

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ, ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΗΥ200: ΕΠΙΣΤΗΜΟΝΙΚΟΣ ΥΠΟΛΟΓΙΣΜΟΣ

ΕΡΓΑΣΙΑ 1: Αριθμητική Γραμμική Άλγεβρα.
(Ημερομηνία Παράδοσης: Κυριακή 10 Απριλίου 2005, (Ώρα: 23:55))
ΟΝΟΜ/ΝΥΜΟ: ΖΥΓΟΥΡΑ ΒΑΪΑ

USERNAME : vezigour AEM : 301

ΑΣΚΗΣΗ 1: Χρήση της συνάρτησης `lu_run` για την επίλυση των τριγωνικών συστημάτων που δίνονται. Παρακάτω δίνεται ο πίνακας των αποτελεσμάτων για τις διάφορες τιμές των n_x , n_y , c_1 , c_2 και c

n_x	n_y	c_1	c_2	c	χρόνος	σφάλμα
3	10	0	0	0	0	$8.2792e - 015$
10	3	0	0	0	0	$1.4444e - 014$
5	5	0	0	0	0	$3.8170e - 015$
5	5	100	0	0	0	$2.7888e - 015$
5	5	0	100	0	0	$2.7888e - 015$
5	5	0	0	100	0	$6.3273e - 015$

ΣΧΟΛΙΟ: Στον πρώτο πίνακα παρατηρούμε τα εξής:

- i) όταν το n_x είναι μικρότερο από το n_y το σφάλμα είναι αρκετά μεγάλο
- ii) όταν το n_x είναι μεγαλύτερο από το n_y τότε το σφάλμα είναι πιο μικρό
- iii) όταν το $n_x = n_y$ τότε βλέπουμε πως για τις διάφορες τιμές των c, c_1, c_2 το σφάλμα από την πρώτη στην δεύτερη περίπτωση μειώνεται ενώ στις επόμενες αυξάνεται σταδιακά.

ΑΣΚΗΣΗ 2: Χρήση της συνάρτησης `lu_run` για την επίλυση των τριγωνικών συστημάτων που δίνονται. Παρακάτω δίνεται ο πίνακας των αποτελεσμάτων για τις διάφορες τιμές των n_x , n_y , c_1 , c_2 και c

n_x	n_y	c_1	c_2	c	χρόνος	σφάλμα
10	10	0	0	0	0.0160	$2.3669e - 014$
10	10	0	0	100	0	$4.9728e - 014$
20	20	0	0	0	0.0780	$2.7895e - 013$
20	20	0	0	100	0.0630	$2.7966e - 013$
30	30	0	0	0	0.5000	$1.0780e - 012$
30	30	0	0	100	0.4690	$1.1489e - 012$
40	40	0	0	0	2.3910	$2.8543e - 012$
40	40	0	0	100	2.3130	$3.0588e - 012$
50	50	0	0	0	8.4690	$7.2805e - 012$
50	50	0	0	100	8.2810	$7.6273e - 012$

ΣΧΟΛΙΟ: Στον παραπάνω πίνακα παρατηρούμε ότι:

- i)για τις διάφορες τιμές των n_x, n_y (10,20,30,40,50) όταν το c από 0 γίνει 100 τότε το σφάλμα αυξάνεται
- ii)ο χρόνος για τις διάφορες τιμές των n_x, n_y (10,20,30,40,50) όταν το c από 0 γίνει 100 μειώνεται

ΑΣΚΗΣΗ 3: Στον παρακάτω πίνακα δίνονται τα αποτελέσματα των 5 συστημάτων με την χρήση της `luinc_run`

	$n_x = n_y$	c	χρόνος	επαναλήψεις	σφάλμα	c	χρόνος	επαναλήψεις	σφάλμα
<i>luinc</i>	10	0	0	1	$2.6472e - 014$	100	0.0160	1	$4.6573e - 014$
	20	0	0.0150	1	$2.7947e - 013$	100	0.0160	1	$2.7947e - 013$
	30	0	0.0320	1	$1.0588e - 012$	100	0.0470	1	$1.1591e - 012$
	40	0	0.1090	1	$2.8510e - 012$	100	0.1100	1	$3.0609e - 012$
	50	0	0.2190	1	$7.4437e - 012$	100	0.2190	1	$7.6968e - 012$

- ΣΧΟΛΙΟ:** Παρατηρούμε ότι για τις διάφορες τιμές(10,20,30,40,50) όταν $n_x=n_y$ και το c από 0 γίνει 100
- i)ο χρόνος εκτέλεσης αυξάνεται ελάχιστα και
 - ii)το σφάλμα έχει μια μικρή αύξηση

