Από την Άλγεβρα των Υπολογισμών στα Υπολογιστικά Συστήματα Άλγεβρας

Επικ. Καθηγητής Νικόλαος Καραμπετάκης
Τμήμα Μαθηματικών
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Θεσσαλονίκη 54006
Email : karampet@math.auth.gr

Περίληψη. Μια νέα ερευνητική περιοχή που αναπτύχθηκε ιδιαίτερα τις τελευταίες τέσσερις δεκαετίες και η οποία είναι αφιερωμένη : α) σε μεθόδους επίλυσης μαθηματικών προβλημάτων μέσω *συμβολικών* αλγορίθμων, και β) στην υλοποίηση των αλγορίθμων αυτών σε υλικό και λογισμικό ηλεκτρονικών υπολογιστών, είναι η *Υπολογιστική Άλγεβρα* (*Computer Algebra*). Η Υπολογιστική Άλγεβρα έχει εφαρμογές σε επιστήμες όπως Μαθηματικά, Φυσική, Επιστήμη των Υπολογιστών, Μηχανική καθώς και στην Εκπαίδευση. Τα προγράμματα τα οποία κάνουν χρήση των μεθόδων της Υπολογιστικής Άλγεβρας, ονομάζονται *Υπολογιστικά Συστήματα Άλγεβρας* (*Computer Algebra Systems*). Στην εργασία αυτή θα αναφερθούμε : στην ιστορική εξέλιξη των συστημάτων αυτών και στους παράγοντες που συντέλεσαν στην ανάπτυξη τους, στις κατηγορίες που χωρίζονται τα συστήματα αυτά, στα κύρια χαρακτηριστικά τους, στα μειονεκτήματα τους καθώς και στον ρόλο που μπορούν να παίξουν στην εκπαίδευση.

1. Το πέρασμα από την Άλγεβρα στην Υπολογιστική Άλγεβρα.

Η ιστορία της Άλγεβρας ξεκινάει από τους αρχαίους Αιγύπτιους και Βαβυλώνιους οι οποίοι ασχολήθηκαν με την επίλυση γραμμικών και δευτεροβάθμιων εξισώσεων όπως και με εξισώσεις με περισσότερους από έναν αγνώστους.

Την παράδοση των Αιγυπτίων και των Βαβυλωνίων συνέχισε ο Έλληνας μαθηματικός Διόφαντος από την Αλεξάνδρεια με τα 13 βιβλία που έγραψε γύρω στα 250 π.χ., από τα οποία σώθηκαν μόνο τα 6. Ένα από τα πιο σημαντικά του έργα είναι τα *Αριθμητικά*, στο οποίο ασχολήθηκε με την επίλυση εξισώσεων μιας ή και περισσότερων μεταβλητών, με την βοήθεια αριθμητικών μεθόδων. Ήταν ο πρώτος που χρησιμοποίησε συμβολισμούς, ένα ενδιάμεσο στάδιο μεταξύ ρητορικής και συμβολικής άλγεβρας. Το έργο του μεταφράστηκε και μελετήθηκε πολύ από τους Άραβες μαθηματικούς.

Η λέξη «*Άλγεβρα*» προέρχεται από την λατινική μεταγραφή της λέξης «*al jabr*» η οποία με την σειρά της προέρχεται από την αλγεβρική πραγματεία «Kitab al jabr wa'l mugabala» (Οι κανόνες της αποκατάστασης και της αναγωγής) που γράφτηκε το 825 μ.Χ. από έναν από τους πιο διάσημους Άραβες μαθηματικούς, τον Ibn Musa Al-Khwarizmi (780-850 μ.Χ.). Η εργασία αυτή είχε την μεγαλύτερη επίδραση που είχε ποτέ εργασία στην Άλγεβρα. Ο Al-Khwarizmi, στην πραγματεία αυτή που ασχολήθηκε με την επίλυση γραμμικών και δευτεροβάθμιων εξισώσεων, με τον όρο «al jabr» περίγραφε τη δυνατότητα μεταφοράς όρων από το ένα μέλος της εξίσωσης

στο άλλο με σύγχρονη αλλαγή πρόσημου, ενώ με τον όρο «mugabala» που σημαίνει «αναγωγή», περίγραφε αυτό που λέμε αναγωγή όμοιων όρων. Η εργασία του Al-Khwarizmi έγινε γνωστή στην Ιταλία σαν «Liber Algorismi» (Βιβλίο του Al-Khwarizmi), και από αυτήν την έκφραση παρέμεινε για ότι έχει να κάνει με υπολογιστικούς μεθόδους η ορολογία «algorismi», η οποία μεταφέρθηκε στην αγγλοσαξωνική γλώσσα ως «algorithm» και στη συνέχεια στα Ελληνικά ως «*αλγόριθμος*».

Από την εποχή του Al-Khwarizmi έως και το τέλος του 19^{ου} αιώνα, ο κύριος στόχος της Άλγεβρας ήταν ο αλγοριθμικός χειρισμός συμβολικών αλγεβρικών εκφράσεων. Ανάμεσα στα ονόματα διάσημων Μαθηματικών που ασχολήθηκαν με τα προβλήματα αυτά ήταν οι Ιταλοί μαθηματικοί Leonardo Fibonacci, Scipione del Fero, Niccolo Tartaglia και Gerolamo Cardano που ασχολήθηκαν με την επίλυση της κυβικής εξίσωσης, καθώς και ο Ludovico Ferrari που ασχολήθηκε με την επίλυση της εξίσωσης 4^{ου} βαθμού. Άλλα σπουδαία ονόματα Μαθηματικών της εποχής αυτής είναι των Rene Descartes (εισαγωγή συμβόλων και αναλυτική γεωμετρία), Carl Friedrich Gauss (κάθε πολυωνυμική εξίσωση έχει ρίζα στο μιγαδικό επίπεδο), Niels Abel και Evariste Galois (μη ύπαρξη κλειστής φόρμουλας για την πεμπτοβάθμια εξίσωση). Στις αρχές του 20^{ου} αιώνα υπάρχει μια στροφή από την Άλγεβρα των υπολογισμών στην *Αφηρημένη* ή αλλιώς *Μοντέρνα Άλγεβρα*, η οποία μελετά αλγεβρικές δομές όπως ομάδες, δακτυλίους, modules, διανυσματικούς χώρους, άλγεβρες.

Από το 1939 έως και σήμερα έχουμε μια ραγδαία ανάπτυξη των υπολογιστών, και της ψηφιακής επεξεργασίας δεδομένων η οποία συνοδεύτηκε από αντίστοιχη πρόοδο και εξέλιξη στον τομέα της *Αριθμητικής Ανάλυσης*. Η *Αριθμητική Ανάλυση* αποτελεί κλάδο των Μαθηματικών που ασχολείται με την εύρεση, ανάπτυξη και εφαρμογή υπολογιστικών μεθόδων οι οποίες αναλύουν σύνθετους μαθηματικούς υπολογισμούς σε απλές πράξεις εκτελέσιμες από έναν ηλεκτρονικό υπολογιστή, και έχει εφαρμογές στην Μηχανική, Στατιστική, Επιχειρησιακή Έρευνα, Φυσική, Αστρονομία, Μετεωρολογία, Ναυπηγική, Τοπογραφία, κ.λ.π.. Κύριος στόχος της Αριθμητικής Ανάλυσης είναι η εύρεση *προσεγγιστικών λύσεων* σε προβλήματα όπου δεν υπάρχει κλειστή φόρμουλα λύσης ή όπου η διάσταση του προβλήματος είναι πολύ μεγάλη (επίλυση εξισώσεων σε μετεωρολογία, πυρηνική φυσική, γεωλογία και άλλες επιστήμες) και ο χρόνος επίλυσης είναι καθοριστικός.

