού της εκτέλεσης του επαναληπτικού αλγορίθμου.

Στο παράδειγμα που ακολουθεί ζητάμε ακρίβεια 50 ψηφίων στον υπολογισμό της ρίζας της εξίσωσης:

$$e^{-x} = x^2$$



Σχήμα 2.3 Μέθοδος Newton-Raphson με κριτήριο τερματισμού.

Όπως βλέπουμε η μέθοδος Newton-Raphson χρειάστηκε 7 επαναλήψεις για να φτάσει στην επιθυμητή ακρίβεια.

2.3 ΜΕΘΟΔΟΣ ΔΙΑΔΟΧΙΚΩΝ ΠΡΟΣΕΓΓΙΣΕΩΝ

Έστω ότι η εξίσωση f(x)=0 γράφεται στη μορφή:

$$x=g(x)$$

όπου η g θεωρείται ότι είναι μία συνεχής συνάρτηση. Η g στην περίπτωση αυτή ονομάζεται και επαναληπτική συνάρτηση. Αν x* είναι μία ρίζα της f(x), τότε, επειδή θα ισχύει f(x*)=0, θα ισχύει και:

$$x^*=g(x^*)$$

Η ρίζα x* ονομάζεται και σταθερό σημείο της g(x). Θεωρώντας τώρα μία αρχική εκτίμηση, έστω x_0, σχηματίζουμε την ακολουθία:

$$x_{i+1}=g(x_i)$$

Αν η $g(x_i)$ συγκλίνει, τότε οδηγεί στην εύρεση της λύσης, δηλαδή:

$$\lim_{i\to+\infty} x_{i+1} = x^*$$

Η μέθοδος αυτή ονομάζεται μέθοδος των διαδοχικών προσεγγίσεων (fixed point iteration). Στη συνέχεια χρησιμοποιούμε τη μέθοδο των διαδοχικών προσεγγίσεων για να επιλύσουμε την εξίσωση:

$$x^3 + 2x^2 + 10x = 20$$

Καταρχάς γράφουμε την εξίσωση στη μορφή:

$$x = \frac{20}{x^2 + 2x + 10}$$

Δηλαδή: $g(x) = \dfrac{20}{x^2 + 2x + 10}$

Και η αναδρομική σχέση γράφεται:

$$x_{i+1} = \frac{20}{x_i^{\,2} + 2x_i + 10}$$

Στη Mathematica ορίζουμε αρχικά τη συνάρτηση:

$$f(x) = \frac{20}{x^2 + 2x + 10}$$

Γίνονται 20 επαναλήψεις με αρχική τιμή x_0=1.

Wolfram Mathematica 9.0 - [Untitled-1 *]

File Edit Insert Format Cell Graphics Evaluation Palettes Window Help

Untitled-1 *

```
In[2]:= f[x_] := 20 / (x^2 + 2 * x + 10)

        n = 20; x = 1; Print["i", ",", "x_i"];
        Do[y = f[x]; Print[i, ",", N[y, 10]]; x = y, {i, 1, n}]
```

```
i,x_i
1,1.538461538
2,1.295019157
3,1.401825309
4,1.354209390
5,1.375298092
6,1.365929788
7,1.370086003
8,1.368241024
9,1.369059812
10,1.368696398
11,1.368857689
12,1.368786103
13,1.368817874
14,1.368803773
15,1.368810032
16,1.368807254
17,1.368808487
18,1.368807940
19,1.368808182
20,1.368808075
```

Σχήμα 2.4 Παράδειγμα εφαρμογής της μεθόδου των διαδοχικών προσεγγίσεων στη Mathematica.

2.4 ΜΕΘΟΔΟΣ ΤΕΜΝΟΥΣΑΣ

Σύμφωνα με τη μέθοδο της τέμνουσας (secant method), προσεγγίζουμε τη συνάρτηση *f*(*x*) με ευθεία που περνάει από τα δύο σημεία: (x_{n-1}, $f(x_{n-1})$) και (x_n, $f(x_n)$). Τα x_{n-1} και x_n είναι διαδοχικές προσεγγίσεις της ρίζας. Η νέα προσέγγιση, x_{n+1}, είναι η τομή της προσεγγιστικής ευθείας με τον άξονα *x*, δηλαδή η ρίζα που ψάχνουμε. Όπως είναι φανερό, πρόκειται για μέθοδο δύο σημείων, αφού πρέπει να επιλέξουμε δύο αρχικά σημεία.

Η αναδρομική σχέση είναι:

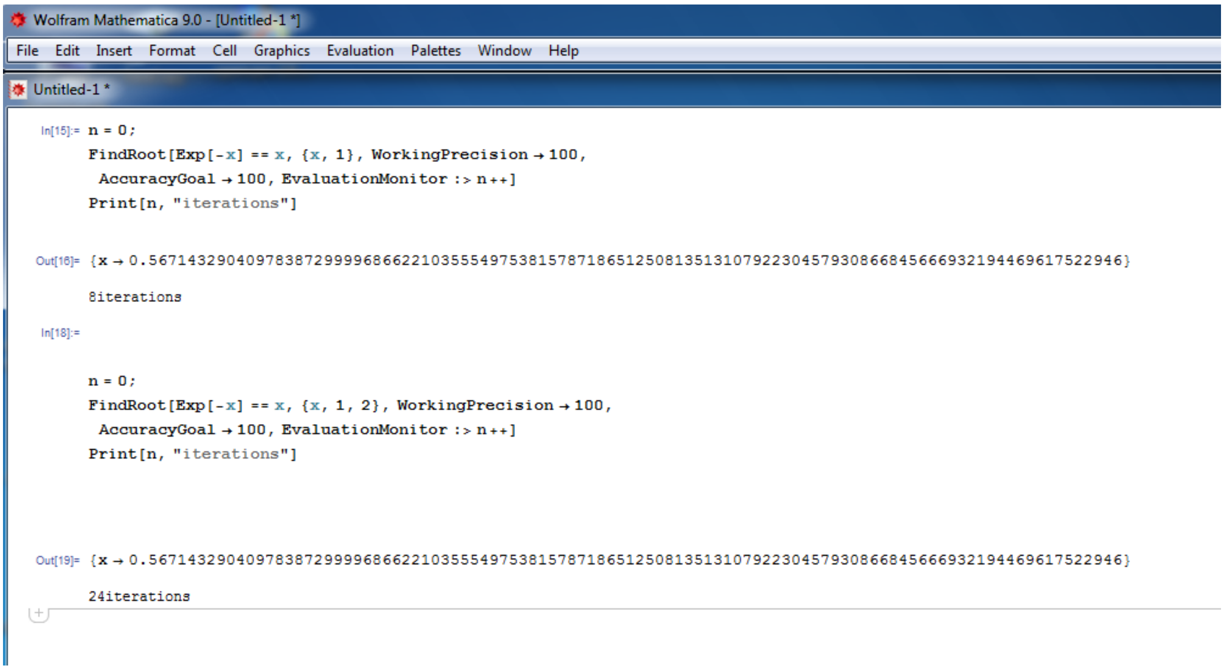
$$x_{n+1} = x_n - f\left(x_n\right)\frac{x_n - x_{n-1}}{f\left(x_n\right) - f\left(x_{n-1}\right)}$$

Στη συνέχεια επιλύεται η εξίσωση:

$$e^{-x} = x$$

με τη μέθοδο Newton-Raphson και τη μέθοδο της τέμνουσας με χρήση της εντολής FindRoot της Mathematica. Όταν δίνεται από το χρήστη ένα αρχικό σημείο η FindRoot χρησιμοποιεί τη μέθοδο Newton-Raphson ενώ όταν δίνονται δύο αρχικά σημεία η FindRoot χρησιμοποιεί τη μέθοδο της τέμνουσας.

Παρατηρούμε ότι η μέθοδος Newton-Raphson φτάνει στην επιθυμητή ακρίβεια σε μόλις 8 επαναλήψεις έναντι των 24 επαναλήψεων της μεθόδου της τέμνουσας.



Σχήμα 2.5 Σύγκριση των μεθόδων Newton-Raphson και Τέμνουσας στη Mathematica.

3. ΠΡΟΒΛΗΜΑΤΑ ΓΡΑΜΜΙΚΗΣ ΑΛΓΕΒΡΑΣ

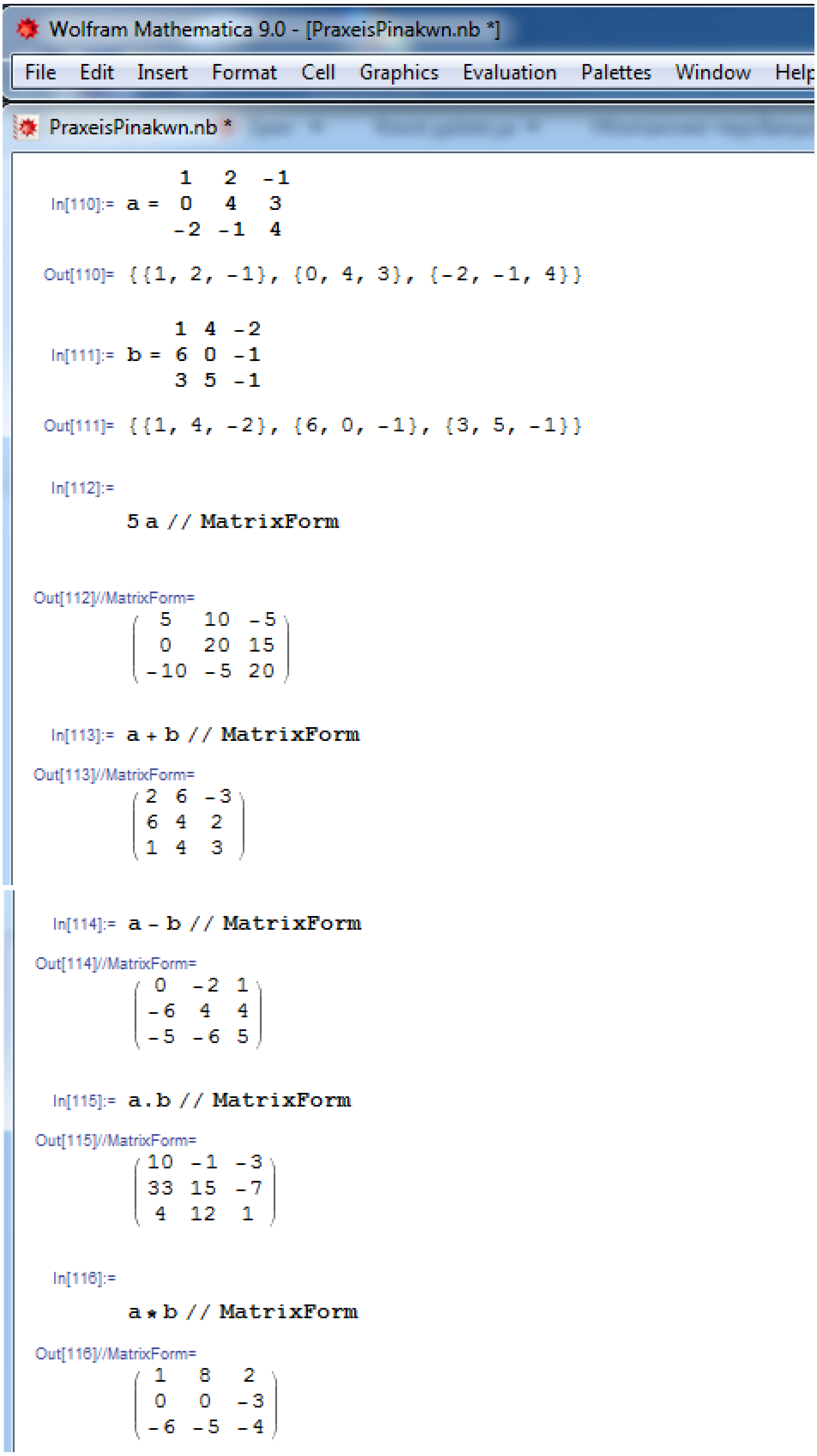
3.1 ΓΕΝΙΚΑ

Στο κεφάλαιο αυτό θα ασχοληθούμε με κλασσικά προβλήματα της γραμμικής άλγεβρας όπως:
Πράξεις με πίνακες
Επίλυση γραμμικών συστημάτων
Απαλοιφή Gauss-Jordan
Ιδιοτιμές και ιδιοδιανύσματα πίνακα

3.2 ΠΡΑΞΕΙΣ ΜΕ ΠΙΝΑΚΕΣ

Στη συνέχεια δείχνουμε πως γίνονται οι πράξεις μεταξύ πινάκων στη Mathematica. Παρουσιάζονται παραδείγματα πολλαπλασιασμού αριθμού με πίνακα καθώς και πρόσθεση, αφαίρεση και πολλαπλασιασμός πινάκων.

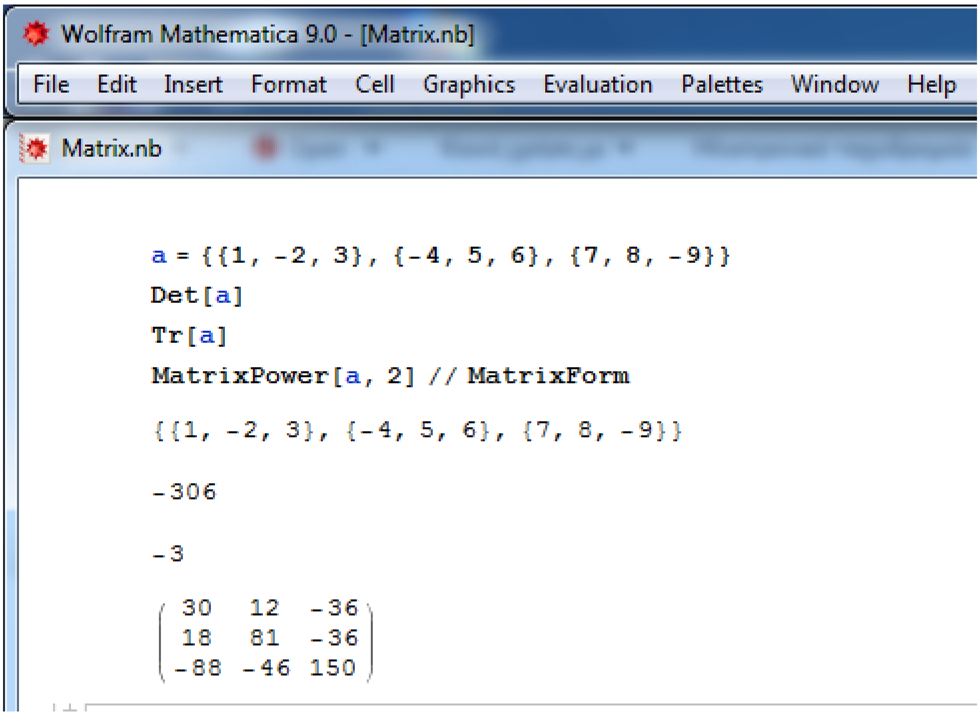
Πρέπει να τονιστεί ότι η εντολή a*b, όπου a, b πίνακες επιστρέφει έναν πίνακα του οποίου κάθε στοιχείο είναι το γινόμενο των αντίστοιχων στοιχείων των πινάκων a και b. Η σωστή εντολή για τον πολλαπλασιασμό πινάκων είναι a.b όπως φαίνεται και στο παράδειγμα που ακολουθεί. Στο παράδειγμα αυτό οι πίνακες a και b έχουν εισαχθεί μέσω της επιλογής table/matrix του μενού insert της Mathematica. Η εντολή MatrixForm χρησιμοποιείται ώστε η Mathematica να δώσει το αποτέλεσμα σε μορφή πίνακα.



Σχήμα 3.1 Απλές πράξεις μεταξύ πινάκων στη Mathematica.

Στο επόμενο παράδειγμα υπολογίζεται η ορίζουσα (determinant) και το ίχνος (trace) ενός πίνακα a. Επίσης ο ίδιος πίνακας υψώνεται στο τετράγωνο. Το ίχνος είναι το άθροισμα των διαγωνίων στοιχείων του πίνακα και έχει ιδιαίτερη σημασία καθώς παραμένει σταθερό σε κάποιους μετασχηματισμούς.

