Επιστημονικός Υπολογισμός 2023-24
Εργαστηριακή Άσκηση

Ημερομηνία κατάθεσης γιά πλήρη βαθμό ως 4/2/2024, 23:59.

- Η εργασία είναι προαιρετική, διατηρείται ως την εξεταστική Σεπτεμβρίου και λαμβάνεται υπόψη μόνον αν λάβετε προβιβάσιμο βαθμό στην τελική εξέταση. Για το 2023, αν δεν παραδοθεί εργασία, ο τελικός βαθμός θα είναι ο βαθμός της γραπτής εξέτασης, χωρίς πολλαπλασιασμό με το συντελεστή 0.7. Αν παραδοθεί εργασία και ο βαθμός της γραπτής εξέτασης είναι προβιβάσιμος, ο τελικός βαθμός θα υπολογιστεί βάσει του τύπου

$$\max\{(\text{βαθμός εξέτασης}) + 0.3 \times (\text{βαθμός άσκησης}), 10\}$$

- Σε όλες τις ερωτήσεις που ζητάται η συγγραφή συνάρτησης MATLAB, το όνομα έχει τη μορφή ****_id όπου στη θέση του id εσείς θα γράψετε το ΑΜ σας. Στη συγγραφή της αναφοράς, παρακαλείστε να θυμηθείτε τις ”καλές πρακτικές” που έχετε μάθει στο ”Συγγραφή και Παρουσίαση Τεχνικών Κειμένων”. Εννοείται ότι θα είναι πολύ καλύτερα αν γράψετε την αναφορά σε LaTeX.
- **Η αναφορά (εκτός από τους κώδικες) πρέπει να είναι σε PDF (όχι Word).** Σε όλες τις περιπτώσεις, **οι κώδικες MATLAB που θα παραδώσετε πρέπει να μπορούν να εκτελεστούν άμεσα από τον ”ελεγκτή”.**

0 Στοιχεία υπολογιστικού συστήματος

Να περιγράψετε τα παρακάτω (Δεν βαθμολογείται αλλά είναι απαραίτητο για την βαθμολόγηση της άσκησης.)

1. Την ημερομηνία που ξεκινήσατε να ασχολείστε με την άσκηση και την ημερομηνία που την ολοκληρώσατε.
2. Τα χαρακτηριστικά του υπολογιστικού συστήματος το οποίο χρησιμοποιήσατε για την υλοποίηση της παρούσας εργαστηριακής άσκησης (π.χ. Υ/Κ, λαπτοπ, PC στο σπίτι).
   (i) Να συμπληρώσετε τα στοιχεία για το σύστημα στο οποίο θα τρέξετε τα πειράματά σας όπως αναφέρονται στον Πίνακα 1. Παρακαλείστε να αναφέρετε Είναι απαραίτητο για τα στοιχεία που θα δώσετε, να αναφέρετε που ή πως τα βρήκατε. Για χρήστες Windows μπορείτε να κατεβάσετε ειδικά προγράμματα όπως το cpuz από τη διεύθυνση http://www.cpuid.com/ το οποίο θα σας δώσει τις πληροφορίες που είναι ζητούμενες. Για χρήστες Linux μπορείτε να βρείτε τις ζητούμενες πληροφορίες μέσω τον εντολών `cat /proc/meminfo` και `cat /proc/cpuinfo` .
   (ii) Έκδοση MATLAB που χρησιμοποιήσατε καθώς και πληροφορίες για τις σχετικές βιβλιοθήκες.
   (iii) Τον πίνακα που προκύπτει όταν εκτελείτε την εντολή bench. Για παράδειγμα, δείτε το Σχήμα 1 στο Παράρτημα.

