Ονοματεπώνυμο: .... Α.Μ. ....· Έτος ...

ΕΠΙΣΤΗΜΟΝΙΚΟΣ ΥΠΟΛΟΓΙΣΜΟΣ Ι *Φεβρουάριος 2003 - τπ - αρ*

Διαβάστε προσεκτικά τις εκφωνήσεις. Για την πλήρη αξιολόγηση του γραπτού σας πρέπει να παρουσιάσετε όλο σας τον συλλογισμό και όλα τα ενδιάμεσα αποτελέσματα. Έχετε 3 ώρες. Οι αλγόριθμοι πρέπει να περιγράφονται με σαφήνεια, π.χ. όπως στις σημειώσεις ή με MATLAB .

ΚΑΛΗ ΕΠΙΤΥΧΙΑ!!!

I. *(40 β.)* Να απαντήσετε σύντομα και περιεκτικά στα παρακάτω ερωτήματα (στις ερωτήσεις Σωστό/Λάθος πρέπει να δικαιολογήσετε τις απαντήσεις για να βαθμολογηθείτε).

1) Να γράψετε αλγόριθμο για την παραγοντοποίηση LU ενός μητρείου A για το οποίο γνωρίζουμε μόνον ότι είναι αντιστρέψιμο και τριδιαγώνιο. Ο αλγόριθμος πρέπει να εκμεταλλεύεται όσο γίνεται την ειδική δομή του A και να επιστρέφει τα στοιχεία των L και U στις θέσεις του A.

2) Σωστό ή Λάθος: Αν η πολυπλοκότητα σε αριθμητικές πράξεις του πολλαπλασιασμού δύο τετραγωνικών μητρώων μεγέθους n είναι $O(n^\beta)$ τότε με $O(n^\beta$ πράξεις μπορούμε να λύσουμε ένα σύστημα $Ax = b$ όπου το αντιστρέψιμο $A \in \mathbb{R}^{n\times n}$ επίσης σε $O(n^\beta)$ πράξεις.

3) Σωστό ή Λάθος: Αν ένα μητρείο A είναι κάτω τριγωνικό και αντιστρέψιμο, τότε για να λύσουμε το σύστημα $Ax = b$ δεν υπάρχει λόγος να χρησιμοποιήσουμε οδήγηση.

4) Σωστό ή Λάθος: Κατά την αριθμητική λύση της διαφορικής εξίσωσης $du/dt = f(t, u)$ με αρχική συνθήκη $u(t_0)$, το τοπικό σφάλμα διακριτοποίησης της μεθόδου στο βήμα t_s για $s > 1$ είναι το σφάλμα που προκύπτει από τη μετάδοση του σφάλματος στρογγύλευσης που περιέχεται στο βήμα t_{s-1}.

5) Έστω $A \in \mathbb{R}^{n\times n}$, $x \in \mathbb{R}^n$, $\omega \in \mathbb{R}$, το ταυτοτικό μητρώο I και η πράξη $y = (A - \omega I)x$. Να υπολογίσετε τον ελάχιστο αριθμό, έστω $\mu_{\min}$, μεταφορών ανά πράξη α.κ.υ. και να γράψετε (χρησιμοποιώντας κατάλληλα εμφωλευμένους βρόχους) την υλοποίηση που επιτυγχάνει αυτό το ελάχιστο με κρυφή μνήμη $O(n)$ δηλώνοντας ακριβώς τις εντολές φόρτωσης (LOAD) και εγγραφής (STORE).

II. (*30 β.*) Ο παρακάτω κώδικας MATLAB μπορεί να υλοποιηθεί έτσι ώστε να παράγει τα επιθυμητά αποτελέσματα πολύ πιο γρήγορα. Να εξηγήσετε τι τροποποιήσεις πρέπει να γίνουν και να γράψετε τον ταχύτερο κώδικα, χρησιμοποιώντας, αν χρειάζεται, και άλλες εντολές και συναρτήσεις της MATLAB .

```
for i = 1 : n
    for j = 1 : m
        a(i, j) = 2 * pi * i * cos(j); b(j, i) = a(i, j);
    end
    c(i, 1 : n) = 0;
end
for k = 1 : min(m, n)
    c = c + a(:, k) * b(k, :)
end
```

III. (*30 β.*) Έστω το διάστημα $\Omega = [a, b]$ και ένα πλέγμα από $n + 2$ ισαπέχοντα σημεία από το a ως το b, δηλ. $a = x_0 < x_1 < x_2 < \cdots < x_n < x_{n+1} = b$. Να δείξετε ότι αν η άγνωστη συνάρτηση $u(x)$ είναι ορισμένη στο διάστημα $x \in [a, b]$ και η 4η παράγωγός $u^{(4)}(x)$ είναι συνεχής $\forall x \in [a, b]$, τότε μπορούμε να προσεγγίσουμε την δεύτερη παράγωγο $u''(x)$ σε κάθε σημείο του πλέγματος με ακρίβεια δεύτερης τάξης. (Υπόδειξη: Αρκεί να δείξετε μία προσέγγιση και να αποδείξετε ότι επιτυγχάνει την επιθυμητή ακρίβεια. Επίσης θυμηθείτε ότι μια συναρτηση συνεχής σε ένα κλειστό διάστημα είναι και φραγμένη εκεί).