ΑΣΚΗΣΗ 4: Στον παρακάτω πίνακα δίνονται τα αποτελέσματα των 5 συστημάτων με την χρήση της `jacobi_run`

	$n_x = n_y$	c	χρόνος	επαναλήψεις	σφάλμα	c	χρόνος	επαναλήψεις	σφάλμα
<i>Jacobi</i>	10	0	0	100	0.1149	100	0	100	0.0160
	20	0	0.6870	400	0.1541	100	0.2500	139	0.0082
	30	0	15.5780	900	0.1986	100	17.7500	900	0.1986
	40	0	97.3600	1600	0.2445	100	29.7180	480	0.0334
	50	0	355.7340	2500	0.2909	100	107.6400	722	0.0517

ΣΧΟΛΙΟ: Παρατηρούμε ότι για τις διάφορες τιμές(10,20,30,40,50) όταν $n_x=n_y$ και το c από 0 γίνει 100
i)ο χρόνος εκτέλεσης μειώνεται(δραματικά για $n_x=n_y=50$)
ii)το σφάλμα μειώνεται επίσης και
iii)ο αριθμός των επαναλήψεων μειώνεται αισθητά(π.χ από 2500->722)

ΑΣΚΗΣΗ 5: Στον παρακάτω πίνακα δίνονται τα αποτελέσματα των 5 συστημάτων με την χρήση της `gs_run`

	$n_x = n_y$	c	χρόνος	επαναλήψεις	σφάλμα	c	χρόνος	επαναλήψεις	σφάλμα
<i>G - S</i>	10	0	0.0630	100	0.0020	100	0.0310	25	$9.9812e - 004$
	20	0	2.1090	360	0.0043	100	0.4370	77	0.0038
	30	0	27.3910	746	0.0095	100	5.9850	156	0.0092
	40	0	156.1560	1257	0.0168	100	31.9070	262	0.0163
	50	0	601.3430	1888	0.0259	100	122.7810	393	0.0258

ΣΧΟΛΙΟ: Παρατηρούμε ότι για τις διάφορες τιμές(10,20,30,40,50) όταν $n_x=n_y$ και το c από 0 γίνει 100
i)ο χρόνος εκτέλεσης μειώνεται δραματικά
ii)το σφάλμα μειώνεται ελάχιστα(εκτός της πρώτης που έχουμε πολύ μεγάλη μείωση) και
iii)ο αριθμός των επαναλήψεων μειώνεται με πολύ γρήγορο ρυθμό.

ΑΣΚΗΣΗ 6: Στον παρακάτω πίνακα δίνονται τα αποτελέσματα των 5 συστημάτων με την χρήση της `cg_run`

	$n_x = n_y$	c	χρόνος	επαναλήψεις	σφάλμα	c	χρόνος	επαναλήψεις	σφάλμα
<i>CG</i>	10	0	0	14	$2.5440e - 007$	100	0.0470	14	$2.5440e - 007$
	20	0	0.1250	29	$2.8347e - 006$	100	0.0780	24	$2.7685e - 006$
	30	0	1.4070	44	$3.6850e - 006$	100	1.1570	36	$4.4713e - 006$
	40	0	7.0150	59	$3.9598e - 006$	100	5.6720	49	$3.7638e - 006$
	50	0	22.0930	75	$4.2605e - 006$	100	18.7500	62	$3.7857e - 006$

ΣΧΟΛΙΟ: Παρατηρούμε ότι για τις διάφορες τιμές(10,20,30,40,50) όταν $n_x=n_y$ και το c από 0 γίνει 100
i)ο χρόνος εκτέλεσης μειώνεται ελάχιστα(εκτός απο την πρώτη που αυξάνεται ελάχιστα επίσης)
ii)το σφάλμα μειώνεται (εκτός της περίπτωσης που έχουμε $n_x=n_y=30$ όπου και αυξάνεται σημαντικά)
και τέλος
iii)ο αριθμός των επαναλήψεων μειώνεται ελάχιστα(εκτός της πρώτης που παραμένει ο ίδιος).

ΓΕΝΙΚΟ ΣΧΟΛΙΟ: Παρατηρώντας τις επαναληπτικές μεθόδους `jacobi_run`, `gs_run` και `cg_run` καταλαβαίνουμε πως από τις τρεις η πιο συμφέρουσα είναι `cg` αφού για τα ίδια n_x και n_y έχουμε μικρότερο χρόνο,σφάλμα αλλά και αριθμό επαναλήψεων