Είναι φανερό ότι αν στην εντολή MatrixPower βάλουμε άλλον αριθμό εκτός του 2 θα πάρουμε την αντίστοιχη δύναμη του πίνακα a.



Σχήμα 3.2 Υπολογισμός ορίζουσας, ίχνους και δύναμης πίνακα στη Mathematica.

3.3 ΑΝΑΣΤΡΟΦΟΣ ΚΑΙ ΑΝΤΙΣΤΡΟΦΟΣ ΠΙΝΑΚΑΣ

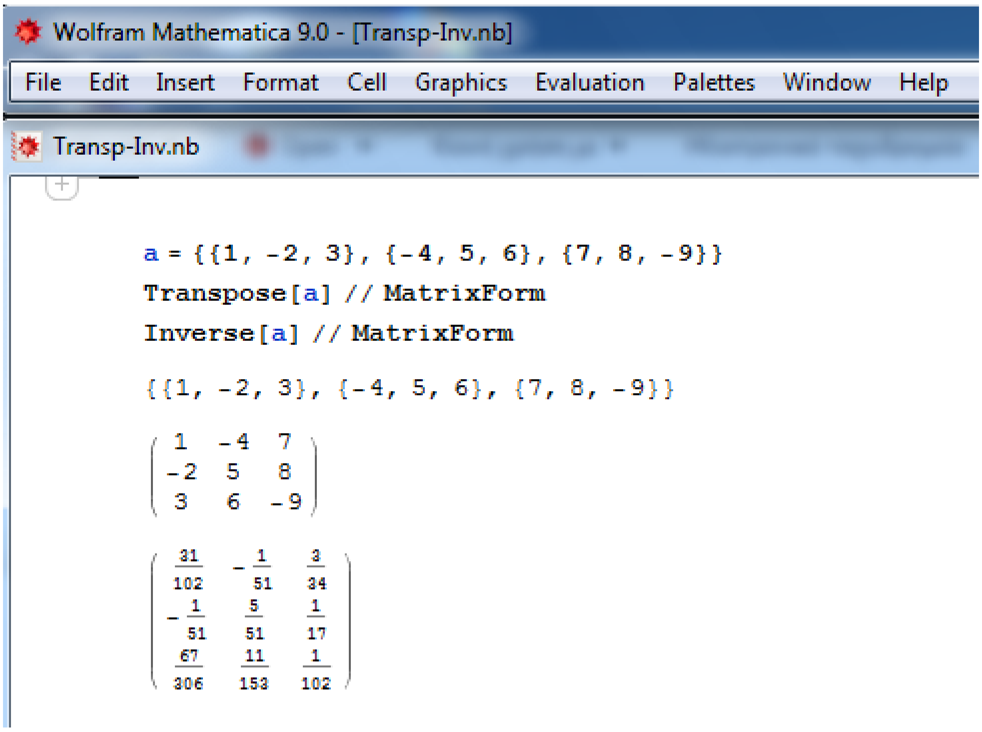
Ανάστροφος (transpose) ενός πίνακα A είναι ο πίνακας που προκύπτει αν κάνουμε τις γραμμές στήλες. Συμβολίζεται με A^T. Αντίστροφος (inverse) ενός τετραγωνικού πίνακα A είναι ο πίνακας A^{-1}, αν ισχύει:

$$A \cdot A^{-1} = A^{-1} \cdot A = I$$

όπου Ι είναι ο μοναδιαίος πίνακας. Ο αντίστροφος ενός πίνακα είναι μοναδικός.

Ένας τετραγωνικός πίνακας έχει αντίστροφο αν η ορίζουσά του είναι διάφορη του μηδενός. Σε αυτήν την περίπτωση ονομάζεται αντιστρέψιμος ή μη-ιδιάζων. Αν η ορίζουσα ενός τετραγωνικού πίνακα είναι ίση με μηδέν, τότε ο πίνακας ονομάζεται ιδιάζων και δεν έχει αντίστροφο. Στο παράδειγμα που ακολουθεί υπολογίζουμε τον ανάστροφο και τον αντίστροφο του πίνακα:

$$A = \begin{bmatrix} 1 & -2 & 3 \\ -4 & 5 & 6 \\ 7 & 8 & -9 \end{bmatrix}$$



Σχήμα 3.3 Υπολογισμός του ανάστροφου και του αντίστροφου πίνακα με τη Mathematica.

Στο παράδειγμα της προηγούμενης ενότητας υπολογίσαμε την ορίζουσα του πίνακα αυτού και βρήκαμε ότι είναι διάφορη του μηδενός. Συνεπώς, ο A είναι αντιστρέψιμος.

3.4 ΕΠΙΛΥΣΗ ΓΡΑΜΜΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Στην περίπτωση αυτή θέλουμε να επιλύσουμε ένα γραμμικό σύστημα n εξισώσεων με n αγνώστους, το οποίο σε μητρωική μορφή γράφεται ως:

$$A \cdot x = b$$

όπου:
A είναι ο $n \times n$ πίνακας των συντελεστών
x είναι το ζητούμενο διάνυσμα με n στοιχεία
b είναι το διάνυσμα των σταθερών με n στοιχεία

Ένας τρόπος επίλυσης είναι η μέθοδος του αντιστρόφου πίνακα. Σε αυτήν τη μέθοδο πολλαπλασιάζουμε από αριστερά και τα δύο μέλη της παραπάνω σχέσης με τον αντίστροφο του πίνακα A. Παίρνουμε:

$$A^{-1} \cdot A \cdot x = A^{-1} \cdot b \Rightarrow x = A^{-1} \cdot b$$

Απαιτείται φυσικά ο υπολογισμός του αντιστρόφου πίνακα A^{-1}.
Στο παράδειγμα που ακολουθεί λύνουμε με τη Mathematica το σύστημα:

x+y+2z=-1
2x-y+2z=-4
4x+y+4z=-2

Ο πίνακας A είναι:

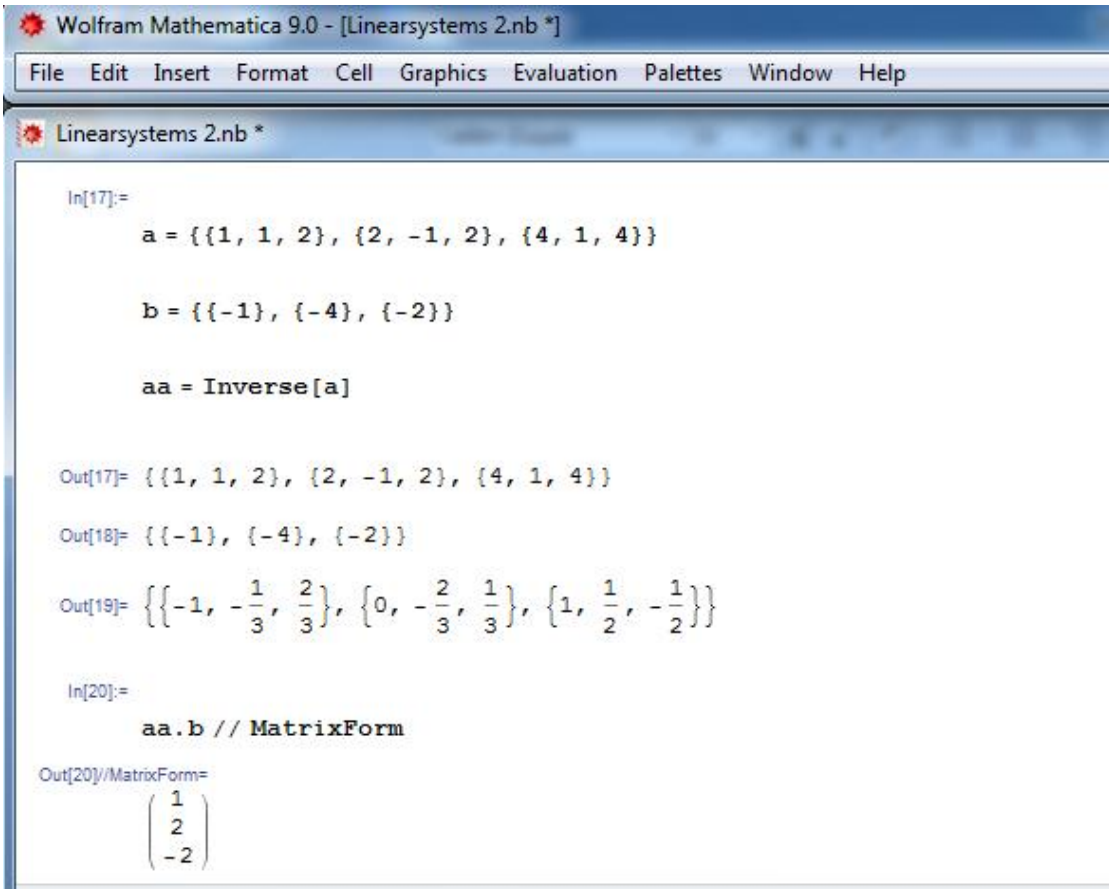
$$A = \begin{bmatrix} 1 & 1 & 2 \\ 2 & -1 & 2 \\ 4 & 1 & 4 \end{bmatrix}$$

και ο πίνακας b είναι:

$$b = \begin{bmatrix} -1 \\ -4 \\ -2 \end{bmatrix}$$

Αρχικά υπολογίζεται ο αντίστροφος του πίνακα A και στη συνέχεια η λύση:

$$x = \begin{bmatrix} 1 \\ 2 \\ -2 \end{bmatrix}$$



Σχήμα 3.4 Επίλυση γραμμικού συστήματος με τη μέθοδο του αντιστρόφου πίνακα στη Mathematica.

3.5 Η ΜΕΘΟΔΟΣ GAUSS-SEIDEL

Στις επαναληπτικές μεθόδους ξεκινάμε από μία αρχική προσέγγιση της λύσης, έστω $x^{(0)}$ και παράγουμε μια ακολουθία καλύτερων προσεγγίσεων η οποία συγκλίνει στη λύση σε άπειρες επαναλήψεις. Στην παρούσα εργασία θα εφαρμόσουμε τη μέθοδο Gauss-Seidel η οποία είναι μία πολύ δημοφιλής μέθοδος. Η μέθοδος αυτή βασίζεται στις παρακάτω επαναληπτικές σχέσεις:

$$x_i^{(k+1)} = x_i^{(k)} + \frac{1}{A_{ii}}\left(b_i - \sum_{j=1}^{i-1} A_{ij} x_j^{(k+1)} - \sum_{j=i}^{n} A_{ij} x_j^{(k)}\right) \quad \text{i=1,2,…,n}$$

Είναι φανερό από την παραπάνω σχέση ότι στη μέθοδο αυτή, ο υπολογισμός του $x_i^{(k+1)}$ χρειάζεται τις «νέες τιμές» $x_j^{(k+1)}$ για $j < i$ και τις «παλαιές τιμές» $x_j^{(k)}$ για $j \geq i$. Λέγοντας «νέες τιμές» εννοούμε τις τιμές που έχουν ήδη υπολογιστεί στην τρέχουσα επανάληψη ενώ ως «παλαιές τιμές» θεωρούνται αυτές που υπολογίστηκαν στην αμέσως προηγούμενη επανάληψη.

Στο παράδειγμα που ακολουθεί επιλύεται με τη μέθοδο Gauss-Seidel το γραμμικό σύστημα:

5x-2y+3z=-1
-3x+9y+z=-5
x-y-7z=15

Ως αρχική τιμή χρησιμοποιείται το διάνυσμα:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Οι επαναληπτικές σχέσεις που επιλύουν το συγκεκριμένο πρόβλημα είναι:

$$x^{(i+1)} = \frac{1}{5}\left(2y^{(i)} - 3z^{(i)}\right) - \frac{1}{5}$$

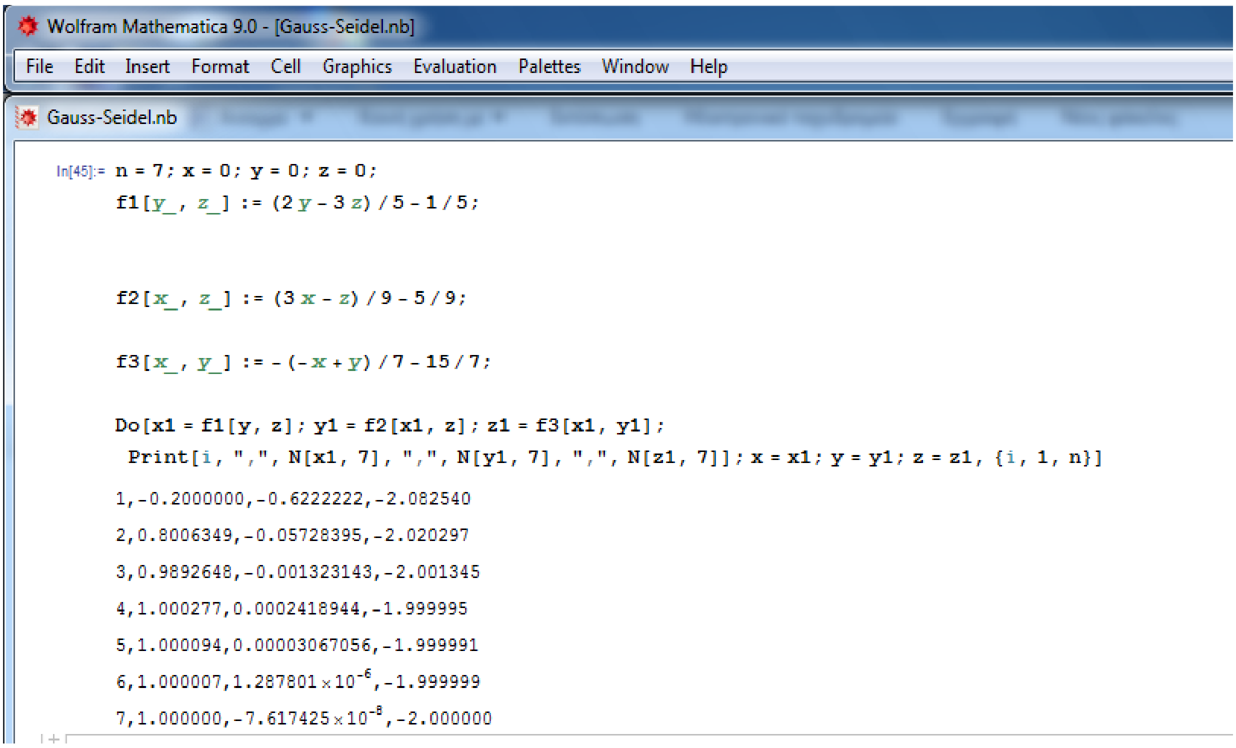
$$y^{(i+1)} = \frac{1}{9}\left(3x^{(i+1)} - z^{(i)}\right) - \frac{5}{9}$$

$$z^{(i+1)} = -\frac{1}{7}\left(-x^{(i+1)} + y^{(i+1)}\right) - \frac{15}{7}$$

Η λύση είναι:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -2 \end{bmatrix}$$

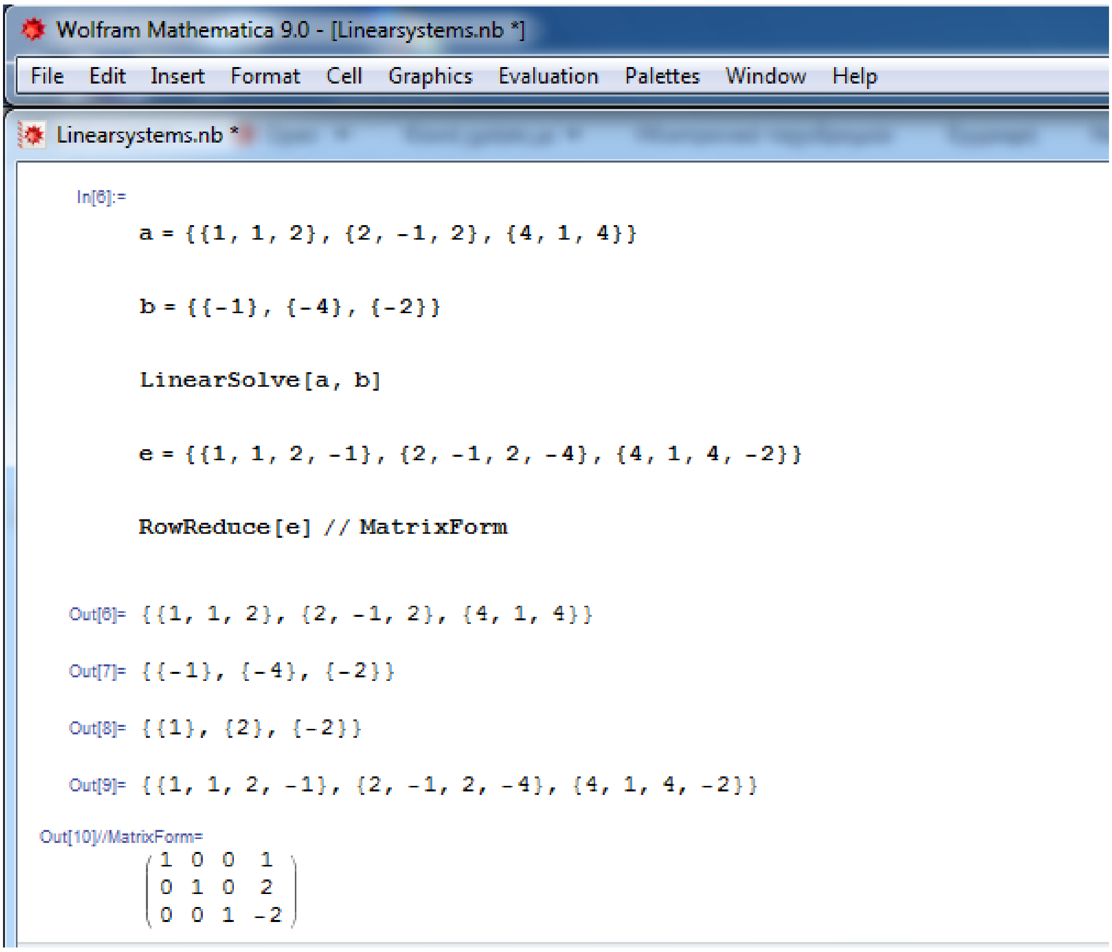
και όπως φαίνεται στο σχήμα που ακολουθεί προσεγγίζεται σε 7 επαναλήψεις.