Πίνακας 1: Στοιχεία για τα πειράματα

Χαρακτηριστικό	ενδεικτική απάντηση
Έναρξη/λήξη εργασίας	15/11/21 - 15/12/21
model	προσωπικό λαπτοπ MacBook Pro[1]
O/S	macOS Catalina 10.15.1
processor name	8-Core Intel Core i9 (i9-9880H) [2]
processor speed	2.3 GHz (base)
number of processors	1
total # cores	8
total # theads	16
FMA instruction	yes
L1 cache	256KB Instruction, 256 KB Data write-back[3]
L2 cache	(per core) 256KB, write-back
L3 cache	(shared) 16MB, write-back
Gflops/s	423.2[4]
Memory	16GB
Memory Bandwidth	41.8 GB/s
MATLAB Version	9.6.0.1174912 (R2019a) Update 5[5]
BLAS	Intel(R) Math Kernel Library Version 2018.0.3 for Intel(R) 64 architecture applications, CNR branch AVX2
LAPACK	Intel(R) Math Kernel Library Version 2018.0.3 for Intel(R) 64 architecture applications, CNR branch AVX2 Linear Algebra PACKage Version 3.7.0

1 Χρονομετρήσεις

Θέλουμε να εξετάσετε κατά πόσον οι χρόνοι εκτέλεσης της εντολής $\mathrm{X = chol(A)}$ ακολουθεί την κυβική πολυπλοκότητα που προβλέπεται για το Ω της κάθε μίας (π.χ. δείτε στις σημειώσεις και διαφάνειες). Για να γίνει αυτό, πρέπει να κατασκευάσετε αντίστοιχη (μαθηματική) συνάρτηση, έστω $T_{chol}(n)$, που μοντελοποιεί το χρόνο εκτέλεσης από μετρήσεις που θα διεξάγετε με τυχαία συμμετρικά θετικά ορισμένα μητρώα μεγέθους $n\times n$ όπου το n λαμβάνει τιμές $\mathrm{n} = [100{:}100{:}2000]$. Θα κατασκευάσετε τη συνάρτηση με την εντολή polyfit της MATLAB. Όπου ζητούνται χρονομετρήσεις, πρέπει να γίνονται βάσει της εντολής timeit. Μπορεί να είναι χρήσιμες οι σχετικές διαφάνειες του Σετ4, Σετ6 και ο κώδικας test_timeit_ger που είχε αναρτηθεί.) Οπότε:

1. Χρησιμοποιώντας την polyfit να βρείτε την κατάλληλη κυβική συνάρτηση (δηλ. $\alpha_3 n^3 + \alpha_2 n^2 + \alpha_1 n + \alpha_0$) που ταιριάζει με τις χρονομετρήσεις (να ελαχιστοποιεί το τετράγωνο του σφάλματος, όπως κάνει η polyfit).

2. Θεωρώντας ότι δεν γνωρίζετε τους χρόνους εκτέλεσης, να υπολογίσετε τις τιμές που προβλέπονται για τους χρόνους απο την παραπάνω συνάρτηση. Αυτό να το κάνετε για n ίσο με τις παραπάνω τιμές (δηλ.$[100{:}100{:}2000]$) καθώς και για τις τιμές $[150{:}100{:}1550]$.

3. Στη συνέχεια να εξετάσετε την ακρίβεια της κάθε πρόβλεψης χρησιμοποιώντας α) τους χρόνους που υπολογίσατε στο πρώτο μέρος για $\mathrm{n} = [100{:}100{:}2000]$ και β) χρονομετρώντας τις εντολές για $\mathrm{n} = [150{:}100{:}1550]$. Να οπτικοποιήσετε τα αποτελέσματα σε μία γραφική παράσταση (για κάθε εντολή) η οποία θα περιέχει: i) τις χρονομετρήσεις για όλα τα παραπάνω n (με σύμβολο'o' χωρίς σύνδεση με γραμμή). ii) τις τιμές που προβλέπονται για το χρόνο εκτέλεσης από τη συνάρτηση (με συνεχή γραμμή).