Πολλές φορές όμως μας έχει τύχει να θέλουμε να υπολογίσουμε την *ακριβή λύση* ενός προβλήματος όπως για παράδειγμα η επίλυση ενός γραμμικού συστήματος εξισώσεων που εμπεριέχει παραμέτρους ή γενικά θέλουμε να επαναλάβουμε έναν αυξημένο αριθμό *συμβολικών υπολογισμών* για την επίλυση ενός μαθηματικού προβλήματος. Η επιθυμία να χρησιμοποιήσουμε τον υπολογιστή για την επίτευξη των πολλών και επίπονων συμβολικών υπολογισμών οδήγησε σε ένα καινούριο ερευνητικό πεδίο που έχει ως κύριο στόχο την ανάπτυξη : α) συστημάτων (υλικού (hardware) και λογισμικού (software)) για συμβολικές πράξεις, και β) αποδοτικών συμβολικών αλγορίθμων για την επίλυση μαθηματικά τυποποιημένων προβλημάτων. Αυτή η νέα ερευνητική περιοχή που αναπτύχθηκε ιδιαίτερα τις τελευταίες 4 δεκαετίες και η οποία ανήκει στην συνοριακή περιοχή των *Μαθηματικών* και της *Επιστήμης των Υπολογιστών* ονομάστηκε *Υπολογιστική Άλγεβρα* (*Computer Algebra*). Μερικά από τα κύρια σημεία στα οποία επικεντρώνεται η *Υπολογιστική Άλγεβρα* [2], είναι η μελέτη της αναπαράστασης αλλά και του χειρισμού μεγάλων ακεραίων, πολυωνύμων, ρητών συναρτήσεων αλλά και αλγορίθμων που συσχετίζονται με αυτά (όπως για παράδειγμα

ο ευκλείδειος αλγόριθμος και η εύρεση του μέγιστου κοινού διαιρέτη), η μελέτη αλγορίθμων για τον χειρισμό πολυωνύμων μιας ή και περισσοτέρων μεταβλητών (όπως ο αλγόριθμος της διαίρεσης, της παραγοντοποίησης πολυωνύμων, της εύρεσης των Groebner βάσεων), η μελέτη αλγορίθμων για την επίλυση γραμμικών συστημάτων (πάνω σε διαφορετικά πεδία ορισμών), η εύρεση κανονικών μορφών πινάκων (όπως η Smith κανονική μορφή και η Jordan κανονική μορφή), καθώς και η μελέτη αλγορίθμων για την θεωρία αριθμών, την μεταθετική άλγεβρα, την αλγεβρική γεωμετρία, την θεωρία ομάδων, την θεωρία αναπαράστασης, τον λογισμό (αθροίσματα και ολοκληρώματα, διαφορικές εξισώσεις και εξισώσεις διαφορών, δυναμικά συστήματα), την αλγεβρική θεωρία πολυπλοκότητας κ.α.. Η Υπολογιστική Άλγεβρα μελετάει επίσης θέματα σχετικά με την Επιστήμη Υπολογιστών όπως αναπαράσταση γνώσης, αφηρημένοι τύποι δεδομένων, σχεδιασμός υπολογιστικών συστημάτων άλγεβρας, παράλληλα υπολογιστικά συστήματα άλγεβρας, μέσα αλληλεπίδρασης (interfaces), προτυποποίηση καθώς και υλοποίηση σε hardware των υπολογιστικών συστημάτων άλγεβρας.

Περιοχές στις οποίες η Υπολογιστική Άλγεβρα έχει εφαρμογές σήμερα είναι στην Φυσική (Φυσική Στοιχειωδών Σωματιδίων, Θεωρία Βαρύτητας, Διαφορική Γεωμετρία, Διαφορικές Εξισώσεις κ.α.), στα Μαθηματικά, στην επιστήμη των υπολογιστών (θεωρία κωδίκων και κρυπτογραφία, σχεδιασμός VLSI κυκλωμάτων, επεξεργασία σήματος, συστήματα αναπαράστασης γνώσης στα Μαθηματικά κ.α.), στους Μηχανικούς (ρομποτική, σχεδίαση και μοντελοποίηση με βοήθεια Η/Υ, ψηφιακή επεξεργασία ήχου κ.α.), και στην Εκπαίδευση (ως βοηθητικό μέσο διδασκαλίας, αλλά και ως αντικείμενο διδασκαλίας).

2. Υπολογιστικά Συστήματα Άλγεβρας.

Τα προγράμματα τα οποία κάνουν χρήση των μεθόδων της Υπολογιστικής Άλγεβρας, ονομάζονται *Υπολογιστικά Συστήματα Άλγεβρας* (ΥΣΑ) (*Computer Algebra Systems*). Τα ΥΣΑ σε καμιά περίπτωση δεν πρέπει να θεωρηθούν ως ανταγωνιστικά των αριθμητικών προγραμμάτων, αλλά αντίθετα ως συμπληρωματικά εργαλεία. Τα ΥΣΑ μπορούν να χωριστούν σε δύο μεγάλες κατηγορίες ανάλογα με την χρήση τους : α) τα *ΥΣΑ γενικού σκοπού* (*general purpose CAS*) ή αλλιώς τα συστήματα που εμπεριέχουν συναρτήσεις για τα περισσότερα πεδία των Μαθηματικών π.χ. *Macsyma*, *Reduce*, *Maple*, *Mathematica*, κ.α., και β) τα *ΥΣΑ ειδικού σκοπού* (*special purpose CAS*) ή αλλιώς τα συστήματα τα οποία ειδικεύονται σε συγκεκριμένες περιοχές των μαθηματικών π.χ. *PARI* (Θεωρία Αριθμών), *DELiA* (Διαφορικές Εξισώσεις) κ.α.. Πέρα από τις δύο αυτές κατηγορίες έχουμε επίσης *πακέτα* (*packages*) προγραμμάτων τα οποία έχουν δημιουργηθεί στην γλώσσα προγραμματισμού ενός ΥΣΑ γενικού σκοπού π.χ. το *CALI* είναι ένα πακέτο του ΥΣΑ *REDUCE* το οποίο περιέχει αλγορίθμους για μεταθετική άλγεβρα, το *Control System Professional* είναι ένα πακέτο του *MATHEMATICA* το οποίο εμπεριέχει αλγορίθμους για την Θεωρία Ελέγχου κ.α..

Η εξέλιξη των ΥΣΑ επηρεάστηκε από 3 βασικούς παράγοντες : α) την ανάπτυξη των γλωσσών προγραμματισμού, β) την ανάπτυξη αποδοτικών μαθηματικών αλγορίθμων για τον χειρισμό πολυωνύμων, ρητών συναρτήσεων και ακόμα πιο γενικών συναρτήσεων, γ) το πλήθος των εφαρμογών που δημιούργησε την τεράστια ώθηση στην ανάπτυξη ΥΣΑ και αλγορίθμων. Για παράδειγμα τα πρώτα ΥΣΑ δημιουργήθηκαν στην γλώσσα προγραμματισμού LISP (που αναπτύχθηκε το

1960/61) π.χ. το πρόγραμμα SAINT δημιουργήθηκε σε LISP το 1960 και είχε ως στόχο την συμβολική ολοκλήρωση συναρτήσεων. Αργότερα, νέα ΥΣΑ, όπως το MAPLE και το MATHEMATICA, δημιουργήθηκαν στην γλώσσα προγραμματισμού C, η οποία μπορούσε να διαχειριστεί καλύτερα λειτουργικούς πόρους του υπολογιστή όπως η μνήμη. Τα νέα αυτά ΥΣΑ κατανάλωναν λιγότερη μνήμη και συνεπώς μπορούσαν να χρησιμοποιηθούν σε μικρότερα συστήματα υπολογιστών, γεγονός που αποτέλεσε το λόγο της ευρείας διάδοσης τους. Στον παρακάτω πίνακα έχουμε μια μικρή ιστορική αναδρομή στα πιο σημαντικά ΥΣΑ των τελευταίων τεσσάρων δεκαετιών.