Σχήμα 3.5 Εφαρμογή της μεθόδου Gauss-Seidel στη λύση γραμμικού συστήματος με τη Mathematica.

3.6 ΑΠΑΛΟΙΦΗ GAUSS-JORDAN

Η μέθοδος Gauss-Jordan χρησιμοποιείται για την επίλυση γραμμικών συστημάτων. Εκτελώντας κατάλληλους μετασχηματισμούς στον λεγόμενο επαυξημένο πίνακα του συστήματος, που αποτελείται από τον πίνακα A και το διάνυσμα b, ώστε ο πίνακας A να καταστεί μοναδιαίος, τότε τα στοιχεία του μετασχηματισμένου διανύσματος b είναι η ζητούμενη λύση του συστήματος.



Σχήμα 3.6 Επίλυση γραμμικού συστήματος με την εντολή LinearSolve και με τη μέθοδο Gauss-Jordan στη Mathematica.

Μπορεί όμως να χρησιμοποιηθεί και για την εύρεση του αντιστρόφου ενός τετραγωνικού πίνακα. Αν εκτελέσουμε τους παραπάνω μετασχηματισμούς στον πίνακα A και συγχρόνως στον μοναδιαίο πίνακα $n \times n$, I_n, τότε ο I_n θα μετασχηματιστεί στον αντίστροφο πίνακα του A, δηλαδή τον A^{-1}.

Στο παράδειγμα του σχήματος 3.6 λύνουμε το γραμμικό σύστημα της ενότητας 3.4, δηλαδή το:

x+y+2z=-1
2x-y+2z=-4
4x+y+4z=-2

Αρχικά το σύστημα λύνεται με την εντολή LinearSolve της Mathematica και στη συνέχεια με την εντολή RowReduce, η οποία υλοποιεί τη μέθοδο Gauss-Jordan. Ο επαυξημένος πίνακας του συστήματος είναι προφανώς ο πίνακας:

$$E = \begin{bmatrix} 1 & 1 & 2 & -1 \\ 2 & -1 & 2 & -4 \\ 4 & 1 & 4 & -2 \end{bmatrix}$$

3.7 ΙΔΙΟΤΙΜΕΣ ΚΑΙ ΙΔΙΟΔΙΑΝΥΣΜΑΤΑ ΠΙΝΑΚΑ

Οι ιδιοτιμές λ_i ενός πίνακα A προκύπτουν από τη λύση της λεγόμενης χαρακτηριστικής εξίσωσης:

$$\det(A - \lambda I) = 0$$

Στη συνέχεια λύνοντας το σύστημα:

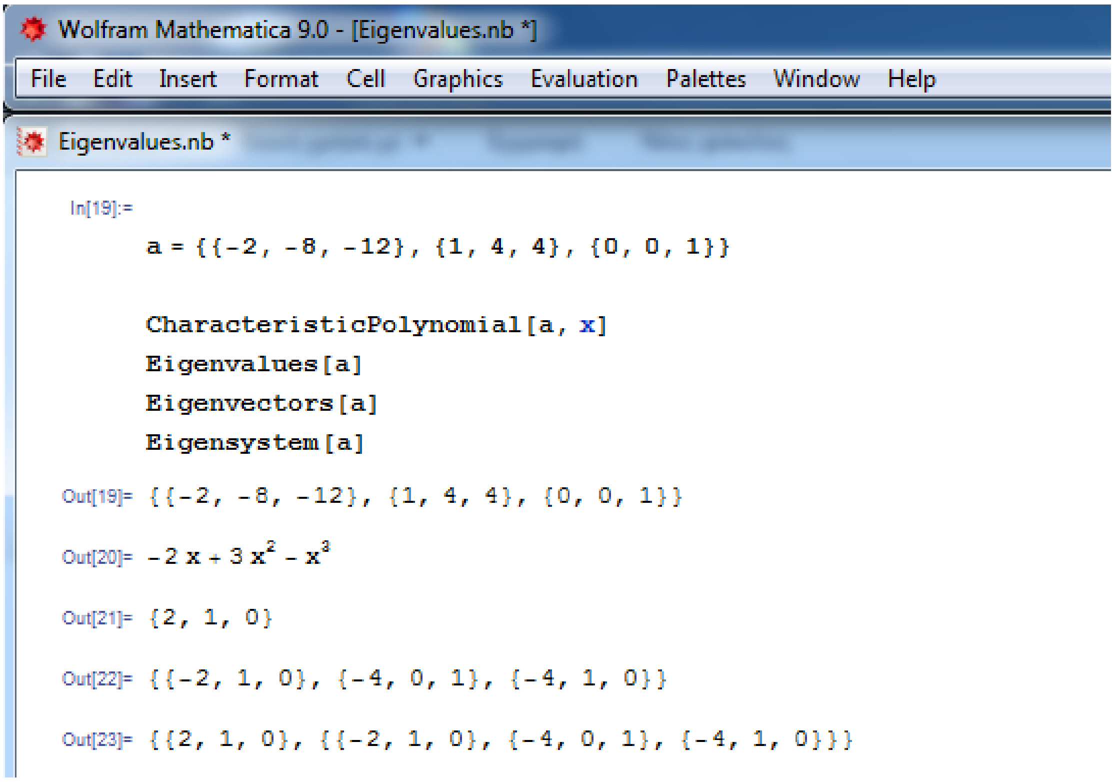
$$(A - \lambda_i I) \cdot x_i = 0$$

για κάθε ιδιοτιμή λ_i και βρίσκουμε το ιδιοδιάνυσμα x_i που της αντιστοιχεί.

Σε κάθε ιδιοτιμή μπορεί να αντιστοιχούν και περισσότερα από ένα ιδιοδιανύσματα ανάλογα με την πολλαπλότητα της λ_i ως ρίζας της χαρακτηριστικής εξίσωσης.

Στο παράδειγμα που ακολουθεί βρίσκουμε με τις κατάλληλες εντολές της Mathematica το χαρακτηριστικό πολυώνυμο, τις ιδιοτιμές και τα ιδιοδιανύσματα του πίνακα:

$$A = \begin{bmatrix} -2 & -8 & -12 \\ 1 & 4 & 4 \\ 0 & 0 & 1 \end{bmatrix}$$



Σχήμα 3.7 Εύρεση των ιδιοτιμών και των ιδιοδιανυσμάτων ενός πίνακα με τη Mathematica.

Οι ιδιοτιμές βρέθηκαν ότι είναι: 2,1,0. Συνεπώς ο πίνακας Α είναι όμοιος με τον διαγώνιο πίνακα:

$$\Delta = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Κατά τον μετασχηματισμό ομοιότητας το ίχνος του πίνακα παραμένει σταθερό. Πράγματι παρατηρούμε ότι:

Tr(A)=Tr(Δ) αφού -2+4+1=2+1+0

Επίσης παρατηρούμε ότι ο πίνακας Α έχει μηδενική ιδιοτιμή. Αυτό σημαίνει ότι ο πίνακας Α δεν είναι αντιστρέψιμος αφού από τη χαρακτηριστική εξίσωση, για λ=0 παίρνουμε:

$$\det(A) = 0$$

4. ΠΑΡΑΓΩΓΙΣΗ ΚΑΙ ΟΛΟΚΛΗΡΩΣΗ ΣΥΝΑΡΤΗΣΕΩΝ

4.1 ΓΕΝΙΚΑ

Στο κεφάλαιο αυτό θα ασχοληθούμε με μερικά πολύ σημαντικά προβλήματα της ανάλυσης όπως είναι η παραγώγιση και η ολοκλήρωση συναρτήσεων. Το Mathematica προσφέρει τη δυνατότητα αναλυτικών υπολογισμών της παραγώγου και του αόριστου και ορισμένου ολοκληρώματος. Επίσης υπάρχει και η δυνατότητα εφαρμογής αριθμητικών μεθόδων.

4.2 ΠΑΡΑΓΩΓΟΣ ΣΥΝΑΡΤΗΣΗΣ

Η τιμή $f'(x_0)$ της παραγώγου μιας συνάρτησης f(x) σε ένα σημείο x_0, όπως είναι γνωστό, υπολογίζεται από το παρακάτω όριο:

$$f'(x_0) = \lim_{h \to 0} \frac{f(x_0 + h) - f(x_0)}{h}$$

Η συνάρτηση f(x) είναι παραγώγισιμη στο x_0 εφόσον το παραπάνω όριο υπάρχει. Στην πράξη αντί του παραπάνω ορισμού, χρησιμοποιούμε τους γνωστούς κανόνες παραγώγισης για τον αναλυτικό υπολογισμό της παραγώγου μιας συνάρτησης σε κάθε σημείο του πεδίου ορισμού της.

Στην αριθμητική ανάλυση είναι πολύ χρήσιμο να μπορούμε να υπολογίσουμε προσεγγιστικά τις παραγώγους μιας συνάρτησης όταν γνωρίζουμε τις τιμές της συνάρτησης στα σημεία: …x_{-1}, x_0, x_1, …
Για ισαπέχοντα σημεία, με απόσταση έστω h, έχουμε τους παρακάτω τύπους για να υπολογίσουμε την τιμή της παραγώγου στο x_0:

Προς τα εμπρός διαφορές:

$$f'(x_0) \approx \frac{f(x_1) - f(x_0)}{h}$$

Προς τα πίσω διαφορές:

$$f'(x_0) \approx \frac{f(x_0) - f(x_{-1})}{h}$$

Κεντρικές διαφορές:

$$f'(x_0) \approx \frac{f(x_1) - f(x_{-1})}{2h}$$

Στη συνέχεια δίνονται κάποια παραδείγματα υπολογισμού παραγώγων στη Mathematica.

Αρχικά υπολογίζεται η πρώτη και η δεύτερη παράγωγος της συνάρτησης:

$$f(x) = 5x^3 + 3x^2 - 4x + 1$$

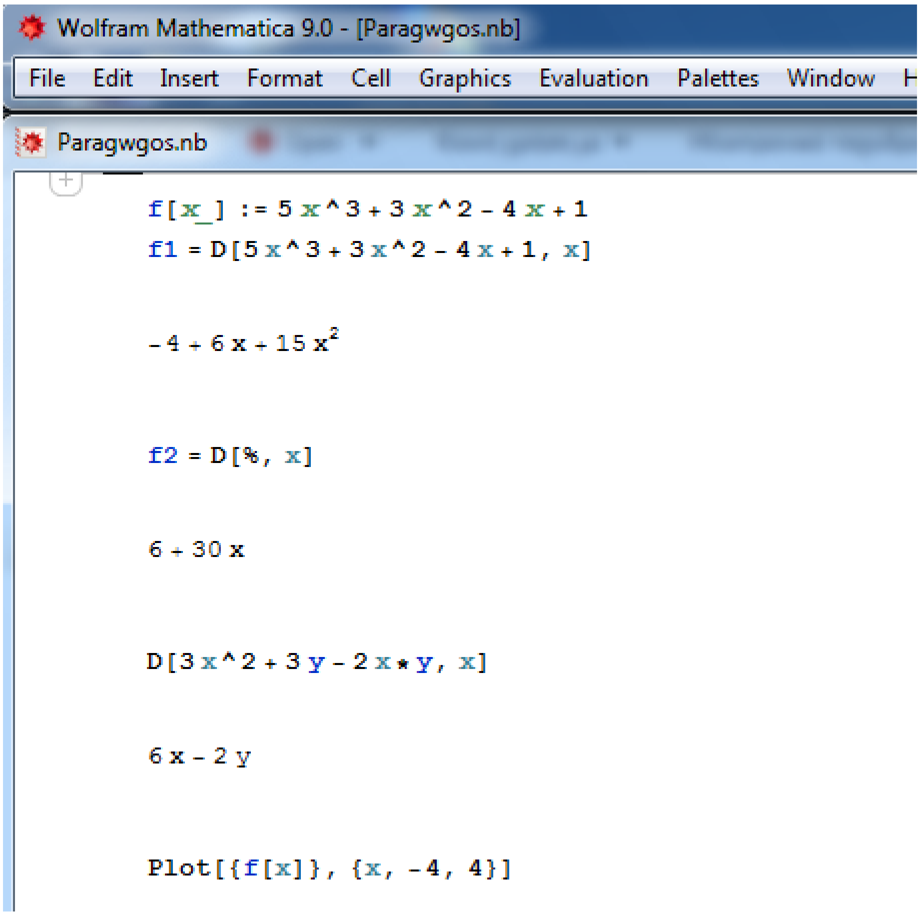
Στην εντολή D[f[x],x] με την οποία εκτελείται η παραγώγιση πρέπει να δηλώσουμε ότι η παραγώγιση γίνεται ως προς τη μεταβλητή x.