4. Να δοκιμάσετε να επαναλάβετε το 1ο μέρος παραπάνω ζητώντας από την polyfit να κατασκευάσει πολυώνυμα 2ου βαθμού και 4ου βαθμού. Να συγκρίνετε τις προβλέψεις αυτών με τα κυβικά

[2]https://ark.intel.com/content/www/us/en/ark/products/192987/intel-core-i9-9880h-processor-16m-cache-up-to-4-80-ghz.html
[3]`https://en.wikichip.org/wiki/intel/core_i9/i9-9880h`
[4]https://gadgetversus.com/processor/intel-core-i9-9880h-vs-intel-core-i7-9750h/
[5]Εντολή version στη MATLAB.

πολυώνυμα.

ΣΥΣΤΑΣΗ 1): Λόγω του εύρους στις τιμές του n, μπορεί να έχετε πιο αξιόπιστα αποτελέσματα στην polyfit αν την καλέσετε ως [p,s,mu]=polyfit(x,y,deg) όπου το deg είναι ο βαθμός του επιθυμητού πολυωνύμου (π.χ. 3, 2,4). Δείτε help polyfit για λεπτομέρειες εφαρμογής.

Παραδοτέο

i) Σύντομη περιγραφή όσων κάνατε και επαρκή σχόλια στα αρχεία με τους κώδικες. ii) Σχολιασμός των αποτελεσμάτων.

2 Ειδικοί επιλυτές και αραιά μητρώα

Στην παρούσα ερώτηση θα αναφερόμαστε αποκλειστικά σε μητρώα ζώνης (π.χ. τριδιαγώνια ή παραλλαγή τους) οπότε θα μπορούν να αποθηκευτούν σε λίγα διανύσματα, όπως π.χ. στην αναπαράσταση LAPACK για μητρώα ζώνης . Πρόκειται ουσιαστικά για την Άσκηση της σελ. 17 του σετ 11.

1. Να υλοποιήσετε συνάρτηση για την επίλυση τριδιαγώνιων συστημάτων όπως περιγράφεται στο σετ11, διαφάνειες 13 κ.ο.κ. Ως είσοδο θα λαμβάνουν ένα τριδιαγώνιο μητρώο στη μορφή που αναφέραμε καθώς και ένα array B το οποίο μπορεί να αποτελείται από 1 ή περισσότερα δεξιά μέλη.

2. Να μελετήσετε τη μέθοδο και να δείτε όσο καλύτερα τις ιδιότητές της. Ο αλγόριθμος θα πρέπει να υλοποιείται αποκλειστικά με βάση πράξεις μεταξύ των διανυσμάτων των διαγωνίων (με πολλαπλασιασμούς Hadamard διανυσμάτων) και με αυτών με τα διανύσματα στο B. Ενδιαφέρουν ιδιαίτερα: Το αριθμητικό κόστος ως συνάρτηση του n,

3. Να πειραματιστείτε με διαγώνια κυρίαρχα μητρώα και να εξετάσετε και να σχολιάσετε τη συμπεριφορά των νορμών των υπερ/υπό-διαγωνίων.

Παραδοτέο

i) Σύντομη περιγραφή όσων κάνατε και επαρκή σχόλια στα αρχεία με τους κώδικες. ii) Σχολιασμός των αποτελεσμάτων.

3 Τανυστές[6]

Θα πρέπει να υλοποιήσετε συναρτήσεις που θα εκτελούν πράξεις τανυστών αλλά δεν θα χρειάζονται το TT και οι τανυστές θα είναι σε μορφή multidimensional arrays (MDA). Οι συναρτήσεις πρέπει να κάνουν τους ελέγχους που είναι απαραίτητοι για τα στοιχεία εισόδου (αν επαρκούν, αν είναι του σωστού τύπου, αν οι διαστάσεις είναι έγκυρες) και σε περίπτωση που δεν είναι έγκυρα θα πρέπει να επιστρέφουν μήνυμα σχετικά με το σφάλμα.