Χρόνος	**Υπολογιστικό Σύστημα Άλγεβρας**	**Σκοπός**
1961	SAINT	Αόριστα Ολοκληρώματα
1964-66	ALTRAN, MATHLAB	Χειρισμός πολυωνυμικών και ρητών συναρτήσεων
1966-67	SIN	Συμβολική ολοκλήρωση
1968 - σήμερα	REDUCE http://www.rrz.uni-koeln.de/REDUCE	Ξεκίνησε για υπολογισμούς στην Φυσική. Επιλύει προβλήματα μεγάλης κλίμακας σε Μαθηματικά, Φυσικές Επιστήμες, και στην Επιστήμη των Μηχανικών.
1968	MATHLAB-68	Νέα έκδοση του Mathlab
1968 - σήμερα	MACSYMA http://www.macsyma.com	Γενικού σκοπού ΥΣΑ
Τέλη 1970's	muMATH	
1980	MAPLE http://www.maplesoft.com	Γενικού σκοπού ΥΣΑ
Αρχές 1980's	DERIVE	Γενικού σκοπού ΥΣΑ, νέα έκδοση του muMATH
1984 – σήμερα	SINGULAR http://www.mathematik.uni-kl.de/pub/~zca/Singular	ΥΣΑ για πολυωνυμικούς υπολογισμούς
1988 – σήμερα	SMP, MATHEMATICA http://www.wolfram.com	Γενικού σκοπού ΥΣΑ
1989 - σήμερα	MuPAD http://www.mupad.de http://www.sciface.com	Γενικού σκοπού ΥΣΑ
1991 - σήμερα	AXIOM http://www.nag.co.uk	Ο διάδοχος του Stratchpad. Γενικού σκοπού ΥΣΑ, το οποίο επιτρέπει τους χρήστες να γράφουν αλγορίθμους πάνω σε γενικά πεδία ορισμού
	CAYLEY	Θεωρία ομάδων
Τέλη 1980's	MAGMA http://www.maths.usyd.edu.au:8000/u/magma	Γενικού σκοπού ΥΣΑ για Άλγεβρα, Θεωρία Αριθμών, Αλγεβρική Γεωμετρία, Αλγεβρική Τοπολογία, Αλγεβρική Συνδιαστική κ.α..
1986-1997	GAP, GAP 2 (2000) http://www-gap.dcs.st-and.ac.uk/~gap	Ομάδες, αλγόριθμοι και προγραμματισμός, υπολογιστική διακριτή άλγεβρα
	FORM	Υπολογισμοί σε Φυσική Υψηλών Ενεργειών
1990-1996	LiE http://www.mathlabo.univ-poitiers.fr/~maavl/LiE	Υπολογισμούς σε Lie άλγεβρα
1992	MACAULAY 2 http://www.math.uiuc.edu/Macaulay2	Αλγεβρική Γεωμετρία και Μεταθετική Άλγεβρα
Μέσα 1980's - 2000	PARI ftp://megrez.math.u-bordeaux.fr/pub/pari	Θεωρία Αριθμών

Τα κύρια χαρακτηριστικά των ΥΣΑ είναι συνήθως τα παρακάτω :

α) Ακριβείς υπολογισμοί.

Όταν λύνουμε ένα υπολογιστικό πρόβλημα στον υπολογιστή, η ακρίβεια της λύσης εξαρτάται από τρεις βασικές ιδιότητες : α) την παράσταση των αριθμών στην μηχανή – την μέθοδο στρογγυλοποίησης και το εύρος των αριθμών, β) το υπολογιστικό πρόβλημα – την ευαισθησία των λύσεων σε σχετικές αλλαγές των δεδομένων, και γ) τον υπολογιστικό αλγόριθμο – την αριθμητική ευστάθεια του αλγορίθμου. Ένας τυπικός υπολογιστής μπορεί να χρησιμοποιεί έναν συγκεκριμένο αριθμό bits για την αποθήκευση αριθμών ο οποίος είναι τυπικά 8, 16, 32, 36, 48 ή 64 bits. Αυτός ο περιορισμός της αναπαράστασης των αριθμών έχει άμεση συνέπεια στην ακρίβεια και στο μέγεθος των αριθμών που χρησιμοποιούμε σε αριθμητικούς υπολογισμούς και δεν είναι ικανός για τον χειρισμό συμβολικών υπολογισμών.

Παράδειγμα. Αν προσπαθήσουμε να υπολογίσουμε τον πρώτο αριθμό Fibonacci που διαιρείται με το 100 στο περιβάλλον της Fortran 95 θα έχουμε

```
program fibonacci
implicit none ! Variables
INTEGER*4::f, f1, f2, i ! Body of fibonacci
f1 = 1;     f2 = 1;     i = 3; f = f1 + f2;
Do While (Mod (f, 100) /= 0)
f1 = f2; f2 = f; f = f1 + f2; i = i + 1
End Do
Print *, f, i
end program fibonacci
```

Αποτελέσματα εκτέλεσης του προγράμματος

```
 708252800      96
Press any key to continue
```

Αν αλλάξουμε το εύρος των τιμών των ακεραίων f, f1,f2,i από Integer*4 (μέγιστος ακέραιος $2^{31}-1$), σε Integer*8 (μέγιστος ακέραιος $2^{63}-1$) θα πάρουμε διαφορετικό αποτέλεσμα :

```
 8284360270132553400          522
Press any key to continue
```

Και στις 2 παραπάνω περιπτώσεις ο προγραμματιστής μπορεί εύκολα να ξεγελαστεί μιας και οι δύο αριθμοί έχουν στο τέλος δύο μηδενικά που σημαίνει ότι διαιρούνται με 100, ενώ το πρόγραμμα λογικά φαίνεται σωστό. Με την ίδια προγραμματιστική λογική το πρόβλημα λύνεται στο ΥΣΑ του Mathematica δίνοντας ως αποτέλεσμα :

```
f1 = 1;
f2 = 1;
i = 3; f = f1 + f2;
While[Mod[f, 100] ≠ 0,
  f1 = f2;
  f2 = f;
  f = f1 + f2;
  ++i];
f
i

996921667718930338621440576020 0

150
```

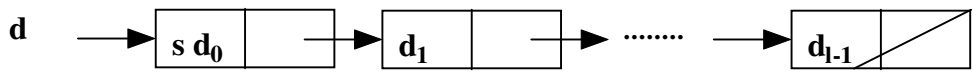
Παρατηρούμε δηλαδή ότι ο $150^{ος}$ όρος της ακολουθίας Fibonacci διαιρείται με το 100, ενώ ο συγκεκριμένος αριθμός αποτελείται από 31 ψηφία, και δεν έχει καμιά σχέση με το αποτέλεσμα που πήραμε στο περιβάλλον της Fortran 95. Ολόγος για τον οποίο το πρόβλημα αυτό δεν λύθηκε στο περιβάλλον της Fortran 95 είναι το περιορισμένο εύρος ψηφίων που χρησιμοποιεί το περιβάλλον της Fortran 95 για την αναπαράσταση των ακεραίων (4 bytes (δηλ. 32 bits) ή 8 bytes (δηλ. 64bits)). ■