Με την εντολή Plot γίνονται οι γραφικές παραστάσεις της συνάρτησης f(x) καθώς και της πρώτης και της δεύτερης παραγώγου της στο διάστημα:

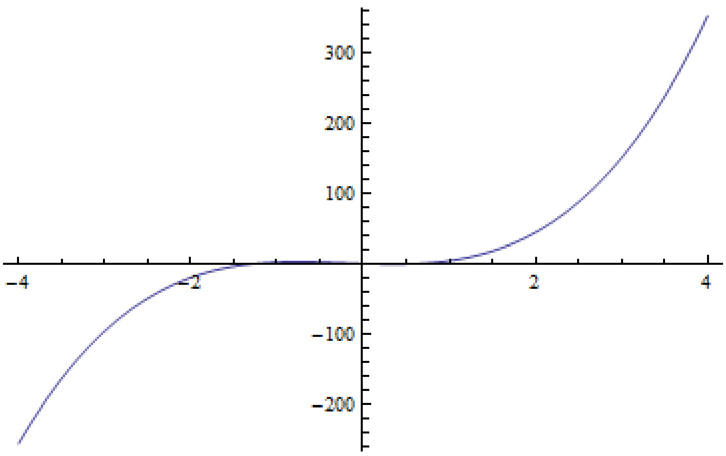
$$-4 \le x \le 4$$

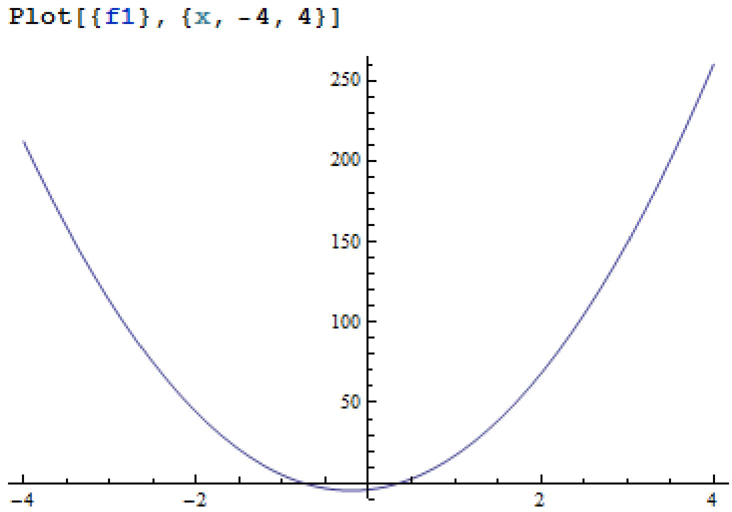
Στη συνέχεια υπολογίζεται η μερική παράγωγος ως προς x της συνάρτησης δύο μεταβλητών:

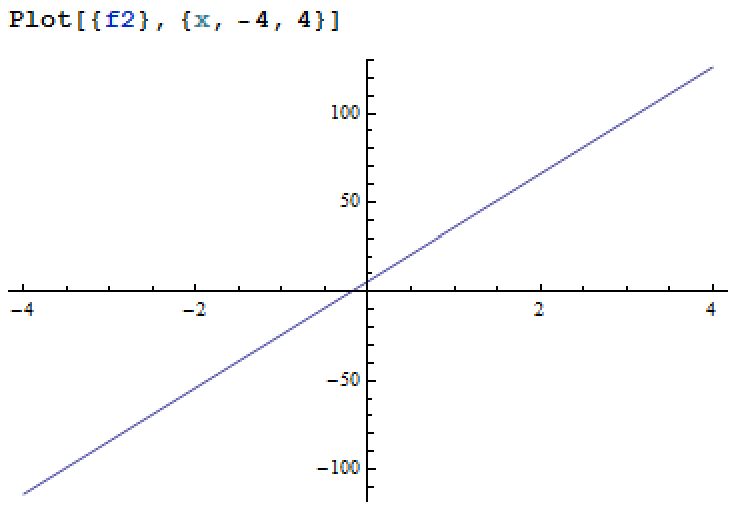
$$f(x,y)=3x^2+3y-2xy$$



Σχήμα 4.1 Υπολογισμοί παραγώγων στη Mathematica.







Σχήμα 4.2 Γραφική παράσταση της συνάρτησης του προηγούμενου σχήματος και των παραγώγων της.

4.3 ΑΟΡΙΣΤΟ ΚΑΙ ΟΡΙΣΜΕΝΟ ΟΛΟΚΛΗΡΩΜΑ

Ένα από τα βασικά προβλήματα στα μαθηματικά είναι ο υπολογισμός του ολοκληρώματος μιας συνάρτησης πραγματικής μεταβλητής. Η Mathematica δίνει τη δυνατότητα αναλυτικού υπολογισμού του αόριστου ολοκληρώματος (παράγουσα συνάρτηση), καθώς επίσης και του ακριβούς υπολογισμού του ορισμένου ολοκληρώματος. Ακολουθούν παραδείγματα.

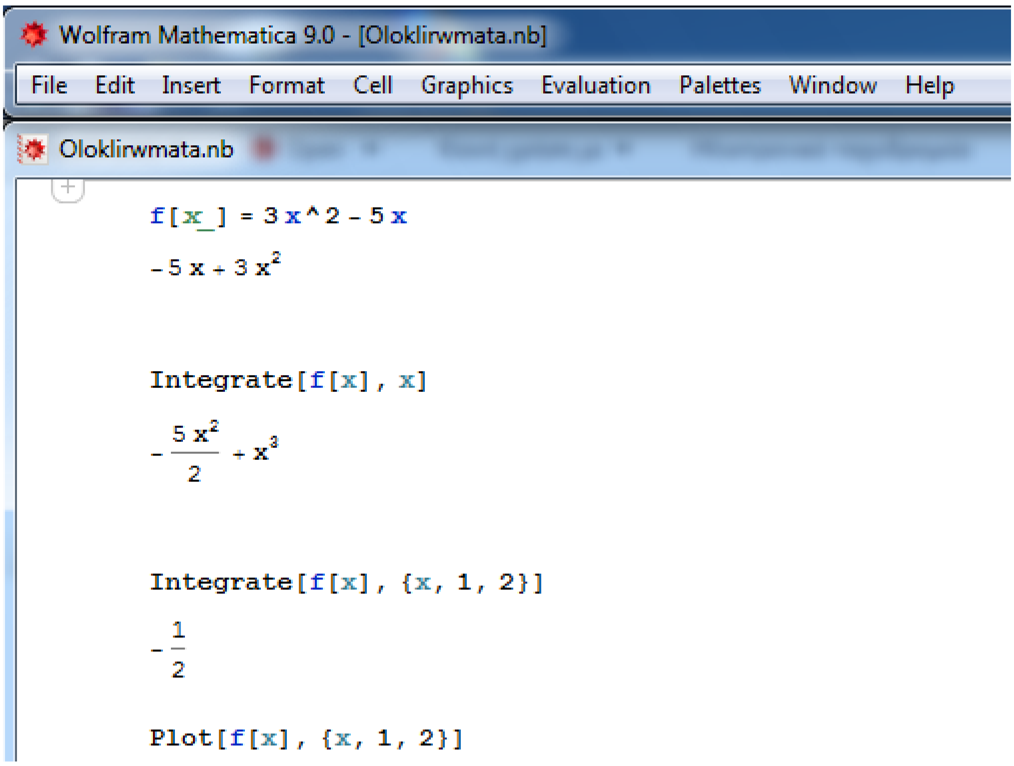
Στο σχήμα 4.3 φαίνεται και ο υπολογισμός του μήκους της καμπύλης:

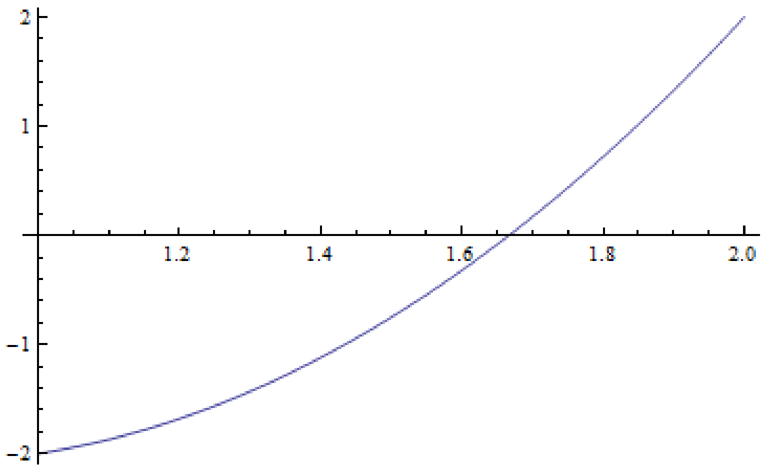
$$y = 3x^2 - 5x$$

από το x=1 έως το x=2, με χρήση του τύπου:

$$L = \int_a^b \sqrt{1 + [y'(x)]^2}\, dx$$

Στην εντολή Integrate δίνουμε την ολοκληρωτέα συνάρτηση, τη μεταβλητή ως προς την οποία θα γίνει η ολοκλήρωση και αν το ολοκλήρωμα είναι ορισμένο δίνουμε και τα όρια ολοκλήρωσης.






```
g = D[f[x], x]
-5 + 6 x
Integrate[Sqrt[1 + g^2], {x, 1, 2}]
```

$$\frac{1}{12}\left(34\sqrt{2} - \mathrm{ArcSinh}[1] + \mathrm{ArcSinh}[7]\right)$$

Σχήμα 4.3 Παραδείγματα υπολογισμού ολοκληρωμάτων στη Mathematica.

Για τον υπολογισμό ολοκληρωμάτων με συγκεκριμένα όρια έχουν αναπτυχθεί διάφορες αριθμητικές μέθοδοι. Όλες εκφράζουν το ζητούμενο ολοκλήρωμα ως άθροισμα των τιμών της συνάρτησης σε n συγκεκριμένα σημεία x_i στο διάστημα ολοκλήρωσης, πολλαπλασιασμένων με κατάλληλες σταθερές w_i:

$$\int_a^b f(x)dx \approx \sum_{i=1}^{n} w_i f(x_i)$$

Αν τα σημεία είναι ισαπέχοντα, οι μέθοδοι παράγουν τύπους στη γενική κατηγορία των τύπων Newton–Cotes. Σε αυτήν την κατηγορία ανήκουν οι μέθοδοι τραπεζίου και Simpson. Και στις δύο αυτές μεθόδους χωρίζουμε το διάστημα ολοκλήρωσης σε υποδιαστήματα και σε κάθε υποδιάστημα υπολογίζουμε το ολοκλήρωμα της συνάρτησης με χρήση του αντίστοιχου τύπου. Η ολοκληρωτέα συνάρτηση σε κάθε υποδιάστημα προσεγγίζεται στον κανόνα του τραπεζίου με γραμμική συνάρτηση και στον κανόνα του Simpson με παραβολή.

Τύπος Τραπεζίου:

$$\int_{x_0}^{x_1} f(x)dx \approx \frac{x_1 - x_0}{2}\left[f(x_0) + f(x_1)\right]$$

Τύπος Simpson:

$$\int_{x_0}^{x_2} f(x)dx \approx \frac{h}{3}\left[f(x_0) + 4f(x_1) + f(x_2)\right]$$

όπου: $h = x_1 - x_0 = x_2 - x_1$

Στο προηγούμενο παράδειγμα υπολογίσαμε το ολοκλήρωμα της συνάρτησης:

$$f(x) = 3x^2 - 5x$$

στο διάστημα [1,2]. Η τιμή που βρέθηκε ήταν $-\frac{1}{2}$.

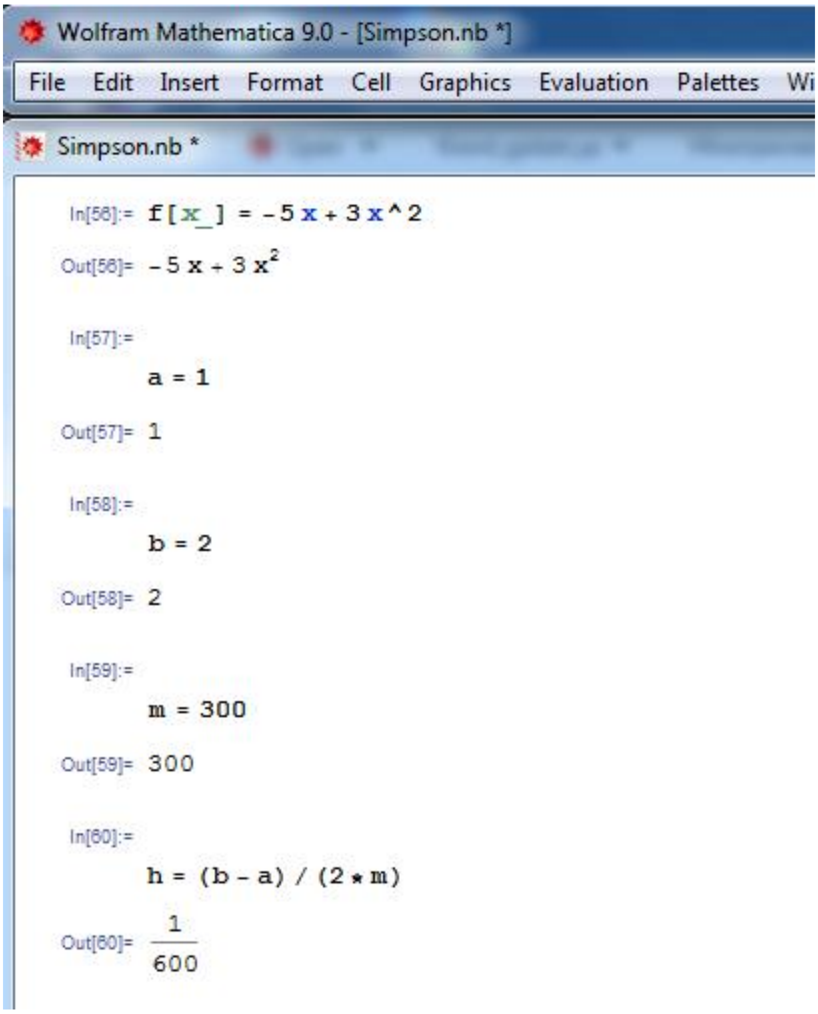
Στη συνέχεια χρησιμοποιήσαμε τον παρακάτω κώδικα (σχήμα 4.4) για την εφαρμογή του κανόνα του Simpson στον υπολογισμό του ίδιου ολοκληρώματος. Χρησιμοποιήσαμε διάφορες τιμές του πλήθους των υποδιαστημάτων στα οποία χωρίζεται το διάστημα ολοκλήρωσης [1,2], ώστε να δούμε την εξάρτηση του αποτελέσματος από αυτήν την παράμετρο. Το πλήθος των υποδιαστημάτων, στον παρακάτω κώδικα συμβολίζεται με το γράμμα m.

Τα αποτελέσματά μας συνοψίζονται στον πίνακα 4.1 που ακολουθεί.

Πίνακας 4.1 Εξάρτηση του αποτελέσματος από το πλήθος των διαμερίσεων στον κανόνα του Simpson

m	Τιμή Ολοκληρώματος
50	-0.52574
100	-0.51310
150	-0.50879
200	-0.50661
250	-0.50530
300	-0.50442

Είναι φανερό από τα δεδομένα του πίνακα 4.1 ότι καθώς το πλήθος των διαμερίσεων αυξάνεται η τιμή του ολοκληρώματος που υπολογίζεται αριθμητικά μέσω του κανόνα του Simpson συγκλίνει στην ακριβή τιμή που είναι -0.5.



```
In[61]:=
        SumEven = 0
Out[61]= 0

In[62]:=
        Do[SumEven = SumEven + f[a + h * 2 * k], {k, m - 1}]

In[63]:=
        SumOdd = 0
Out[63]= 0

In[64]:=
        Do[SumOdd = SumOdd + f[a + h * (2 k - 1)], {k, m - 1}]

In[65]:=
        R = (h / 3) * (f[a] + f[b] + 2 * SumEven + 4 * SumOdd)
```

Out[65]= $-\frac{27\,238\,601}{54\,000\,000}$

```
In[66]:=
        Print[N[R, 5]]
        -0.50442
```

Σχήμα 4.4 Εφαρμογή του κανόνα του Simpson στο ολοκλήρωμα του προηγούμενου παραδείγματος.

5. ΔΙΑΦΟΡΙΚΕΣ ΕΞΙΣΩΣΕΙΣ

5.1 ΓΕΝΙΚΑ

Στο παρόν κεφάλαιο θα ασχοληθούμε με την επίλυση Διαφορικών Εξισώσεων και Συστημάτων Διαφορικών Εξισώσεων με τη Mathematica.

Διαφορική ονομάζεται η εξίσωση στην οποία εμφανίζεται μία άγνωστη συνάρτηση και οι παράγωγοι αυτής. Αν επιπλέον είναι γνωστές κάποιες τιμές της άγνωστης συνάρτησης και των παραγώγων της τότε έχουμε ένα πρόβλημα αρχικών τιμών.

