

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ, ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
HY200: ΕΠΙΣΤΗΜΟΝΙΚΟΣ ΥΠΟΛΟΓΙΣΜΟΣ

ΕΡΓΑΣΙΑ 1: Αριθμητική Γραμμική Άλγεβρα.

n_x	n_y	c_1	c_2	c	χρόνος	σφάλμα
3	10	0	0	0	0	$8.0005e-015$
10	3	0	0	0	0	$1.6852e-014$
5	5	0	0	0	0	$3.7321e-015$
5	5	100	0	0	0	$2.6899e-015$
5	5	0	100	0	0	$3.2501e-015$
5	5	0	0	100	0	$6.3273e-015$
10	10	0	0	0	0.0150	$2.6232e-014$
20	20	0	0	0	0.0310	$2.5990e-013$
30	30	0	0	0	0.3120	$1.0038e-012$
40	40	0	0	0	1.6560	$2.6837e-012$
50	50	0	0	0	6.1720	$7.0978e-012$
10	10	0	0	100	0	$4.9728e-014$
20	20	0	0	100	0.0410	$2.5738e-013$
30	30	0	0	100	2.970	$1.0141e-012$
40	40	0	0	100	1.6250	$2.8267e-012$
50	50	0	0	100	6.1250	$7.4974e-012$

Πίνακας 1. Αποτελέσματα με χρήση της `lu`.

Παρατηρήσεις:

1. Στις δύο πρώτες περιπτώσεις όπου το n_x ή n_y είναι μικρότερο του 5 τότε δεν εμφανίζεται ο πίνακας με τα μη μηδενικά στοιχεία.
2. Όταν το $n_x=n_y$ η λύση είναι συμμετρική.
3. Όταν βάζουμε συντελεστή στο c_1 ή c_2 τότε το διάγραμμα της λύσης έχει τρία ακρότατα και όχι ένα.
4. Αντιθέτως όταν έχουμε συντελεστή μόνο στο c τότε έχουμε ένα ακρότατο. Παρατηρούμε επίσης ότι όταν αυτό έχει την τιμή 100 τότε το σημείο αυτό είναι πιο ψηλά απ'ότι όταν είναι 0.

5. Καθώς αυξάνεται ο αριθμός των n_x και n_y ο χρόνος μεγαλώνει ενώ το σφάλμα μειώνεται. Βλέπουμε όμως ότι όταν δώσουμε πολλά σημεία (πέραν των 30) ενώ ο χρόνος επίσης μεγαλώνει το σφάλμα μεγαλώνει. Άρα δε δουλεύει καλά για πολλά σημεία.

Ασκήσεις 3, 4, 5 και 6:

	$n_x = n_y$	c	χρόνος	επαν.	σφάλμα	c	χρόνος	επαν.	σφάλμα
<i>luinc</i>	10	0	0		$2.6472e - 014$	100	0		$4.6573e - 014$
	20	0	0		$2.8339e - 013$	100	0		$2.7947e - 013$
	30	0	0.0310		$1.0588e - 012$	100	0.047		$1.1591e - 012$
	40	0	0.0790		$2.8510e - 012$	100	0.0620		$3.0609e - 012$
	50	0	0.1400		$7.4437e - 012$	100	0.1250		$7.6968e - 012$
<i>Jacobi</i>	10	0	0	100	$2.4734e - 004$	100	0	43	$4.9531e - 006$
	20	0	0.2960	400	$8.8342e - 005$	100	0.1250	139	$4.7231e - 006$
	30	0	3.2810	900	$5.1935e - 005$	100	1.0780	285	$4.9409e - 006$
	40	0	18.3750	1600	$3.6468e - 005$	100	6.6250	480	$4.9761e - 006$
	50	0	71.1250	2500	$2.8015e - 005$	100	20.5310	722	$4.9764e - 006$
<i>G - S</i>	10	0	0.0160	100	$8.4316e - 006$	100	0	25	$4.3281e - 006$
	20	0	0.5930	360	$4.9394e - 006$	100	0.1400	77	$4.3936e - 006$
	30	0	5.7810	746	$4.9818e - 006$	100	1.2500	156	$4.7805e - 006$
	40	0	29.4850	1257	$4.9941e - 006$	100	6.2190	262	$4.8521e - 006$
	50	0	108.9380	1888	$4.9846e - 006$	100	23.3280	393	$4.9679e - 006$
<i>CG</i>	10	0	0	14	$2.5440e - 007$	100	0	12	$9.5757e - 007$
	20	0	0.0470	29	$2.8347e - 006$	100	0.0470	24	$2.7685e - 006$
	30	0	0.3280	44	$3.6850e - 006$	100	0.2660	36	$4.4713e - 006$
	40	0	1.3590	59	$3.9598e - 006$	100	1.1570	49	$3.7638e - 006$
	50	0	4.2190	75	$4.2605e - 006$	100	3.4840	62	$3.7857e - 006$

Πίνακας 2. Αποτελέσματα με χρήση λογισμικού για αραιούς πίνακες και επαναληπτικές μεθόδους.

Παρατηρήσεις:

- (a) Στην *luinc* όταν υπάρχει ο συντελεστής $c (= 100)$ τότε έχουμε περίπου τον ίδιο χρόνο αλλά έχουμε μεγαλύτερο σφάλμα. Επιπλέον κάνει λιγότερο χρόνο από την *lu* αφού δεν πέρνει όλο τον πίνακα.

- (b) Όταν το c είναι 0 η jacobi εχαντλεί όλες τις επαναλήψεις ενώ όταν πέρνει την τιμή 100 κάνει λιγότερες και λιγότερο χρόνο. Το σφάλμα είναι μικρότερο όμως παρατηρούμε και εδώ ότι όταν πέρνουμε περισσότερα από 30 στοιχεία το σφάλμα δεν βελτιώνεται.
- (c) Στη μέθοδο C-S στη περίπτωση όπου το c είναι 0 οι επαναλήψεις είναι λιγότερες και το σφάλμα είναι μικρότερο από ότι στη jacobi αλλά ο χρόνος είναι μεγαλύτερος. Για $c=100$ οι επαναλήψεις και ο χρόνος είναι ακόμη λιγότερα αλλά το σφάλμα είναι περίπου το ίδιο.
- (d) Η μέθοδος CG είναι η καλύτερη από όλες καθώς εκτελεί **πολύ** λιγότερες επαναλήψεις σε **πολύ** μικρότερο χρόνο και επιπλέον μειώνεται και το σφάλμα και για τις δύο τιμές του c .
- (e) Σε όλες τις μεθόδους για λίγα στοιχεία (10 ή 20) τότε ο χρόνος είναι μηδενικός.