Παράδειγμα. Παρακάτω βλέπουμε την χρήση της πλήρους ακρίβειας σε πράξεις στο ΥΣΑ του Mathematica

```
In[1]:= 30! / 2^26 + 20^20

Out[1]= 10881017562119053391570312 5

In[2]:= 30! / (2^30 - 1)

                38276025946925116686336000000 0
Out[2]= ------------------------------------------
                          1549411
```

■

Τα ΥΣΑ χρησιμοποιούν διαφορετικές τεχνικές αναπαράστασης ακεραίων και ρητών αριθμών στην μνήμη του υπολογιστή και για το λόγο αυτό δεν αντιμετωπίζουν προβλήματα που έχουν να κάνουν με την ακρίβεια και το μέγεθος των αριθμών. Ο μεγαλύτερος ακέραιος που μπορεί για παράδειγμα να αναπαρασταθεί σε έναν υπολογιστή που χρησιμοποιεί λέξεις (word) των 32 bits είναι $2^{31}-1=2147483647$ ενώ σε ένα ΥΣΑ κάθε ακέραιος $d = s\sum_{i=0}^{l-1} d_i b^i$ μπορεί να αναπαρασταθεί από μια συνδεόμενη λίστα της μορφής :



ή από έναν δυναμικό πίνακα :

$$\begin{bmatrix} sl & d_0 & d_1 & \cdots & d_{l-1} \end{bmatrix}$$

και συνεπώς το μέγεθος του εξαρτάται από την συνολική μνήμη του υπολογιστή και όχι από το εύρος των λέξεων. Παρόμοια ένας ρητός αριθμός μπορεί να αναπαρασταθεί από έναν κόμβο :

LINK1	LINK2

όπου το LINK1 είναι ένας δείκτης που οδηγεί στον αριθμητή (λίστα ή δυναμικός πίνακας) και παρόμοια το LINK2 είναι δείκτης που οδηγεί στον παρονομαστή. Συνεπώς το βασικό πλεονέκτημα των ΥΣΑ είναι ότι μπορούν να χρησιμοποιήσουν όποια ακρίβεια θέλουμε. Από την άλλη πλευρά αυτός ο τρόπος αναπαράστασης αριθμών δημιουργεί 2 μειονεκτήματα : α) χρειαζόμαστε περισσότερη μνήμη για την αποθήκευση των αριθμών αυτών, και β) χρειαζόμαστε περισσότερη ταχύτητα για την επεξεργασία των αριθμών αυτών. Αυτά τα 2 μειονεκτήματα κάνουν τα ΥΣΑ όχι κατάλληλα για επίλυση προβλημάτων μεγάλης κλίμακας, επειδή απαιτούν μνήμη και χρόνο. Παρόλα αυτά πολλές προσπάθειες έχουν γίνει στις ημέρες μας για την επίλυση των προβλημάτων αυτών με την συνεχή έρευνα για νέους και πιο αποδοτικούς αλγόριθμους. Κάνοντας χρήση της ακρίβειας που εμείς επιθυμούμε μπορούμε επίσης να αποφύγουμε προβλήματα αριθμητικής ευστάθειας (numerical stability) τα οποία συνήθως συναντούμε σε αριθμητικά πακέτα προγραμμάτων.

β) διαδραστικότητα (interactivity)

Ενώ παλαιότερα έπρεπε να γράψεις πρόγραμμα για την επίλυση ενός μαθηματικού προβλήματος, τώρα τα περισσότερα προγράμματα έχουν την μορφή συναρτήσεων που καλείς με κατάλληλη επιλογή των ορισμάτων, παίρνοντας απευθείας απάντηση.

Παράδειγμα. Στις παρακάτω 3 εντολές : α) δίνουμε τα στοιχεία ενός πίνακα A, β) εμφανίζουμε τον πίνακα, και γ) ζητούμε τις ιδιοτιμές του πίνακα.

```
In[1]:= A = {{1, 2, 3}, {2, 3, 1}, {3, 2, 1}}

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

In[2]:= MatrixForm[A]

Out[2]//MatrixForm=
        1  2  3
        2  3  1
        3  2  1

In[3]:= Eigenvalues[A]

Out[3]= {6, -2, 1}
```

γ) συμβολικούς υπολογισμούς

Μπορούμε να χειριστούμε σύμβολα με τον ίδιο τρόπο που χειριζόμαστε νούμερα. Μερικοί από τους συμβολικούς υπολογισμούς που συνήθως υποστηρίζονται είναι :

γ1) απλοποιήσεις

```
In[1]:= Simplify[x (x - 2 y) ^3 + y (2 x - y) ^3]

Out[1]= (x - y) (x + y)^3
```

γ2) αλλαγή της μορφής των εκφράσεων,

In[2]:=
```
Expand[ (x - y)  (x + y) ^ 3]
```

Out[2]= $x^4 + 2\,x^3\,y - 2\,x\,y^3 - y^4$

γ3) επίλυση γραμμικών και ορισμένων μη-γραμμικών εξισώσεων,

In[3]:=
```
Solve[x^2 - 5 x + 6 == 0, x]
```

Out[3]= $\{\{x \to 2\}, \{x \to 3\}\}$

In[4]:=
```
Reduce[a * x + b == 0, x]
```

Out[4]= $(b == 0 \&\& a == 0) \,||\, \left(a \neq 0 \&\& x == -\frac{b}{a}\right)$

In[5]:=
```
Solve[x^2 - b * x + 6 == 0, x]
```

Out[5]= $\left\{\left\{x \to \frac{1}{2}\left(b - \sqrt{-24 + b^2}\right)\right\}, \left\{x \to \frac{1}{2}\left(b + \sqrt{-24 + b^2}\right)\right\}\right\}$

In[6]:=
```
Solve[{
    x1 - x2 + 2 * x3 + x4 == -2,
    -2 * x1 + x2 - 3 * x3 - 5 * x4 == 4,
    x1 - x2 + x3 + 6 * x4 == 0,
    2 * x1 + 3 * x2 + 5 * x3 - 7 * x4 == 1}, {x1, x2, x3, x4}]
```

Out[6]= $\left\{\left\{x1 \to -\frac{51}{2},\ x2 \to \frac{11}{3},\ x3 \to \frac{73}{6},\ x4 \to \frac{17}{6}\right\}\right\}$

In[7]:=
```
Solve[{
    x1 + x2 - x3 + x5 == 1,
    -x1 - x2 + 2 * x3 - x4 + x5 == 2,
    2 * x1 + 2 * x2 - 3 * x3 + 3 * x4 + x5 == 0,
    x1 + x2 - 3 * x4 + 2 * x5 == 3}, {x1, x2, x3, x4, x5}]
```

Solve :: svars : Equations may not give solutions for all "solve " variables . More...