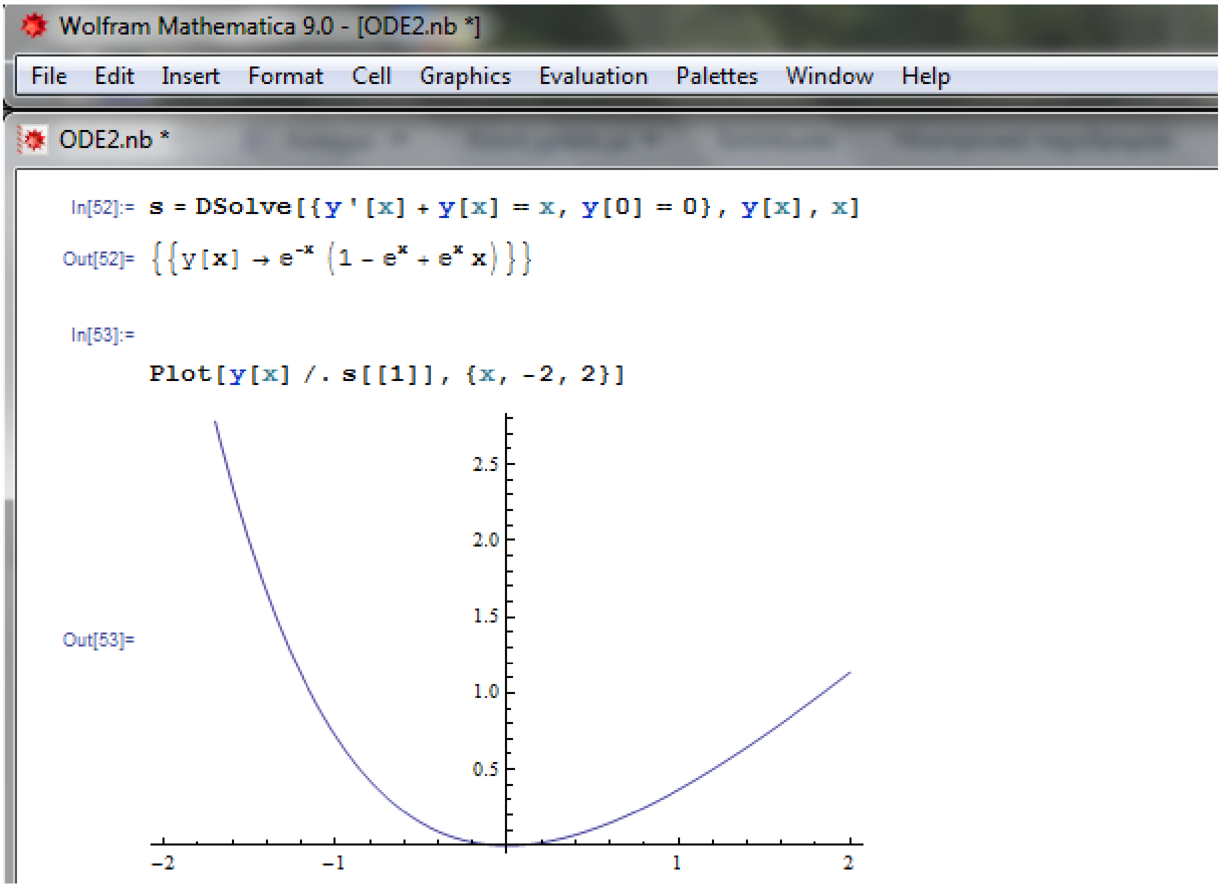
5.2 ΕΠΙΛΥΣΗ ΔΙΑΦΟΡΙΚΩΝ ΕΞΙΣΩΣΕΩΝ ΜΕ ΤΗΝ ΕΝΤΟΛΗ DSOLVE

Στη συνέχεια λύνεται με την εντολή DSolve της Mathematica το πρόβλημα αρχικών τιμών:

$$y'(x) + y(x) = x \qquad y(0) = 0$$

Στην εντολή DSolve δίνουμε τη διαφορική εξίσωση, την αρχική συνθήκη και δηλώνουμε επίσης τη συνάρτηση και την ανεξάρτητη μεταβλητή.
Η συνάρτηση που ικανοποιεί τη διαφορική εξίσωση και την αρχική συνθήκη, δηλαδή η λύση του προβλήματος αρχικών τιμών είναι:

$$y(x) = e^{-x}\left(1 - e^{x} - e^{x}x\right)$$



Σχήμα 5.1 Επίλυση διαφορικής εξίσωσης με την εντολή DSolve.

5.3 ΣΥΣΤΗΜΑΤΑ ΔΙΑΦΟΡΙΚΩΝ ΕΞΙΣΩΣΕΩΝ

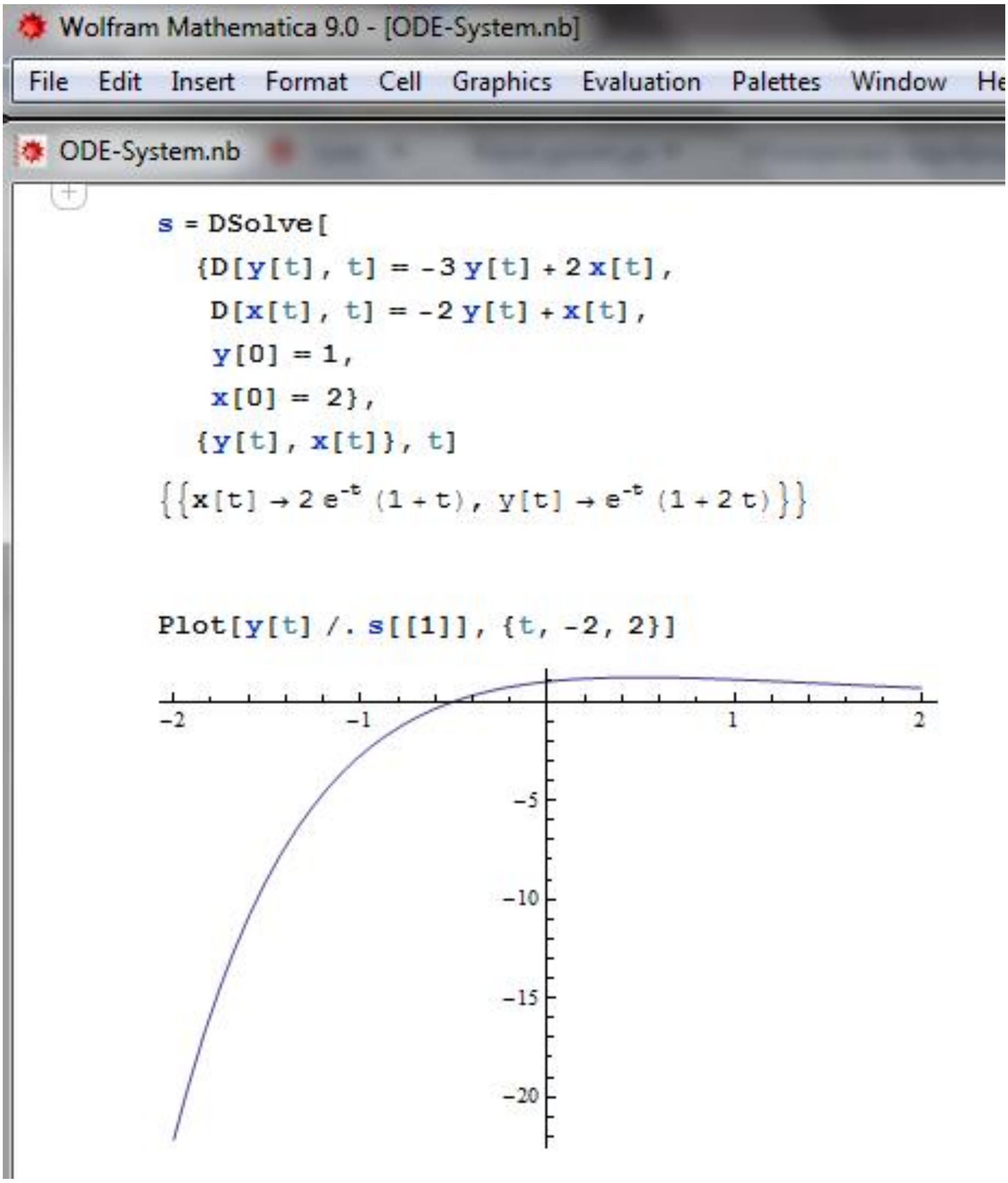
Πολύ συχνά παρουσιάζεται η ανάγκη να λύσουμε ένα σύστημα διαφορικών εξισώσεων. Για παράδειγμα το σύστημα:

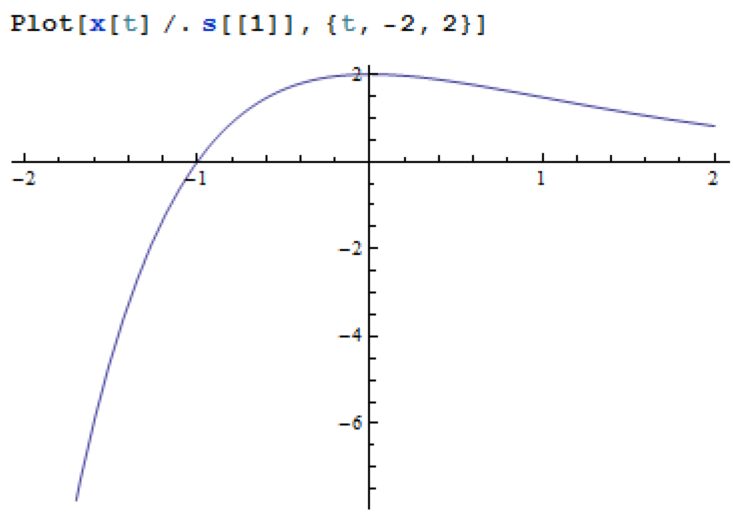
$$y'(t) = -3y(t) + 2x(t)$$

$$x'(t) = -2y(t) + x(t)$$

όπου οι άγνωστες συναρτήσεις y και x εξαρτώνται από την ίδια ανεξάρτητη μεταβλητή t.

Η εντολή DSolve μπορεί να χρησιμοποιηθεί για τη λύση συστημάτων διαφορικών εξισώσεων όπως φαίνεται στο σχήμα που ακολουθεί.





Σχήμα 5.2 Επίλυση συστήματος διαφορικών εξισώσεων με την εντολή DSolve.

5.4 ΑΡΙΘΜΗΤΙΚΗ ΕΠΙΛΥΣΗ ΔΙΑΦΟΡΙΚΩΝ ΕΞΙΣΩΣΕΩΝ

Στις περιπτώσεις που μία διαφορική εξίσωση δεν μπορεί να λυθεί αναλυτικά, δηλαδή δεν μπορεί να υπολογιστεί ο τύπος που δίνει την άγνωστη συνάρτηση καταφεύγουμε στις αριθμητικές μεθόδους. Υπολογίζουμε δηλαδή τις τιμές της άγνωστης συνάρτησης σημείο προς σημείο. Λόγω της μεγάλης σημασίας που έχουν οι διαφορικές εξισώσεις, έχουν αναπτυχθεί πάρα πολλές αριθμητικές μέθοδοι. Στην συνέχεια θα ασχοληθούμε με δύο από αυτές τις μεθόδους. Θα αναφερθούμε στη μέθοδο του Euler, η οποία είναι μία από τις πρώτες μεθόδους που αναπτύχθηκαν αλλά σήμερα δεν χρησιμοποιείται καθόλου. Θα περιγράψουμε με περισσότερη λεπτομέρεια τη μέθοδο Runge-Kutta 4^{ης} τάξεως, η οποία θεωρείται μία από τις καλύτερες μεθόδους και χρησιμοποιείται ευρύτατα.

5.4.1 Μέθοδος Euler

Αν και η μέθοδος του Euler δεν χρησιμοποιείται σήμερα συχνά στην πράξη, παραμένει σημαντική γιατί αποτελεί τη βάση για την ανάπτυξη άλλων ακριβέστερων μεθόδων.

Έστω ότι έχουμε να επιλύσουμε τα πρόβλημα αρχικών τιμών:

$$y' = f(x, y), \quad y(x_0) = y_0$$

Επομένως, η ευθεία ε που διέρχεται από το σημείο (x_0,y_0) και έχει κλίση:

$$\kappa = y'(x_0) = f(x_0, y_0)$$

εφάπτεται στην καμπύλη της άγνωστης συνάρτησης $y(x)$, στο σημείο αυτό. Άρα, για μικρές μεταβολές της τιμής του x, έστω Δx, μπορούμε να θεωρήσουμε πως η καμπύλη της $y(x)$ και η ευθεία ε πρακτικά ταυτίζονται.

Άρα έχουμε:

$$\Delta y = \kappa \Delta x$$

και

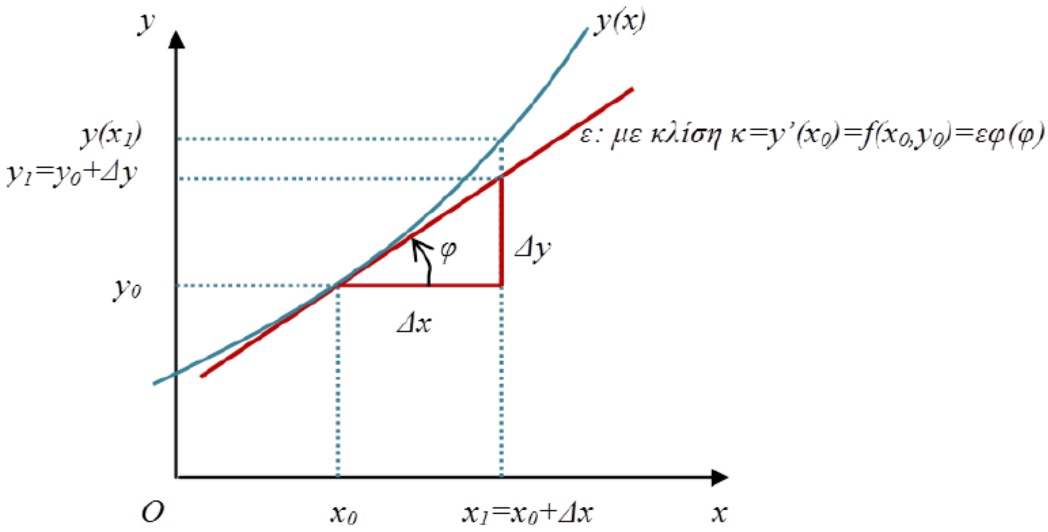
$$y(x_1) = y(x_0 + \Delta x) = y(x_0) + \Delta y = y(x_0) + f(x_0, y_0)\Delta x$$

Θέτοντας το βήμα $\Delta x = h$, παίρνουμε:

$$y(x_1) = y(x_0) + hf(x_0, y_0)$$

Ο παραπάνω τύπος του Euler μας επιτρέπει τον υπολογισμό των τιμών της συνάρτησης y(x) που είναι η λύση της διαφορικής εξίσωσης.

Στο παρακάτω σχήμα 5.3 δίνεται η γραφική αναπαράσταση της μεθόδου του Euler



Σχήμα 5.3 Γραφική αναπαράσταση της μεθόδου Euler [8].

5.4.2 Μέθοδοι Runge-Kutta

Η μέθοδος του Euler ελέγχει την πρώτη παράγωγο, δηλαδή την κλίση, της άγνωστης συνάρτησης y(x) και υπολογίζει την επόμενη τιμή της, με τη βοήθεια αυτής της κλίσης. Αντίθετα, η μέθοδος των Runge–Kutta δοκιμάζει την κλίση της y(x), μέσω της διαφορικής εξίσωσης, σε περισσότερα προσεκτικά επιλεγμένα σημεία και αποφασίζει την πλέον κατάλληλη κλίση, μέσω της οποίας υπολογίζει την επόμενη τιμή.