1. Μία (εσείς επιλέγετα) από τις συναρτήσεις ttv_myid, ttm_myid για τον τροπικό πολλαπλασιασμό τανυστή τριών τρόπων με μητρώο ή διάνυσμα ώστε τα αποτελέσματα που προκύπτουν να είναι ίδια με εκείνα των αντίστοιχων συναρτήσεων (ttv, ttm) του Tensor Toolbox (TT, που θα πρέπει να έχετε κατεβάσει στο χώρο εργασίας σας για να κληθούν όταν χρειάζεται έλεγχος αποτελεσμάτων). Οι συναρτήσεις παρακάτω:

[6] Για το μέρος αυτό, πρέπει να κατεβάσετε το Tensor Toolbox που αναφέραμε στην αντίστοιχη διάλεξη.

- Θα λαμβάνουν ως είσοδο και επιστρέφουν multidimensional arrays (MDA) και δεν θα χρειάζονται το TT.
- Θα πρέπει να επιστρέφεται μήνυμα σφάλματος αν δεν δίνονται έγκυρα στοιχεία εισόδου είτε λόγω είδους (π.χ. να μην είναι vector ενώ προβλεπεται vector, κ.λπ.) καθώς και αν οι διαστάσεις δεν είναι κατάλληλες για τη συνάρτηση.
- Δεν είναι αποδεκτή η υλοποιήσή τους μέσω εντολών από το TT ούτε με χρήση αντίστοιχων (νέων) εντολών της Mathworks, όπως tensorprod, pagemtimes ή αλλες σχετικές.
- Θα χρησιμοποιούν τις συναρτήσεις ttv, ttm, ttt για να ελέγχουν την ορθότητα των αποτελεσμάτων στα data sets που αναφέρονται.

Οι μεταβλητές εισόδου θα είναι απλούστερες των αρχικών ttv, ttm) και ειδικότερα, αρκεί υλοποίηση που λειτουργεί ως εξής. Για το

```
Y = ttv_myid(X,V,N) computes the product of a multidimensional array X with a (column) ...
    vector V.  The integer N specifies the dimension in X along which V is multiplied.  ...
    If size(V) = [I,1], then X must have size(X,N) = I.  Note that ndims(Y) = ndims(X) - ...
    1 because the N-th dimension is removed.
```

Αντίστοιχα για το ttm_myid. Δηλ. θα υλοποιήσετε μόνον τροπικό πολλαπλασιασμό με ένα αντικείμενο (μητρώο ή διάνυσμα ως προς τον δοθέντα τρόπο και όχι πολλαπλασιασμούς με αλληλουχία μητρώων ή διανυσμάτων που επιτρέπουν οι κώδικες του TT.

2. Τη συνάρτηση ttt_myid που θα είναι απλή μορφή της ttt για το εξωτερικό και το εσωτερικό γινόμενο των τανυστών που αντιστοιχούν στις MDA εισόδου. Ειδικότερα: Το C=ttt_myid(A,B) επιστρέφει στο C το MDA που περιέχει τις τιμές και τη σωστή διάταξη του τανυστή $A \circ B$, αν αυτοί ήταν τανυστές. και το t=ttt_myid(A,B,'all') επιστρέφει το εσωτερικό γινόμενο των τανυστών.

Παραδοτέο

i) Σύντομη περιγραφή των υλοποιήσεων και επαρκή σχόλια στα αρχεία με τους κώδικες. ii) Συνάρτηση test_tensor, που επιβεβαιώνει την ορθότητα των συναρτήσεων που αναπτύξατε συγκρίνοντας με τις αντίστοιχες εντολές του TT. Θα πρέπει να χρησιμοποιήσετε ως δεδομένα τις εντολές:

```
% Αρχικοποίηση
clear;  tol=1e-8;nd=3;  rng(0);  err=zeros(1,nd+2);
ndim=[2,3,4];  Atemp= randi(5,ndim);  Btemp=randi(4,ndim);
X=randi([-1,1],max(ndim),1);  A=tensor(Atemp);  B=tensor(Btemp);
```