Out[7]= $\left\{\left\{x1 \to \frac{9}{2} - x2 - \frac{7\,x5}{2},\ x3 \to \frac{7}{2} - \frac{5\,x5}{2},\ x4 \to \frac{1}{2} - \frac{x5}{2}\right\}\right\}$

γ4) πράξεις πινάκων,

In[8]:=
```
A = {
    {1, 1, 1},
    {a, b, c},
    {a^2, b^2, c^2}}
```

Out[8]= $\{\{1, 1, 1\}, \{a, b, c\}, \{a^2, b^2, c^2\}\}$

In[9]:=
```
Det[A]
```

Out[9]= $-a^2\,b + a\,b^2 + a^2\,c - b^2\,c - a\,c^2 + b\,c^2$

In[10]:=
```
Inverse[A]
```

Out[10]= $\left\{\left\{\frac{-b^2\,c + b\,c^2}{-a^2\,b + a\,b^2 + a^2\,c - b^2\,c - a\,c^2 + b\,c^2}, \frac{b^2 - c^2}{-a^2\,b + a\,b^2 + a^2\,c - b^2\,c - a\,c^2 + b\,c^2}, \frac{-b + c}{-a^2\,b + a\,b^2 + a^2\,c - b^2\,c - a\,c^2 + b\,c^2}\right\},\right.$
$\left\{\frac{a^2\,c - a\,c^2}{-a^2\,b + a\,b^2 + a^2\,c - b^2\,c - a\,c^2 + b\,c^2}, \frac{-a^2 + c^2}{-a^2\,b + a\,b^2 + a^2\,c - b^2\,c - a\,c^2 + b\,c^2}, \frac{a - c}{-a^2\,b + a\,b^2 + a^2\,c - b^2\,c - a\,c^2 + b\,c^2}\right\},$
$\left.\left\{\frac{-a^2\,b + a\,b^2}{-a^2\,b + a\,b^2 + a^2\,c - b^2\,c - a\,c^2 + b\,c^2}, \frac{a^2 - b^2}{-a^2\,b + a\,b^2 + a^2\,c - b^2\,c - a\,c^2 + b\,c^2}, \frac{-a + b}{-a^2\,b + a\,b^2 + a^2\,c - b^2\,c - a\,c^2 + b\,c^2}\right\}\right\}$

γ5) υπολογισμός ορίων,

In[11]:= `Limit[Sin[x]/x, x → 0]`

Out[11]= 1

In[12]:= `Limit[(√(1+x) - √(1-x))/x, x → 0]`

Out[12]= 1

In[13]:= `Limit[√(x^2 - 6x + 9)/(x - 3), x → 3, Direction → -1]`

Out[13]= 1

In[14]:= `Limit[√(x^2 - 6x + 9)/(x - 3), x → 3, Direction → 1]`

Out[14]= -1

γ6) υπολογισμός σειρών όπως αθροίσματος και γινομένου,

In[15]:= `Sum[1/k^2, {k, 1, Infinity}]`

Out[15]= $\frac{\pi^2}{6}$

In[16]:= `Sum[1/k, {k, 1, Infinity}]`

Sum::div : Sum does not converge . More...

Sum::div : Sum does not converge . More...

Out[16]= $\sum_{k=1}^{\infty} \frac{1}{k}$

γ7) διαφόριση συναρτήσεων,

In[17]:= `D[x Sin[x], x]`

Out[17]= x Cos[x] + Sin[x]

In[18]:= `D[x^n, {x, 4}]`

Out[18]= (-3 + n) (-2 + n) (-1 + n) n x^(-4+n)

In[19]:= $\partial_x \left(x^2 - 5x + \frac{6}{x-1}\right)$

Out[19]= $-5 - \frac{6}{(-1+x)^2} + 2x$

In[20]:= `D[x Sin[x + y], {x, 1}, {y, 2}]`

Out[20]= -x Cos[x + y] - Sin[x + y]

γ8) ολοκλήρωση συναρτήσεων,

In[21]:= $\int \frac{x-1}{x^2-5x+6}\, dx$

Out[21]= 2 Log[-3 + x] - Log[-2 + x]

In[22]:= $\int_0^1 \frac{x}{\sqrt{1+x^2-x^4}}\, dx$

Out[22]= ArcCot[2]

γ9) επίλυση διαφορικών εξισώσεων και εξισώσεων διαφορών,

```
In[23]:= DSolve[y''[x] == a y'[x] + y[x], y, x]
```

$$\text{Out[23]= } \left\{\left\{y \to \text{Function}\left[\{x\},\ e^{\frac{1}{2}\left(a-\sqrt{4+a^2}\right)x}\, C[1] + e^{\frac{1}{2}\left(a+\sqrt{4+a^2}\right)x}\, C[2]\right]\right\}\right\}$$

```
In[24]:= DSolve[{y[x] == -z'[x], z[x] == -y'[x]}, {y[x], z[x]}, x]
```

$$\text{Out[24]= } \left\{\left\{z[x] \to \frac{1}{2} e^{-x}\left(1+e^{2x}\right) C[1] - \frac{1}{2} e^{-x}\left(-1+e^{2x}\right) C[2],\ y[x] \to -\frac{1}{2} e^{-x}\left(-1+e^{2x}\right) C[1] + \frac{1}{2} e^{-x}\left(1+e^{2x}\right) C[2]\right\}\right\}$$

```
In[25]:= RSolve[{F[n+2] == F[n+1] + F[n], F[1] == 1, F[2] == 1}, F[n], n] // FullSimplify
```

$$\text{Out[25]= } \left\{\left\{F[n] \to \frac{-\left(\frac{1}{2}\left(1-\sqrt{5}\right)\right)^n + \left(\frac{1}{2}\left(1+\sqrt{5}\right)\right)^n}{\sqrt{5}}\right\}\right\}$$

δ) χρήση επιλεγμένης ακρίβειας

Εκτός από την πλήρη ακρίβεια, μπορούμε να επιλέξουμε εμείς την ακρίβεια με την οποία θέλουμε να εργαστούμε.

```
In[1]:= N[Pi, 100]
```

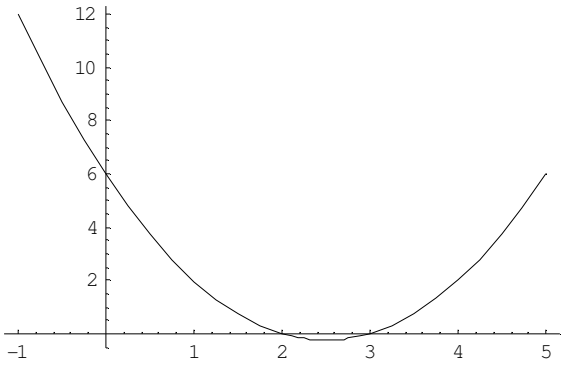
```
Out[1]= 3.14159265358979323846264338327950288419716939937510582097494459230781640628620899862803482534211706 8
```

ε) γραφική παράσταση των αποτελεσμάτων μας

Μπορούμε να δούμε τα αποτελέσματα μας σε διδιάστατα και τριδιάστατα γραφικά (XY plots, XYZ plots, polar plots, log plots κ.α.) και να εξάγουμε τα αποτελέσματα μας σε format που θέλουμε.