Πρόκειται για μια οικογένεια μεθόδων. Η πιο συνηθισμένη επιλογή στην πράξη είναι η μέθοδος Runge-Kutta 4ης τάξεως, η οποία για την επίλυση του προβλήματος αρχικών τιμών:

$$y' = f(x, y), \quad y(x_0) = y_0$$

βασίζεται στις παρακάτω σχέσεις:

$$y(x_1) = y(x_0 + h) = y(x_0) + h\frac{K_1 + 2K_2 + 2K_3 + K_4}{6}$$

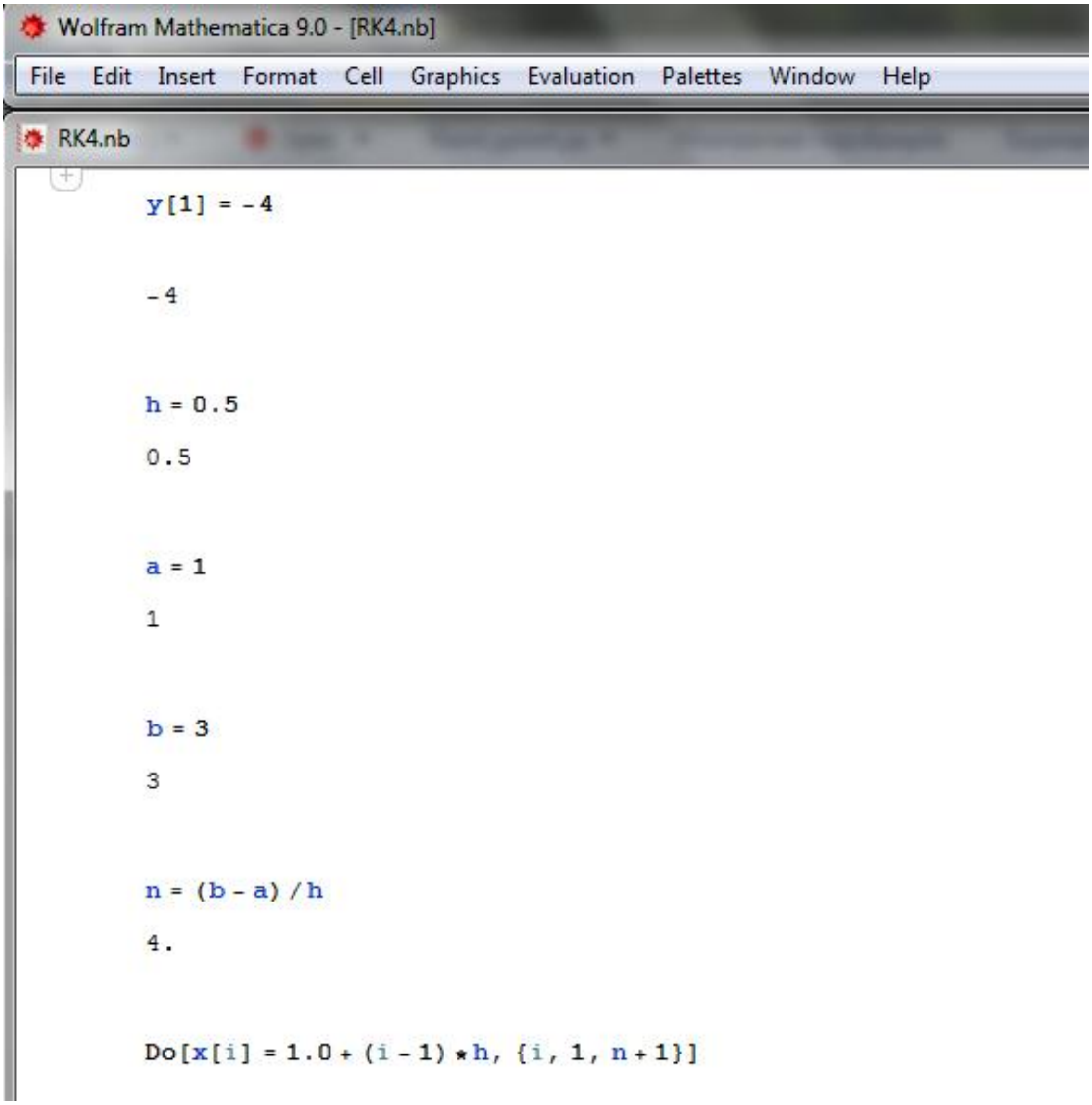
όπου:

$$K_1 = f(x_0, y_0)$$

$$K_2 = f\left(x_0 + \frac{h}{2}, y_0 + \frac{h}{2}K_1\right)$$

$$K_3 = f\left(x_0 + \frac{h}{2}, y_0 + \frac{h}{2}K_2\right)$$

$$K_4 = f(x_0 + h, y_0 + hK_3)$$



```
In[7]:=
      f[x_, y_] = y - x^2
Out[7]= -x^2 + y

In[8]:=
      Do[{S1 = f[x[i], y[i]], S2 = f[x[i] + h / 2, y[i] + h * S1 / 2],
        S3 = f[x[i] + h / 2, y[i] + h * S2 / 2], S4 = f[x[i] + h, y[i] + h * S3],
        y[i + 1] = y[i] + (S1 + 2 * S2 + 2 * S3 + S4) * h / 6}, {i, 1, n}]

In[9]:= Do[Print[x[i], "    ", y[i]], {i, 1, n + 1}]

      1.    -4
      1.5    -7.58659
      2.    -14.4578
      2.5    -27.0679
      3.    -49.4621
```

Σχήμα 5.4 Εφαρμογή της μεθόδου Runge-Kutta 4^{ης} τάξεως. Το βήμα είναι h=0.5.

Στο παράδειγμα του σχήματος 4.5 εφαρμόζουμε τη μέθοδο αυτή, στη Mathematica για την επίλυση του προβλήματος αρχικών τιμών:

$$y'(x) = y - x^2, \qquad y(1) = -4$$

Χρησιμοποιούμε την αρχική τιμή y(1)=-4 για να υπολογίσουμε τις τιμές της άγνωστης συνάρτησης y(x) στο διάστημα [1,3]. Το βήμα επιλέχθηκε αρχικά: h=0.5. Στη συνέχεια μεταβάλλουμε το βήμα ώστε να δούμε την επίδραση που έχει στη λύση.

Στη συνέχεια επιλύουμε το ίδιο πρόβλημα αρχικών τιμών με βήμα αρχικά h=0.25 και στη συνέχεια h=0.05. Τα αποτελέσματα φαίνονται στα σχήματα που ακολουθούν.

1.	-4
1.25	-5.49367
1.5	-7.58834
1.75	-10.4902
2.	-14.464
2.25	-19.8497
2.5	-27.0839
2.75	-36.7269
3.	-49.4984

Σχήμα 5.5 Αποτελέσματα για h=0.25.

1.	-4		
1.05	-4.25894	2.05	-15.4164
1.1	-4.53654	2.1	-16.4275
1.15	-4.83401	2.15	-17.5012
1.2	-5.15262	2.2	-18.6411
1.25	-5.49373	2.25	-19.8506
1.3	-5.85873	2.3	-21.1337
1.35	-6.24911	2.35	-22.4943
1.4	-6.66642	2.4	-23.9368
1.45	-7.11231	2.45	-25.4655
1.5	-7.58849	2.5	-27.0852
1.55	-8.09678	2.55	-28.8007
1.6	-8.63907	2.6	-30.6173
1.65	-9.21737	2.65	-32.5403
1.7	-9.83377	2.7	-34.5755
1.75	-10.4905	2.75	-36.7289
1.8	-11.1899	2.8	-39.0068
1.85	-11.9343	2.85	-41.4159
1.9	-12.7264	2.9	-43.963
1.95	-13.5689	2.95	-46.6557
2.	-14.4645	3.	-49.5015

Σχήμα 5,6 Αποτελέσματα για h=0.05.

Η γενική λύση της διαφορικής εξίσωσης είναι:

$$y(x) = Ae^{x} + x^{2} + 2x + 2$$

Αν εφαρμόσουμε την αρχική συνθήκη y(1)=-4 βρίσκουμε τη σταθερά A. Η τιμή είναι:

$$A = -\frac{9}{e}$$

Συνεπώς η ακριβής λύση του προβλήματος αρχικών τιμών είναι:

$$y(x) = -\frac{9}{e}e^{x} + x^{2} + 2x + 2$$

Από αυτήν βρίσκουμε ότι η ακριβής τιμή της συνάρτησης y(x) στο σημείο x=3 είναι:

$$y(3) = -49.5015$$

Παρατηρούμε ότι για h=0.5 το σφάλμα είναι -49.5015-(-49.4621)=0.0606 το οποίο είναι προσεγγιστικά ίσο με 0.5^4=0.0625. Ομοίως για h=0.25 το σφάλμα είναι -49.5015-(-49.4984)=-0.0031 το οποίο είναι προσεγγιστικά ίσο με 0.25^4=0.0039. Για h=0.05 το σφάλμα είναι -49.5015-(-49.5015)=0.0000 το οποίο είναι προσεγγιστικά ίσο με 0.05^4=0.00000625.

Μπορεί να αποδειχθεί ότι πράγματι το σφάλμα της μεθόδου Runge-Kutta 4ης τάξεως είναι $O(h^4)$. Για λόγους σύγκρισης το σφάλμα της μεθόδου του Euler είναι O(h). Είναι φανερό ότι η μέθοδος Runge-Kutta είναι πολύ ακριβής μέθοδος. Επίσης είναι πολύ γρήγορη μέθοδος. Για όλους αυτούς τους λόγους χρησιμοποιείται ευρέως στις εφαρμογές.

5.5 ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ LAPLACE

Ο μετασχηματισμός Laplace Y(s), μιας συνάρτησης y(x) ορίζεται από το ολοκλήρωμα:

$$L\{y(x)\} = Y(s) = \int_0^\infty e^{-sx} y(x) dx$$

Για να επιλύσουμε ένα πρόβλημα αρχικών τιμών χρησιμοποιώντας τον μετασχηματισμό Laplace. Ακολουθούμε τα εξής βήματα. Μετασχηματίζουμε και τα δύο μέλη της διαφορικής εξίσωσης μετατρέποντάς την σε αλγεβρική εξίσωση με άγνωστο τον μετασχηματισμό Laplace της ζητούμενης συνάρτησης y(x). Στο βήμα αυτό κάνουμε χρήση των παρακάτω πολύ χρήσιμων ιδιοτήτων:

$$L\{y'(x)\} = sY(s) - y(0)$$

$$L\{y^{(n)}(x)\} = s^n Y(s) - s^{n-1} y(0) - s^{n-2} y'(0) - \ldots$$

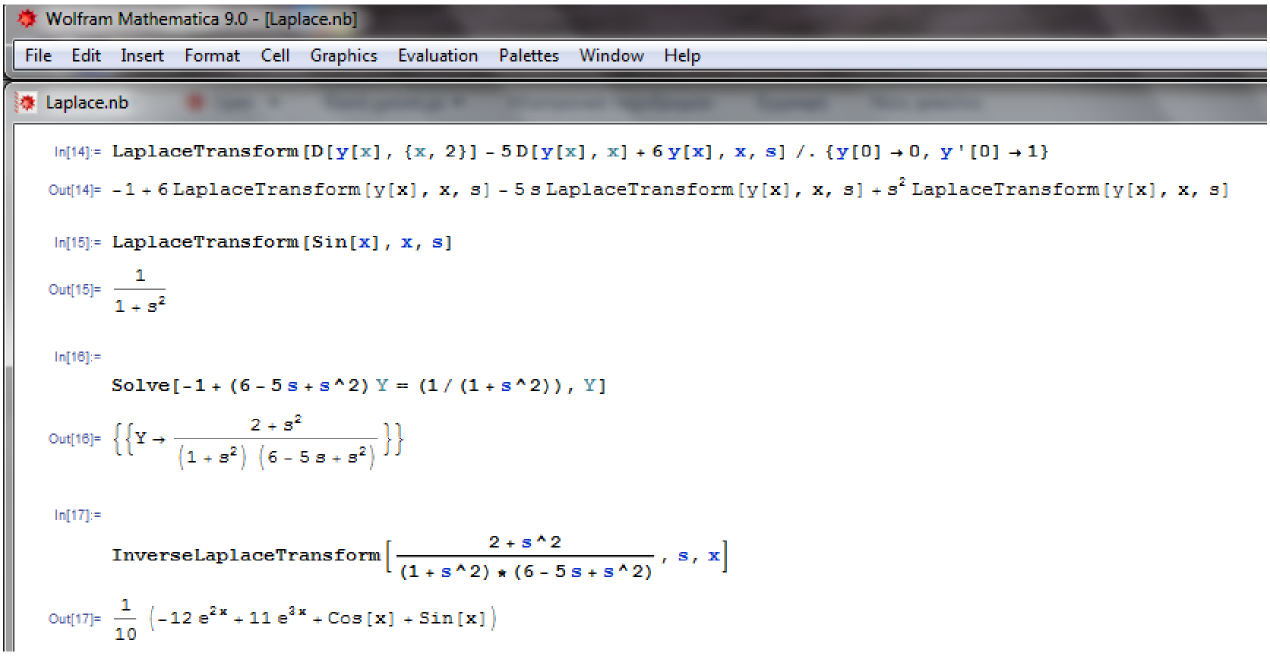
όπου y(0), y΄(0), … είναι οι τιμές της συνάρτησης y(x) και των παραγώγων της για x=0.

Στη συνέχεια επιλύουμε την αλγεβρική εξίσωση και υπολογίζουμε τη συνάρτηση Y(s). Τέλος εφαρμόζουμε αντίστροφο μετασχηματισμό Laplace και βρίσκουμε την άγνωστη συνάρτηση y(x). Ακολουθεί η επίλυση με τη μέθοδο αυτή του προβλήματος αρχικών τιμών:

$$y''(x) - 5y'(x) + 6y(x) = \sin x$$

$$y(0) = 0, \; y'(0) = 1$$

Υπολογίζουμε ξεχωριστά το μετασχηματισμό Laplace του πρώτου και του δευτέρου μέλους της διαφορικής εξίσωσης. Στη συνέχεια λύνεται η αλγεβρική εξίσωση που προκύπτει και υπολογίζεται η συνάρτηση Y(s). Τέλος, με εφαρμογή του αντίστροφου μετασχηματισμού Laplace υπολογίζεται η λύση y(x) της διαφορικής εξίσωσης.



Σχήμα 5.7 Επίλυση διαφορικής εξίσωσης με τη μέθοδο του μετασχηματισμού Laplace στη Mathematica.

6. ΤΑΛΑΝΤΩΣΕΙΣ

6.1 ΓΕΝΙΚΑ

Στο παρόν κεφάλαιο θα χρησιμοποιήσουμε τις μεθόδους επίλυσης διαφορικών εξισώσεων που παρουσιάστηκαν στο κεφάλαιο 5 στη μελέτη των ταλαντώσεων. Παρουσιάζονται παραδείγματα ελεύθερων και εξαναγκασμένων ταλαντώσεων ενός και δύο βαθμών ελευθερίας.

6.2 ΦΘΙΝΟΥΣΕΣ ΤΑΛΑΝΤΩΣΕΙΣ

Έστω ότι θέλουμε να μελετήσουμε την ταλάντωση του απλού συστήματος ενός βαθμού ελευθερίας που φαίνεται στο σχήμα 6.1. Η διαφορική εξίσωση που περιγράφει την φθίνουσα ταλάντωση του συστήματος είναι:

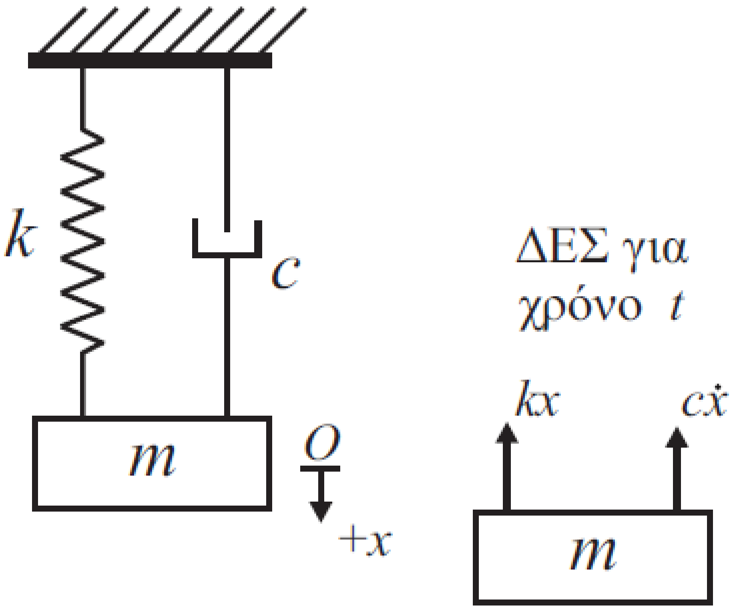
$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = 0$$

Η ιδιοσυχνότητα ταλάντωσης του αντίστοιχου μηχανικού συστήματος χωρίς τον αποσβεστήρα είναι:

$$\omega_n = \sqrt{\frac{k}{m}}$$

Η κρίσιμη τιμή της σταθεράς απόσβεσης c είναι:

$$c_{cr} = 2\sqrt{mk}$$



Σχήμα 6.1 Μηχανικό σύστημα ενός βαθμού ελευθερίας με ελατήριο και αποσβεστήρα [10].