Στο τέλος, θα πρέπει να εκτελέσετε τις παρακάτω εντολές:

```
try
    for k=1:nd
        err(k)=norm(ttv_myid(A,X(1:ndim(k),1),k)-double(ttv(A,X(1:ndim(k),1),k)))
    end
    assert(max(err)<tol ,'ttm modal multiplication fails')
    catch ME1
end
try
    err(nd+1)  =   norm(tensor(ttt_myid(A,B))-ttt(A,B));
    assert (err(nd+1)<tol,'ttt outer multiplication fails')
```

```
    catch ME2
end
try
    err(nd+2)  =    abs(ttt_myid(A,B, 'all')-double(ttt(A,B,[1:nd])));
    assert (err(nd+2)<tol, 'ttt inner product fails')
    catch ME3
end
if (exist('ME1'))
    ME1.message
end
if (exist('ME2'))
    ME2.message
end
if (exist('ME3'))
    ME3.message
end
```

4 Επίλυση ΣΘΟ συστημάτων με PCG

Γνωρίζουμε ότι (σε αριθμητική άπειρης ακρίβειας), η pcg με είσοδο το ΣΘΟ μητρώο A και δεξιό μέλος b, επιστρέφει προσέγγιση της λύσης του $Ax = b$ η οποία ελαχιστοποιεί την A-νόρμα του σφάλματος, δηλ. το $\arg\min_y \|x_{\text{sol}} - y\|_A$. Εσείς πρέπει να τροποποιήσετε τη συνάρτηση pcg της Mathworks ώστε να μπορεί να δέχεται ως επιπλέον στοιχείο εισόδου ένα διάνυσμα xsol που εκλαμβάνεται ως ακριβής (π.χ. είναι γνωστή ή την έχουμε υπολογίσει με διαφορετική, πιο ακριβή μέθοδο) λύση και να επιστρέφει, επιπλέον του resvec, ένα διάνυσμα errvec, ίδιου μεγέθους, που περιέχει την A-νόρμα του σφάλματος από το $x^{(0)}$ μέχρι το τελευταίο βήμα της μεθόδου.

Η νέα συνάρτηση θα ονομάζεται pcg_myid. Αναμένεται να την αντιγράψετε τις σχετικές συναρτήσεις της Mathworks (βλ. στον κατάλογο sparfun) και να τις τροποποιήσετε όπου χρειάζεται ώστε να δίνεται η δυνατότητα της παρακάτω κλήσης:

```
function [x,flag,relres,iter,resvec,errvec] = ...
    pcg(A,b,tol,maxit,M1,M2,x0,xsol,varargin)
....
%    [X,FLAG,RELRES,ITER,RESVEC,ERRVEC] = PCG(A,B,TOL,MAXIT,M1,M2,X0,XSOL) ...
    specifies the exact solution and returns the A-norm of the error.
```


Θα δοκιμάσετε τη συνάρτηση για δύο μητρώα. I) Το κλασικό μητρώο που προέρχεται από τη διακριτοποίηση με κεντρισμένες διαφορές της της εξίσωσης Poisson με συνθήκες Dirichlet στο τετράγωνο $[0, 1] \times [0, 0.5]$, δηλ. το ακρότατο κάτω αριστερά σημείο είναι το $[0, 0]$ και το ανώτατο δεξιό σημείο είναι $[1, 1/2]$. Και τα δύο αυτά σημεία είναι στο σύνορο. Η εξίσωση είναι

$$-(u_{xx} + u_{yy}) \quad = \quad \pi^2 \sin(\pi x) - 6xy - 2y$$

Οι συνοριακές τιμές είναι $u(0, y) = 0$, $u(1, y) = 1 + y^3$, $u(x, 0) = x^2 + \sin(\pi x)$, $u(x, 1/2) = x^2 + x/8$. Η απόσταση διαδοχικών κόμβων h είναι ίδια και στις δύο κατευθύνσεις, δηλ. $x_{i+1} - x_i = y_{i+1} - y_i = h$. Υπάρχουν $n = 30$ εσωτερικοί κόμβοι για κάθε τιμή του y, δηλ. $(x_i, y), i = 1, ..., 30$, επομένως $x_i = \frac{i}{n+1}$. Με αυτό το h ορίζονται οι τιμές των $y_1, ..., y_m$. Επομένως, το πλέγμα θα περιέχει $n \times m$ εσωτερικούς κόμβους και το μητρώο θα είναι $mn \times mn$.