ε1) Διδιάστατα γραφικά,

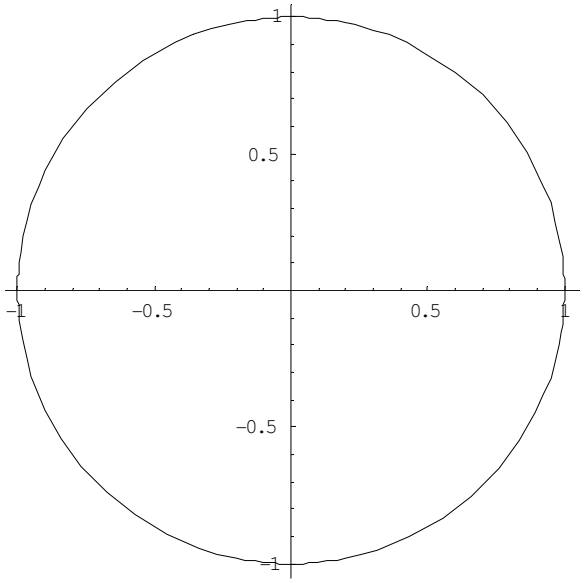
```
In[1]:= Plot[x^2 - 5x + 6, {x, -1, 5}]
```




```
Out[1]= - Graphics -
```

ε2) Γραφική παράσταση παραμετρικών εξισώσεων,

In[2]:= **ParametricPlot[{Cos[t], Sin[t]}, {t, 0, 2 * Pi}, AspectRatio → Automatic]**



Out[2]= **-** Graphics **-**

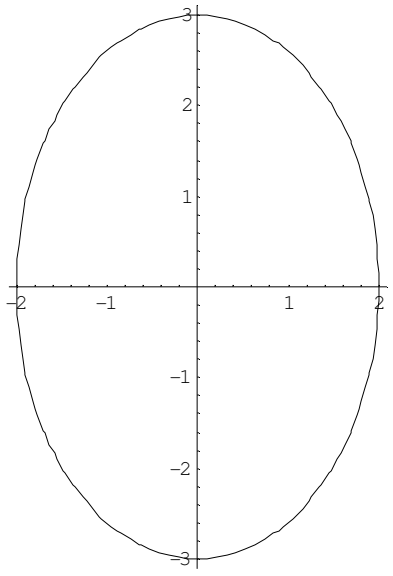
ε3) Επίλυση εξισώσεων και γραφική τους παράσταση,

In[3]:= **<<Graphics`ImplicitPlot`**

In[4]:= **Options[ImplicitPlot]**

Out[4]= {AspectRatio → Automatic, Axes → Automatic, AxesLabel → None, AxesOrigin → Automatic,
AxesStyle → Automatic, Background → Automatic, ColorOutput → Automatic, DefaultColor → Automatic,
Epilog → {}, Frame → False, FrameLabel → None, FrameStyle → Automatic, FrameTicks → Automatic,
GridLines → None, PlotLabel → None, PlotPoints → 39, PlotRange → Automatic, PlotRegion → Automatic,
PlotStyle → Automatic, Prolog → {}, RotateLabel → True, Ticks → Automatic, DefaultFont :→ $DefaultFont,
DisplayFunction :→ $DisplayFunction, FormatType :→ $FormatType, TextStyle :→ $TextStyle, ImageSize → Automatic}

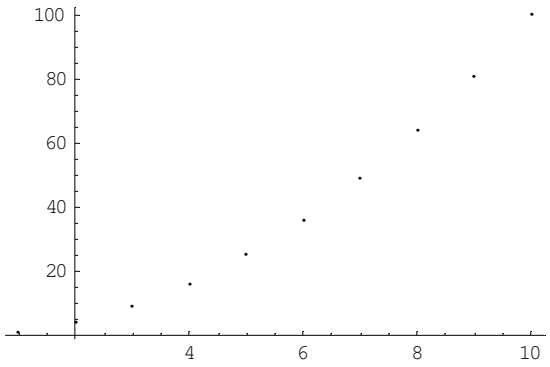
In[5]:= **ImplicitPlot[x^2/4 + y^2/9 == 1, {x, −2, 2}, {y, −3, 3}, AxesOrigin → {0, 0}]**



Out[5]= **-** ContourGraphics **-**

ε4) Γραφική παράσταση σημείων,

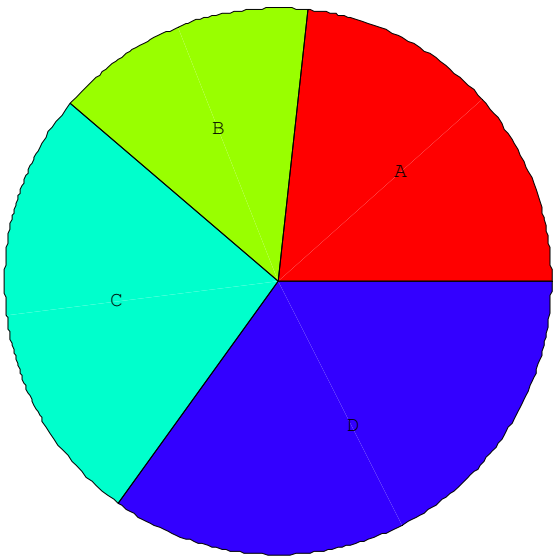
```
In[6]:= Table[{x, x^2}, {x, 1, 10}]

Out[6]= {{1, 1}, {2, 4}, {3, 9}, {4, 16}, {5, 25}, {6, 36}, {7, 49}, {8, 64}, {9, 81}, {10, 100}}

In[7]:= ListPlot[%]
```




```
Out[7]= - Graphics -
```

ε5) Στατιστικά διαγράμματα,

```
In[8]:= << Graphics`

In[9]:= p = {30, 20, 34, 45}

Out[9]= {30, 20, 34, 45}

In[10]:= PieChart[p, PieLabels→ {A, B, C, D}]
```




```
Out[10]= - Graphics -
```

ε6) Κινούμενες γραφικές παραστάσεις,

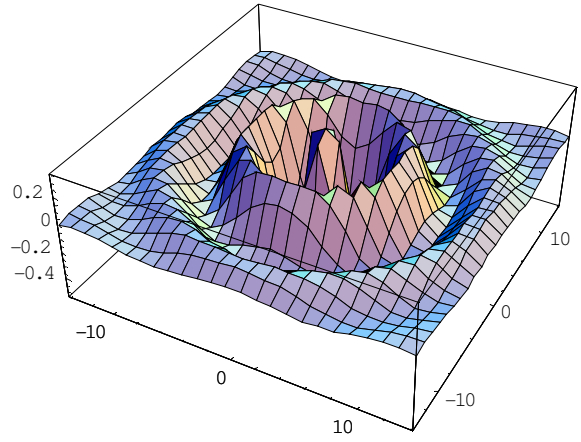
```
In[11]:= << Graphics`Animation`

In[12]:= Animate[Plot[a * x^2 - 5 x + 6, {x, -10, 10}, PlotRange→ {-15, 15}], {a, -2, 2, 0.1}]
```

Η παραπάνω εντολή σχεδιάζει την γραφική παράσταση της συνάρτησης $ax^2 - 5x + 6$ στο διάστημα [-10,10] για τις τιμές του a από -2 έως και 2 με βήμα 0.1. Μπορούμε να δούμε όλες τις γραφικές παραστάσεις μαζί να

εναλλάσσονται σε ένα παράθυρο, ώστε να δούμε την επίδραση που έχει το *a* στην γραφική παράσταση της καμπύλης.

ε7) τριδιάστατα γραφικά

```
In[13]:= Plot3D[E^(-0.2 Sqrt[x^2+y^2]) Cos[Sqrt[x^2+y^2]], {x, -9Pi/2, 9Pi/2}, {y, -9Pi/2, 9Pi/2}]
```




```
Out[13]= - SurfaceGraphics -
```

στ) βιβλιοθήκη συναρτήσεων,

Έχουν ένα μεγάλο πλήθος συναρτήσεων που καλύπτουν γενικές αλλά και πολύ ειδικές περιοχές των μαθηματικών.