Ο λόγος απόσβεσης J ορίζεται από τη σχέση:

$$J = \frac{c}{c_{cr}}$$

Η ιδιοσυχνότητα ω_d του συστήματος δίνεται από τη σχέση:

$$\omega_d = \omega_n \sqrt{1 - J^2}$$

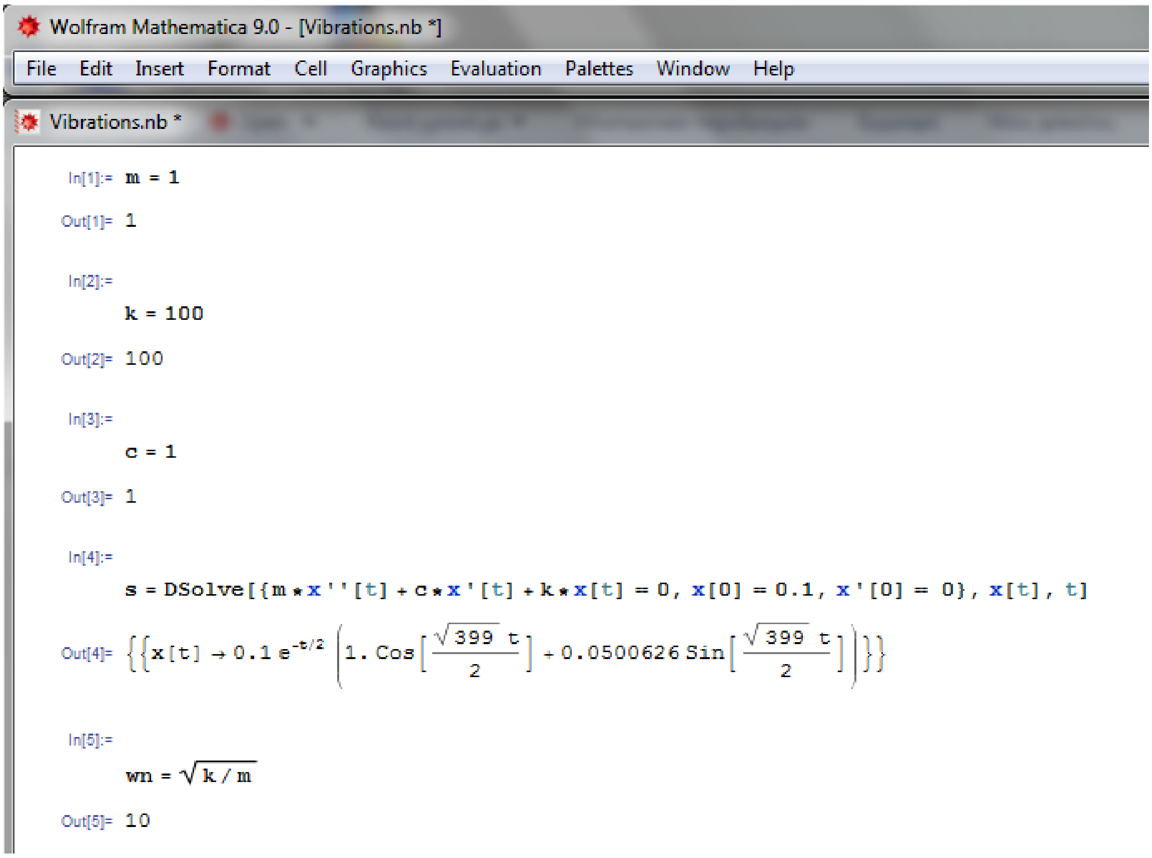
εφόσον βέβαια J<1 (Υπο-αποσβενύμενη ταλάντωση)

Για το σύστημα του σχήματος 6.1 θα θεωρήσουμε ότι για t=0 βρίσκεται στη θέση x=0.1 και έχει μηδενική αρχική ταχύτητα. Δηλαδή, οι αρχικές συνθήκες είναι:

$$x(0) = 0.1 \quad \text{και} \quad \dot{x}(0) = 0$$

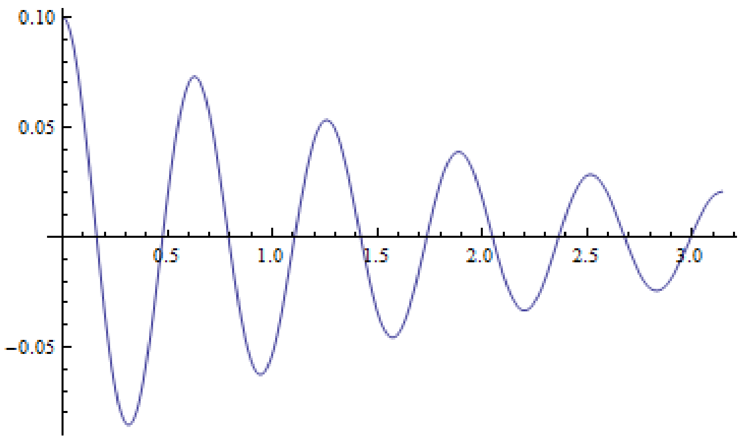
Επίσης υποθέτουμε τις τιμές: m=1, k=100 και c=1. Όλες οι τιμές θεωρείται ότι είναι στο Διεθνές Σύστημα.

Στη συνέχεια επιλύουμε το πρόβλημα αρχικών τιμών με την εντολή DSolve. Η λύση, δηλαδή η συνάρτηση x(t) και η γραφική παράσταση της φαίνονται στο παρακάτω σχήμα.



```
In[6]:=
       Ccr = 2 * √(m * k)
Out[6]= 20

In[7]:=
       J = c / Ccr
Out[7]= 1/20

In[8]:=
       wd = wn * √(1 - J ^ 2)
Out[8]= √399/2

In[9]:=
       T = 2 * 3.14 / wd
Out[9]= 0.628786
```

```
In[10]:=
       Plot[x[t] /. s[[1]], {t, 0, 5 T}]
Out[10]=
```



Σχήμα 6.2 Μελέτη φθίνουσας ταλάντωσης στη Mathematica με την εντολή DSolve.

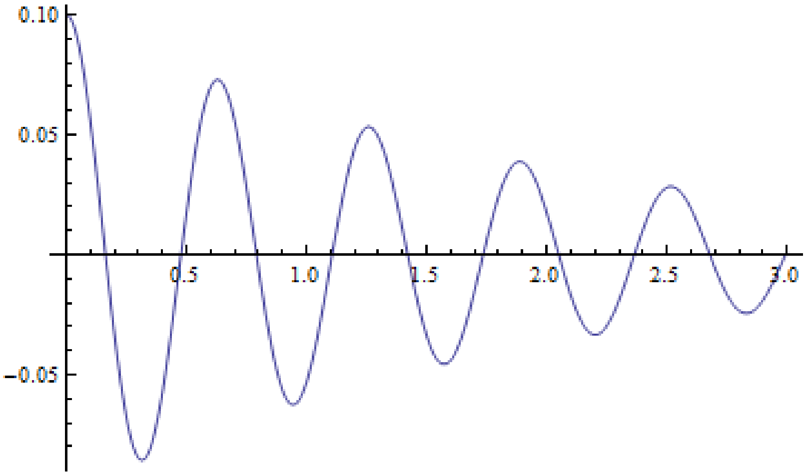
Στη συνέχεια το ίδιο πρόβλημα λύνεται αριθμητικά με την εντολή NDSolve.

Vibrations -NDSolve.nb *

```
In[33]:= m = 1
Out[33]= 1

In[34]:=
        k = 100
Out[34]= 100

In[35]:=
        c = 1
Out[35]= 1

In[36]:=
        s = NDSolve[{m * x''[t] + c * x'[t] + k * x[t] == 0, x[0] == 0.1, x'[0] == 0}, x[t], {t, 0, 3}]
Out[36]= {{x[t] → InterpolatingFunction[{{0., 3.}}, <>][t]}}

In[37]:=
        wn = √(k / m)
Out[37]= 10

In[38]:=
        Ccr = 2 * √(m * k)
Out[38]= 20

In[39]:=
        J = c / Ccr
Out[39]= 1/20

In[40]:=
        wd = wn * √(1 - J ^ 2)
Out[40]= √399 / 2

In[41]:=
        T = 2 * 3.14 / wd
Out[41]= 0.628786
```

```
In[42]:=
     Plot[x[t] /. s[[1]], {t, 0, 3}]
```

Out[42]=



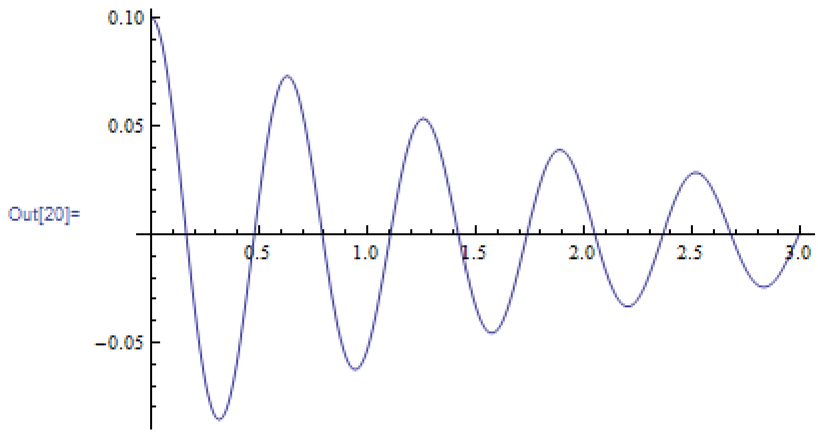
Σχήμα 6.3 Μελέτη φθίνουσας ταλάντωσης στη Mathematica με την εντολή NDSolve.

Στην εντολή NDSolve μπορούμε να δηλώσουμε τη μέθοδο με την οποία θέλουμε να χρησιμοποιηθεί από τη Mathematica για την επίλυση της διαφορικής εξίσωσης. Αντικαθιστώντας την εντολή NDSolve του παραπάνω παραδείγματος με την:

```
In[14]:=
   s = NDSolve[{m*x''[t] + c*x'[t] + k*x[t] == 0, x[0] == 0.1, x'[0] == 0}, x[t], {t, 0, 3}, Method → ExplicitRungeKutta]
```

δηλώνουμε ότι θέλουμε να χρησιμοποιηθεί η μέθοδος Runge-Kutta η οποία παρουσιάστηκε στην ενότητα 5.4.2. Το αποτέλεσμα που δίνει η Mathematica για το πρόβλημα της φθίνουσας ταλάντωσης είναι:

Παρατηρούμε ότι οι δύο παραπάνω αριθμητικές επιλύσεις της διαφορικής εξίσωσης της φθίνουσας ταλάντωσης παρουσιάζουν πολύ μεγάλη ακρίβεια και ουσιαστικά ταυτίζονται με την αναλυτική λύση.



Σχήμα 6.4 Αριθμητική επίλυση της διαφορικής εξίσωσης που περιγράφει τη φθίνουσα ταλάντωση με τη μέθοδο Runge-Kutta.

6.3 ΕΞΑΝΑΓΚΑΣΜΕΝΕΣ ΤΑΛΑΝΤΩΣΕΙΣ

Έστω ότι στο μηχανικό σύστημα του σχήματος 5.8 ασκείται μία εξωτερική περιοδική δύναμη. Η εξίσωση κίνησης γίνεται:

$$m\ddot{x}(t)+c\dot{x}(t)+kx(t)=F_0\cos(\Omega t)$$

Αν η κυκλική συχνότητα Ω της διεγείρουσας δύναμης είναι ίση με την ιδιοσυχνότητα ω_d του μηχανικού συστήματος τότε έχουμε συντονισμό και το πλάτος της εξαναγκασμένης ταλάντωσης γίνεται μέγιστο. Για συχνότητες Ω κοντά στην ω_d το πλάτος έχει γενικά μεγαλύτερες τιμές σε σχέση με την περίπτωση η Ω να διαφέρει πολύ από την ω_d.

Στη συνέχεια παραθέτουμε δύο παραδείγματα επίλυσης με τη Mathematica της παραπάνω διαφορικής εξίσωσης για F_0=20 και για δύο διαφορετικές τιμές της Ω, μία μακριά και μία πολύ κοντά στο συντονισμό.

Wolfram Mathematica 9.0 - [Vibrations 2.nb]

File Edit Insert Format Cell Graphics Evaluation Palettes Window Help

Vibrations 2.nb

m = 1

1

k = 100

100

c = 1

1

wf = 5

5

s = DSolve[{m*x''[t] + c*x'[t] + k*x[t] == 20*Cos[wf*t], x[0] == 0.1, x'[0] == 0}, x[t], t]

$$\Big\{\Big\{x[t] \to 0.00330472\, e^{-t/2} \Big(-50.0758\,\mathrm{Cos}\Big[\tfrac{\sqrt{399}\,t}{2}\Big] + 60.1428\, e^{t/2}\,\mathrm{Cos}\Big[\tfrac{\sqrt{399}\,t}{2}\Big]\,\mathrm{Cos}\Big[\tfrac{1}{2}\big(-10+\sqrt{399}\big)\,t\Big] + 20.1928\, e^{t/2}\,\mathrm{Cos}\Big[\tfrac{\sqrt{399}\,t}{2}\Big]\,\mathrm{Cos}\Big[\tfrac{1}{2}\big(10+\sqrt{399}\big)\,t\Big] - 5.18813\,\mathrm{Sin}\Big[\tfrac{\sqrt{399}\,t}{2}\Big] +$$

$$6.02936\, e^{t/2}\,\mathrm{Cos}\Big[\tfrac{1}{2}\big(-10+\sqrt{399}\big)\,t\Big]\,\mathrm{Sin}\Big[\tfrac{\sqrt{399}\,t}{2}\Big] + 0.673655\, e^{t/2}\,\mathrm{Cos}\Big[\tfrac{1}{2}\big(10+\sqrt{399}\big)\,t\Big]\,\mathrm{Sin}\Big[\tfrac{\sqrt{399}\,t}{2}\Big] + 60.1428\, e^{t/2}\,\mathrm{Sin}\Big[\tfrac{\sqrt{399}\,t}{2}\Big]\,\mathrm{Sin}\Big[\tfrac{1}{2}\big(-10+\sqrt{399}\big)\,t\Big] -$$

$$0.673655\, e^{t/2}\,\mathrm{Cos}\Big[\tfrac{\sqrt{399}\,t}{2}\Big]\,\mathrm{Sin}\Big[\tfrac{1}{2}\big(10+\sqrt{399}\big)\,t\Big] + 20.1928\, e^{t/2}\,\mathrm{Sin}\Big[\tfrac{\sqrt{399}\,t}{2}\Big]\,\mathrm{Sin}\Big[\tfrac{1}{2}\big(10+\sqrt{399}\big)\,t\Big] + 6.02936\, e^{t/2}\,\mathrm{Cos}\Big[\tfrac{\sqrt{399}\,t}{2}\Big]\,\mathrm{Sin}\Big[5\,t - \tfrac{\sqrt{399}\,t}{2}\Big]\Big)\Big\}\Big\}$$