II) Το μητρώο υπ' αριθμ. 1 του SuiteSparse, δηλ. το 1138_bus της υποκατηγορίας HB (Harwell-Boeing). Ως δεξιό μέλος, b=A*ones(n,1) για την κατάλληλη τιμή n=1138.

Να παράγετε ένα σχήμα για κάθε μητρώο όπου να φαίνεται η διακύμανση του σχετικού υπολοίπου και του σχετικού σφάλματος[7] με και χωρίς προρρύθμιση. Ειδικότερα, πρέπει να φαίνεται η διακύμανση των $\|b - Ax^{(m)}\|/\|b\|$ και $\|x_{\text{sol}} - x^{(j)}\|_A/\|x_{\text{sol}}\|_A$..

Σε όλες τις περιπτώσεις, να θέσετε ένα άνω φράγμα στο πλήθος των επαναλήψεων, το maxiter=4 n αν το πρόβλημα είναι $n \times n$.

Για προρρύθμιση να χρησιμοποιήσετε: α) προρρύθμιση με incomplete Cholesky IC(0). β) δική σας προσπάθεια να βρείτε προρρυθμιστή που παράγει βελτιωμένα αποτελέσματα, π.χ. μέσω επιλογών σας στην εντολή ichol. Η βελτίωση πρέπει να αφορά το πλήθος των επαναλήψεων καθώς και το χρόνο εκτέλεσης με βάση το tic-toc, χρησιμοποιώντας μέσους όρους για το χρόνο επίλυσης.

Σε όλες τις περιπτώσεις, επιπλέον των σχημάτων, να συμπεριλάβετε και πίνακα στον οποίο θα φαίνεται το τελικό σχετικό υπόλοιπο και σφάλμα καθώς και οι χρόνοι επίλυσης μέσω tic-toc.

Παραδοτέο

i) Σύντομη περιγραφή των αλλαγών στον pcg, κώδικες, πειράματα επαλήθευσης, ii) σχολιασμός των αποτελεσμάτων.

Α΄ Παράρτημα

Computer Type	LU	FFT	ODE	Sparse	2-D	3-D
This machine	0.0693	0.0472	0.0695	0.0841	0.2684	0.5397
Macintosh (64-bit) 3.5 GHz Intel Core i7	0.0909	0.0593	0.0734	0.0889	0.3833	0.6312
Linux (64-bit) 2.66 GHz Intel Xeon	0.1295	0.0654	0.1933	0.1295	0.5581	0.5806
Windows 7 Enterprise (64-bit) 2.66 GHz Intel Xeon	0.1357	0.0705	0.1678	0.1535	0.4210	0.9613
Windows 7 Professional (64-bit) 3.07 GHz Intel Xeon W3550	0.2462	0.0844	0.1382	0.1476	0.3723	0.8357
Windows 7 Enterprise (64-bit) 2.7 GHz Intel Core i7	0.2926	0.0820	0.1075	0.1311	0.5387	1.0212
Macintosh (64-bit) 2.6 GHz Intel Core i7	0.1880	0.0904	0.0884	0.1297	0.9780	1.0788
Windows 8.1 (64-bit) 2.67 GHz Intel Xeon X5650	0.3146	0.0713	0.2053	0.2084	0.5210	0.8763
Windows XP (32-bit) 2.4 GHz Intel Core 2 Quad	0.6330	0.2110	0.1671	0.2818	0.5823	1.3302

Σχήμα 1: Αποτελέσματα της bench σε Pentium i9.

[7] Για το πρώτο σύστημα, να χρησιμοποιήσετε ως ακριβή λύση το A\b.