ζ) γλώσσα προγραμματισμού,

Πολλά ΥΣΑ διαθέτουν την δικιά τους γλώσσα προγραμματισμού, στην οποία μπορούμε να δημιουργήσουμε τις δικές μας συναρτήσεις. Η γλώσσα προγραμματισμού για παράδειγμα που διαθέτει το Mathematica μας δίνει την δυνατότητα να συνδυάσουμε διάφορες μεθοδολογίες προγραμματισμού όπως ο διαδικασιακός προγραμματισμός (procedural programming), ο συναρτησιακός προγραμματισμός (functional programming), και ο κανονοκεντρικός προγραμματισμός (rule-based programming).

Μπορούμε για παράδειγμα να δώσουμε μέσω κανόνων (κανονοκεντρικός προγραμματισμός) την διαδικασία της παραγώγισης συναρτήσεων :

```
In[1]:= diff[a_?NumberQ, x_] := 0 (* diff[a_,x_]:=0/;FreeQ[a,x] *)

In[2]:= diff[x_^n_Integer:1, x_] := n*x^(n-1)

In[3]:= diff[a_^c_?IntegerQ, x_] := c*a^(c-1)*diff[a, x]

In[4]:= diff[a_+b_, x] := diff[a, x] + diff[b, x]

In[5]:= diff[a_-b_, x] := diff[a, x] - diff[b, x]

In[6]:= diff[a_b_, x_] := diff[a, x]*b + a*diff[b, x]

In[7]:= diff[a_/b_, x_] := (diff[a, x]*b - a*diff[b, x])/b^2
```

και στη συνέχεια να εφαρμόσουμε τους κανόνες αυτούς για τον υπολογισμό των παραγώγων μιας πολυωνυμικής ή ρητής συνάρτησης :

```
In[8]:= Diff[x^3 - 3 x^2 + 6 x - 2, x]

Out[8]= 6 - 6 x + 3 x^2

In[9]:= Diff[(x^3 - 3 x^2 + 6 x - 2)/(x^2 - 1), x]

Out[9]= ((-1 + x^2) (6 - 6 x + 3 x^2) - 2 x (-2 + 6 x - 3 x^2 + x^3))/(-1 + x^2)^2
```

Τα ΥΣΑ μπορούν να χρησιμοποιηθούν στην έρευνα :

α) για τον έλεγχο εικασιών – για να υποστηρίξουν αλλά και να απορρίψουν εικασίες,
β) για να φέρουμε σε πέρας, χωρίς λάθη, ένα μεγάλο πλήθος συμβολικών υπολογισμών που θα απαιτήσει ένας νέος αλγόριθμος,
γ) για να σχεδιάσουμε και να δημιουργήσουμε ένα νέο ΥΣΑ για ένα νέο ερευνητικό πεδίο,
δ) για να προσαρμόσουμε και να βελτιώσουμε τους αλγορίθμους που έχουμε δημιουργήσει για την επίλυση ενός προβλήματος,
ε) για να πάρουμε κλειστές φόρμουλες λύσεων σε μαθηματικά προβλήματα, οι οποίες θα μας δώσουν μια βαθύτερη γνώση για το ίδιο το πρόβλημα,
στ) για να δημιουργήσουμε μαθηματικούς πίνακες όπως πίνακες ολοκληρωμάτων, παραγώγων, ειδικών συναρτήσεων κ.λ.π.,
ζ) για να ανακαλύψουμε νέους αλγορίθμους και τεχνικές,
η) για να έχουμε χρόνο να συγκεντρωθούμε στην κατανόηση των μαθηματικών εννοιών και όχι στους υπολογισμούς.

Παράδειγμα. Ας δούμε αν ισχύει η παρακάτω εικασία :

«Ο αριθμός $2^p - 1$ είναι πρώτος αριθμός αν ο p είναι πρώτος αριθμός.»

Είναι εύκολο να δούμε ότι για p=2,3 η εικασία αυτή ισχύει. Μπορούμε εύκολα να ελέγξουμε μέσω ενός εύκολου προγράμματος στο Mathematica αν ισχύει η εικασία αυτή για όλα τα p.

```
In[1]:= i = 1;
        While[PrimeQ[2^Prime[i] - 1],
          ++i]
        Print[i, "th prime=", Prime[i]]

        5th prime=11
```

Είναι φανερό ότι για p=11 δεν ισχύει η εικασία που κάναμε.

```
In[4]:= FactorInteger[2^11 - 1]

Out[4]= {{23, 1}, {89, 1}}
```

μιας και $2^{11} - 1 = 23^1 \times 89^1$. ■

Παράδειγμα. Ας υποθέσουμε ότι θέλουμε να υπολογίσουμε την ορίζουσα VanDermonde

$$D\left(x_1, x_2, ..., x_n\right) = \begin{vmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ 1 & x_3 & x_3^2 & \cdots & x_3^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} \end{vmatrix}$$

Τότε μπορούμε εύκολα να πειραματιστούμε στο Mathematica για να υπολογίσουμε την πιθανή μορφή που θα έχει η ορίζουσα πριν προχωρήσουμε στην απόδειξη π.χ.

```
In[1]:= A={{1,x1,x1^2},{1,x2,x2^2},{1,x3,x3^2}};

In[2]:= Factor[Det[A]]

Out[2]= -((x1 - x2) (x1 - x3) (x2 - x3))

In[3]:= A={{1,x1,x1^2,x1^3},{1,x2,x2^2,x2^3},{1,x3,x3^2,x3^3},{1,x4,x4^2,x4^3}};

In[4]:=  Factor[Det[A]]

Out[4]= (x1 - x2) (x1 - x3) (x2 - x3) (x1 - x4) (x2 - x4) (x3 - x4)
```

Είναι φανερό από τα παραπάνω ότι αυτό που θέλουμε να αποδείξουμε πιθανώς θα έχει την μορφή :

$$D\left(x_1, x_2, ..., x_n\right) = \left(-1\right)^n \prod_{\substack{i,j \\ i<j}} \left(x_i - x_j\right) = \prod_{\substack{i,j \\ i>j}} \left(x_i - x_j\right) \qquad \blacksquare$$

Τα ΥΣΑ εκτός από τα πλεονεκτήματα που έχουν συνοδεύονται και από μειονεκτήματα όπως :
α) την δυσκολία ορισμού του πεδίου λύσεων στο οποίο αναζητούμε λύσεις,
β) έχουν ιδιαιτερότητες που μαθαίνονται μόνο με την εμπειρία,
γ) δεν καλύπτουν όλα τα υπάρχοντα επιστημονικά πεδία,
δ) δεν δίνουν ακριβείς λύσεις σε προβλήματα για τα οποία δεν υπάρχει ακριβής λύση π.χ. λύση πεμπτοβάθμιας εξίσωσης,
ε) δυσκολία διασύνδεσης με άλλες εφαρμογές,
στ) δυσκολία διαχείρισης προβλημάτων μεγάλης κλίμακας λόγω της χαμηλής ταχύτητας και του μεγάλου μεγέθους μνήμης που καταναλώνουν, από τους πόρους του υπολογιστή,
ζ) πολλές φορές δίνουν γενικές απαντήσεις οι οποίες όμως στερούνται κάποιου νοήματος.

Για τους λόγους που αναφέραμε παραπάνω, ένας συνδυασμός αριθμητικών και συμβολικών μεθόδων είναι μια καλή μεθοδολογία για να ελαττώσουμε ταυτόχρονα τις απώλειες σε ακρίβεια, αλλά και το χάσιμο χρόνου.