$wn = \sqrt{k/m}$

10

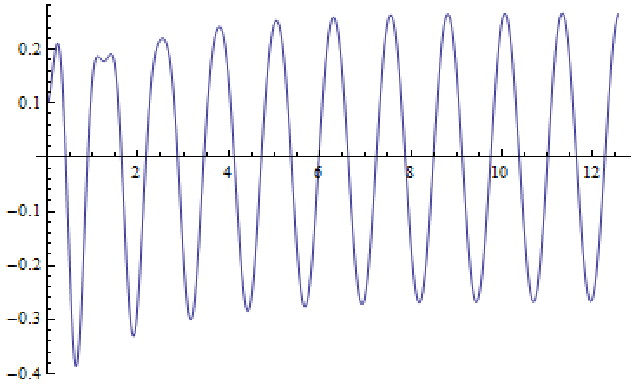
$Ccr = 2*\sqrt{m*k}$

20

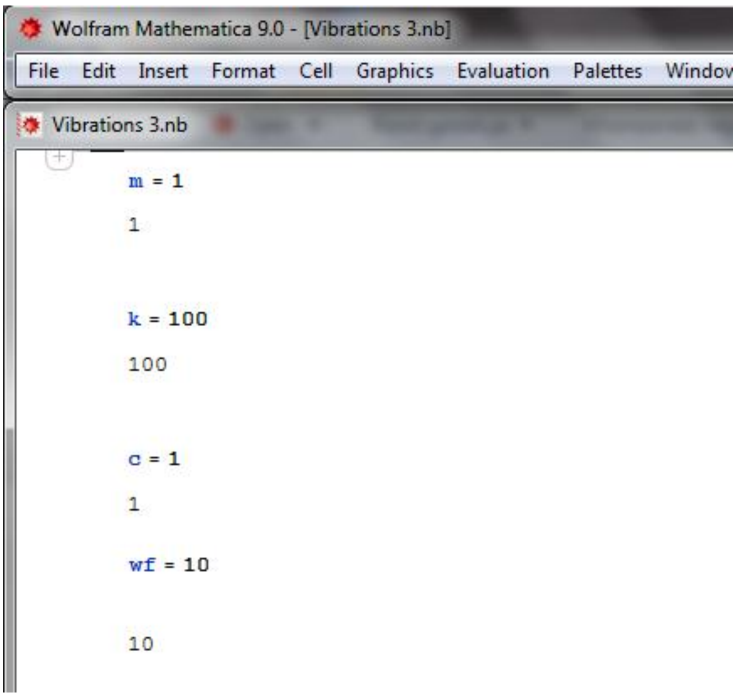
J = c / Ccr

$\frac{1}{20}$

```
wd = wn * √(1 - J ^ 2)

√399
────
 2

T = 2 * 3.14 / wd

0.628786

Tf = 2 * 3.14 / wf

1.256

Plot[x[t] /. s[[1]], {t, 0, 10 Tf}]
```



Σχήμα 6.5 Εξαναγκασμένη ταλάντωση μακριά από το συντονισμό.



```
s = DSolve[{m*x''[t] + c*x'[t] + k*x[t] = 20*Cos[wf*t], x[0] = 0.1, x'[0] = 0}, x[t], t]
```

$$\Big\{\Big\{x[t] \to -0.00250627\, e^{-t/2} \Big(-39.9\, \mathrm{Cos}\Big[\frac{\sqrt{399}\, t}{2}\Big] + \sqrt{399}\, e^{t/2}\, \mathrm{Cos}\Big[\frac{\sqrt{399}\, t}{2}\Big]\, \mathrm{Cos}\Big[\frac{1}{2}\left(-20+\sqrt{399}\right) t\Big] - 19.975\, e^{t/2}\, \mathrm{Cos}\Big[\frac{\sqrt{399}\, t}{2}\Big]\, \mathrm{Cos}\Big[\frac{1}{2}\left(20+\sqrt{399}\right) t\Big] + 797.002\, \mathrm{Sin}\Big[\frac{\sqrt{399}\, t}{2}\Big] -$$

$$798.5\, e^{t/2}\, \mathrm{Cos}\Big[\frac{1}{2}\left(-20+\sqrt{399}\right) t\Big]\, \mathrm{Sin}\Big[\frac{\sqrt{399}\, t}{2}\Big] - 0.499687\, e^{t/2}\, \mathrm{Cos}\Big[\frac{1}{2}\left(20+\sqrt{399}\right) t\Big]\, \mathrm{Sin}\Big[\frac{\sqrt{399}\, t}{2}\Big] + \sqrt{399}\, e^{t/2}\, \mathrm{Sin}\Big[\frac{\sqrt{399}\, t}{2}\Big]\, \mathrm{Sin}\Big[\frac{1}{2}\left(-20+\sqrt{399}\right) t\Big] +$$

$$0.499687\, e^{t/2}\, \mathrm{Cos}\Big[\frac{\sqrt{399}\, t}{2}\Big]\, \mathrm{Sin}\Big[\frac{1}{2}\left(20+\sqrt{399}\right) t\Big] - 19.975\, e^{t/2}\, \mathrm{Sin}\Big[\frac{\sqrt{399}\, t}{2}\Big]\, \mathrm{Sin}\Big[\frac{1}{2}\left(20+\sqrt{399}\right) t\Big] - 798.5\, e^{t/2}\, \mathrm{Cos}\Big[\frac{\sqrt{399}\, t}{2}\Big]\, \mathrm{Sin}\Big[10\, t - \frac{\sqrt{399}\, t}{2}\Big]\Big)\Big\}\Big\}$$

$wn = \sqrt{k/m}$

10

$Ccr = 2*\sqrt{m*k}$

20

```
J = c / Ccr
```

$\frac{1}{20}$

$wd = wn*\sqrt{1 - J^2}$

$\frac{\sqrt{399}}{2}$

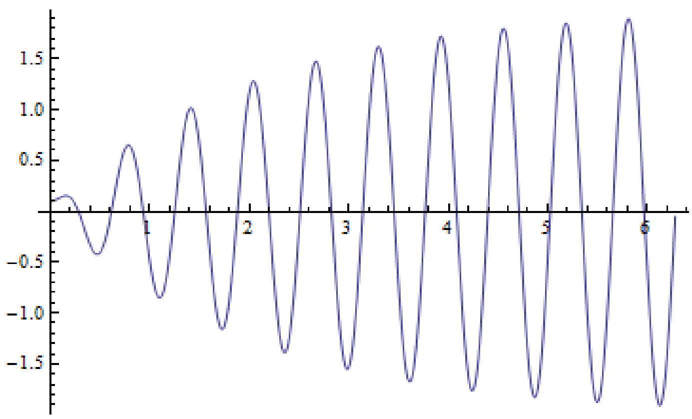
```
T = 2*3.14 / wd
```

0.628786

```
Tf = 2*3.14 / wf
```

0.628

```
Plot[x[t] /. s[[1]], {t, 0, 10 Tf}]
```

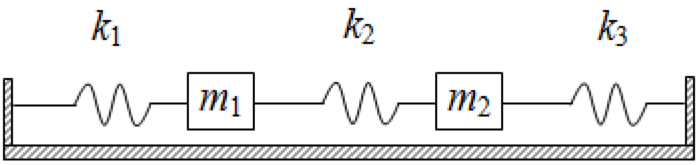


Σχήμα 6.6 Εξαναγκασμένη ταλάντωση κοντά στο συντονισμό.

Και στις δύο περιπτώσεις εξαναγκασμένων ταλαντώσεων που μελετήθηκαν παρατηρούμε ότι μετά από λίγες περιόδους το σύστημα φτάνει σε μία μόνιμη κατάσταση όπου ταλαντώνεται με σταθερό πλάτος και με συχνότητα ίση με τη συχνότητα του διεγέρτη. Φυσικά, κοντά στον συντονισμό το πλάτος της εξαναγκασμένης ταλάντωσης είναι μεγαλύτερο.

6.4 ΤΑΛΑΝΤΩΣΕΙΣ ΔΥΟ ΒΑΘΜΩΝ ΕΛΕΥΘΕΡΙΑΣ

Θα μελετήσουμε την ελεύθερη ταλάντωση ενός συστήματος δύο βαθμών ελευθερίας χωρίς απόσβεση. Το σύστημα φαίνεται στο παρακάτω σχήμα.



Σχήμα 6.7 Σύστημα δύο βαθμών ελευθερίας χωρίς απόσβεση.

Για λόγους απλότητας θέτουμε: $m_1=m_2=m$ και $k_1=k_2=k_3=k$. Οι διαφορικές εξισώσεις που περιγράφουν το σύστημα είναι:

m_1: $m\ddot{x}_1 + 2kx_1 - kx_2 = 0$

m_2: $m\ddot{x}_2 - kx_1 + 2kx_2 = 0$

Έστω m=1 και k=100. Το σύστημα γίνεται:

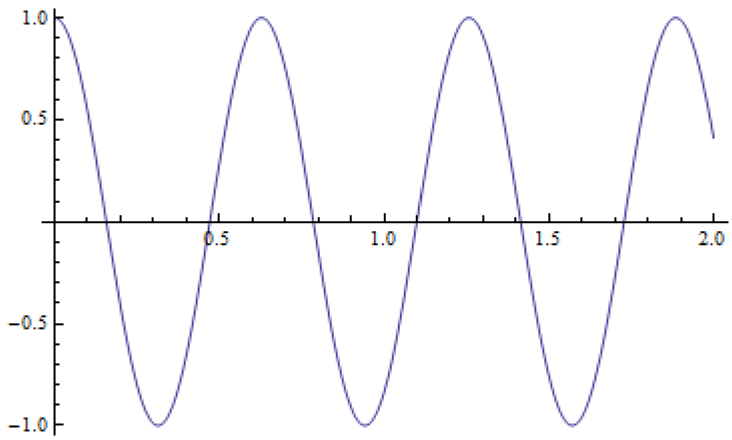
$$\ddot{x}_1 + 200x_1 - 100x_2 = 0$$

$$\ddot{x}_2 - 100x_1 + 200x_2 = 0$$

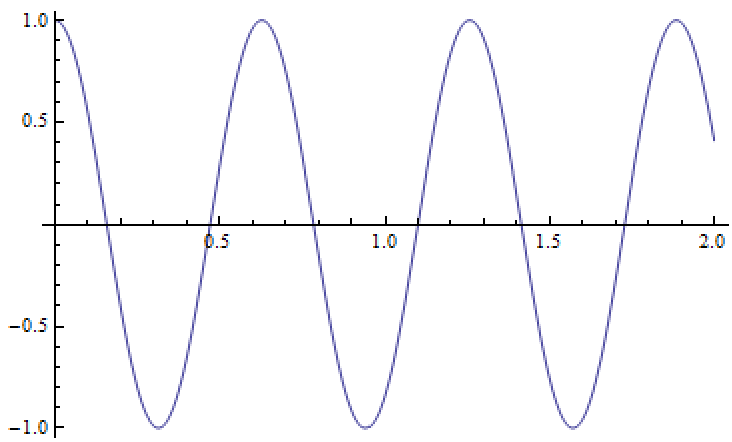
Στη συνέχεια το σύστημα αυτό θα επιλυθεί στη Mathematica με την εντολή DSolve και με τη μέθοδο των ιδιοτιμών-ιδιοδιανυσμάτων.

Vibration_2df.nb *

```
In[88]:= s = DSolve[
    {D[x1[t], {t, 2}] == -200 x1[t] + 100 x2[t],
     D[x2[t], {t, 2}] == -200 x2[t] + 100 x1[t],
     x1[0] == 1, x2[0] == 1,
     x1'[0] == 0, x2'[0] == 0},
    {x1[t], x2[t]}, t]

Out[88]= {{x1[t] → Cos[10 t], x2[t] → Cos[10 t]}}

In[89]:=
  Plot[x1[t] /. s[[1]], {t, 0, 2}]

Out[89]=
```




```
In[90]:=
  Plot[x2[t] /. s[[1]], {t, 0, 2}]

Out[90]=
```



Σχήμα 6.8 Επίλυση προβλήματος ταλάντωσης δύο βαθμών ελευθερίας με την εντολή DSolve.

Το παραπάνω σύστημα γράφεται σε μητρωική μορφή ως εξής:

$$\begin{pmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{pmatrix} = \begin{pmatrix} -200 & 100 \\ 100 & -200 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

Στο παράδειγμα που ακολουθεί υπολογίζονται οι ιδιοτιμές και τα ιδιοδιανύσματα του πίνακα του συστήματος και στη συνέχεια επιλύεται το σύστημα.

vibration_2df_b.nb

```
In[33]:= A = {{-200, 100}, {100, -200}};

In[34]:=
     Eigenvalues[A]

Out[34]= {-300, -100}

In[35]:= Eigenvectors[A]

Out[35]= {{-1, 1}, {1, 1}}

In[36]:= X[t_] = {x1[t], x2[t]}

Out[36]= {x1[t], x2[t]}

In[37]:=
     system = X''[t] == A.X[t]

Out[37]= {x1''[t], x2''[t]} == {-200 x1[t] + 100 x2[t], 100 x1[t] - 200 x2[t]}

In[38]:=
     InitialConditions = {x1[0] == 1, x2[0] == 1, x1'[0] == 0, x2'[0] == 0}

Out[38]= {x1[0] == 1, x2[0] == 1, x1'[0] == 0, x2'[0] == 0}

In[39]:=
     s = DSolve[{system, InitialConditions}, {x1[t], x2[t]}, t]

Out[39]= {{x1[t] → Cos[10 t], x2[t] → Cos[10 t]}}
```

Σχήμα 6.9 Ιδιοτιμές και ιδιοδιανύσματα του πίνακα του συστήματος διαφορικών εξισώσεων.

7. ΣΥΜΠΕΡΑΣΜΑΤΑ

Το λογισμικό Mathematica είναι ένα πολύ ισχυρό μαθηματικό εργαλείο με μεγάλο εύρος εφαρμογών. Χαρακτηριστικό του είναι η δυνατότητα εκτέλεσης συμβολικών υπολογισμών. Έτσι σε αντίθεση με τις συνηθισμένες γλώσσες προγραμματισμού, η Mathematica έχει τη δυνατότητα να δίνει αναλυτικές λύσεις.

Στην παρούσα εργασία με χρήση της Mathematica αντιμετωπίστηκαν προβλήματα επίλυσης εξισώσεων, γραμμικής άλγεβρας, ανάλυσης και διαφορικών εξισώσεων. Παρουσιάστηκαν αναλυτικές και αριθμητικές λύσεις και σε πολλές περιπτώσεις αυτές συγκρίθηκαν μεταξύ τους.

Με την εργασία αυτή αποκτήθηκε σημαντική εμπειρία στη χρήση και την ανάπτυξη προγραμμάτων στη Mathematica καθώς και στην εφαρμογή αριθμητικών μεθόδων για την επίλυση προβλημάτων των εφαρμοσμένων μαθηματικών.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1] Παπαδάκης Κ., Εισαγωγή στο Mathematica, εκδόσεις Τζιόλα 2009.
2] Θεοδώρου Γ. και Θεοδώρου Χρ., Πρακτικός Οδηγός για τη Mathematica, ΑΠΘ 2004.
3] Σταματιάδης Σ., Εισαγωγή στην Αριθμητική Ανάλυση, Πανεπιστήμιο Κρήτης 2017.
4] Μπράτσος Α., Μαθήματα Εφαρμοσμένων Μαθηματικών, ΣΕΑΒ 2015.
5] Πάσχος Κ., Μελέτη και Επίλυση Διαφορικών Εξισώσεων και Συστημάτων Διαφορικών Εξισώσεων, Διπλωματική Εργασία, Πανεπιστήμιο Μακεδονίας 2008.
6] Αθανασίου Γ., Οδηγός για το Λογισμικό Mathematica, Πανεπιστήμιο Αθηνών 2007.
7] Ιωακειμίδης Ν., Εφαρμοσμένα Μαθηματικά ΙΙΙ, εκδόσεις Gotsis 2008.
8] Παπαιωάννου Σ. και Βοζίκης Χ., Αριθμητική Ανάλυση, ΣΕΑΒ 2015.
9] Don E., Schaum's Outlines Mathematica, McGraw-Hill 2009.
10] Τσινόπουλος Σ., Μηχανικές Ταλαντώσεις, Σημειώσεις, Τμήμα Μηχανολόγων Μηχανικών Τ.Ε., ΤΕΙ Δυτικής Ελλάδας 2007.
11] Torrence B. and Torrence E., The Student' s Introduction to Mathematica, Cambridge University Press 2009.