Σήμερα τα ΥΣΑ χρησιμοποιούνται σχεδόν σε όλα τα Πανεπιστήμια, ενώ η χρήση τους επεκτείνεται και στην βιομηχανία αλλά και σε επαγγελματικές εταιρίες.

Μάλιστα σε πολλές χώρες γίνεται προσπάθεια εισαγωγής των συστημάτων αυτών και στην προπανεπιστημιακή εκπαίδευση. Όπως ο λογαριθμικός κανόνας αντικαταστάθηκε από την αριθμομηχανή, η οποία με την σειρά της αντικαταστάθηκε από τις νέες αριθμομηχανές με γραφικό περιβάλλον, έτσι σύντομα και αυτές οι αριθμομηχανές θα αντικατασταθούν από τα ΥΣΑ τα οποία θα αναβαθμίζονται συνέχεια προσφέροντας όλο και πιο μεγάλη γκάμα συναρτήσεων για επίλυση προβλημάτων σε πολλούς τομείς. Θα επηρεάσουν όμως τα ΥΣΑ τις μαθηματικές δεξιότητες και γνώσεις των μαθητών ; Εδώ υπάρχουν απόψεις που διίστανται. Είναι αυτοί που πιστεύουν ότι η χρήση των ΥΣΑ, θα μειώσουν τις μαθηματικές δεξιότητες των μαθητών, όπως σήμερα ελάχιστοι είναι αυτοί που μπορούν να υπολογίσουν την τετραγωνική ρίζα ενός αριθμού χωρίς αριθμομηχανή. Άλλοι πάλι υποστηρίζουν την άποψη ότι όπως η χρήση της αριθμομηχανής απαιτεί την ύπαρξη μιας εξοικείωσης με τους αριθμούς (ή καλύτερα μια αίσθηση των αριθμών), έτσι και η χρήση των ΥΣΑ απαιτεί την εξοικείωση με την χρήση των συμβόλων στα Μαθηματικά. Σύμφωνα με την τελευταία ομάδα, η χρήση των ΥΣΑ βοηθάει στην ενεργή συμμετοχή των μαθητών στην μάθηση, τους δίνει την δυνατότητα να ασχοληθούν περισσότερο με την κατανόηση των μαθηματικών εννοιών, να πειραματιστούν, να συμμετέχουν στην επίλυση πραγματικών προβλημάτων και να ασχολούνται περισσότερο με την ποιοτική ανάλυση των αποτελεσμάτων, να βλέπουν γραφικά νέες έννοιες, και να αναγνωρίζουν κρυμμένα πρότυπα από την λύση προβλημάτων. Σίγουρα χρειάζεται ένας σωστός σχεδιασμός του τρόπου που θα χρησιμοποιήσουμε τα ΥΣΑ ως εργαλεία εκμάθησης των μαθηματικών εννοιών. Το Τμήμα Μαθηματικών του Α.Π.Θ., προσφέρει τα τελευταία 5 χρόνια το μάθημα επιλογής «Συμβολικές Γλώσσες Προγραμματισμού» στο β' εξάμηνο σπουδών, το οποίο έχει ως κύριο στόχο την εισαγωγή των φοιτητών στα ΥΣΑ και κυρίως στο Mathematica. Έχουν μάλιστα δημιουργηθεί ιστοσελίδες με παρουσιάσεις των μαθημάτων, χρήσιμες συνδέσεις κ.α., για την υποστήριξη των φοιτητών http://anemos.web.auth.gr/mathematica/index4.htm και http://users.auth.gr/~epsom/Symbolic/index.htm . Το μάθημα αυτό έχει ιδιαίτερα μεγάλη προσέλευση από τους φοιτητές. Στο Internet μπορεί εύκολα να βρει κανείς και άλλα Πανεπιστημιακά Τμήματα που ασχολούνται με την Υπολογιστική Άλγεβρα αλλά και τα Υπολογιστικά Συστήματα Άλγεβρας όπως :

Τμήμα Μηχανικών Η/Υ, Τηλεπικοινωνιών και Δικτύων του Πανεπιστημίου Θεσσαλίας
http://www.inf.uth.gr/greek/coursedesc_308.htm
http://www.inf.uth.gr/greek/coursedesc_408.htm
Τμήμα Πληροφορικής και Τηλεπικοινωνιών του Καποδιστριακού Πανεπιστημίου Αθηνών
http://eclass.di.uoa.gr/D231/
Τμήμα Μαθηματικών του Πανεπιστημίου Ιωαννίνων
http://www.math.uoi.gr/~nglinos/sm/ma644.html
Τμήμα Μαθηματικών του Πανεπιστημίου Κρήτης
http://www.math.uoc.gr/~marios/sy03/

Πληροφορίες σχετικές με την υπολογιστική άλγεβρα και τα ΥΣΑ, όπως περιοδικά που ασχολούνται με το θέμα αυτό (ACM Communications in Computer Algebra, Journal of Symbolic Computation), συνέδρια (International Symposium on Symbolic and Algebraic Computation (ISSAC)), ομάδες που ασχολούνται με το θέμα αυτό και άλλους ενδιαφέροντες συνδέσμους μπορεί να βρει κανείς στην ιστοσελίδα της ειδικής ομάδας της Association for Computing Machinery που ειδικεύεται στο θέμα αυτό http://www.sigsam.org/ .

Τέλος, περιμένουμε στο εγγύς μέλλον, την ραγδαία ανάπτυξη των ΥΣΑ, με την ενσωμάτωση νέων συναρτήσεων που θα καλύπτουν ευρύτερα πεδία έρευνας, την

βελτιστοποίηση των ήδη υπαρχόντων αλγορίθμων, την προτυποποίηση του τρόπου αναπαράστασης των δεδομένων ώστε να είναι δυνατή η επικοινωνία μεταξύ διαφορετικών ΥΣΑ, τη δυνατότητα χρήσης τους μέσω του Internet (ήδη γίνεται), τη δημιουργία βάσης μαθηματικών προβλημάτων στα οποία θα δοκιμάζονται οι επιδόσεις των ΥΣΑ. Θα πρέπει να μην μείνουμε έξω από αυτές τις τεχνολογικές εξελίξεις. Για να γίνει αυτό θα πρέπει οι νέες αυτές τεχνολογίες να ενσωματωθούν στο εκπαιδευτικό μας σύστημα, βάσει πάντα κατάλληλου σχεδιασμού.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] K. O. Geddes, S.R. Czapor and G. Labahn, 1995, *Algorithms for Computer Algebra*, Kluwer Academic Publisher.

[2] J. Grabmeier, E. Kaltofen and V. Weispfenning, 2003, *Computer Algebra Handbook*, Springer-Verlag, Berlin, Heidelberg, New York.

[3] N. P. Karampetakis and A.I. Vardulakis, 2006, Special issue on the use of computer algebra systems for computer aided control system design, International Journal of Control, Vol.79, Issue 11, pp.1313-1320.

[4] R. Mankiewicz, 2002, *Ιστορία των Μαθηματικών*, Εκδόσεις Αλεξάνδρεια.

[5] Γ. Δάσιος, 1999, *Λογισμός μιας μεταβλητής*, Ελληνικό Ανοικτό Πανεπιστήμιο.

[6] Ν.Π. Καραμπετάκης, Σ. Σταματάκης και Ε. Ψωμόπουλος, 2004, *Μαθηματικά και Προγραμματισμός στο Mathematica*, Εκδόσεις Ζήτη.

[7] The MacTutor History of Mathematics archive, http://www-groups.dcs.st-and.ac.uk/~history/ .