

Introducere în calcul științific

Suport de curs și laborator

pentru studenții anului al II-lea

Ionel-Dumitrel Ghiba

Facultatea de Matematică, Universitatea “Alexandru Ioan Cuza” din Iași

dumitrel.ghiba@uaic.ro



ALEXANDRU IOAN CUZA
UNIVERSITY of IAȘI

Prefață

Calculul științific se află la intersecția dintre matematică, algoritmi și calculul efectiv pe calculator. El pornește de la o întrebare simplă: *cum transformăm un model matematic într-un rezultat numeric în care putem avea încredere?* Răspunsul nu constă doar în aplicarea unei formule. Trebuie să alegem o reprezentare adecvată a problemei, un algoritm suficient de eficient, să înțelegem erorile introduse de aritmetica în virgulă mobilă și să putem decide dacă rezultatul calculat este relevant pentru problema inițială.

Acest curs este conceput ca o primă introducere în aceste idei. Algebra liniară numerică apare în mare parte a cursului, deoarece numeroase probleme de calcul științific se reduc, exact sau după discretizare, la operații cu vectori și matrice: rezolvarea sistemelor liniare, aproximarea datelor, problema celor mai mici pătrate, calculul valorilor și vectorilor proprii, optimizarea unor funcții pătratice, aproximarea de rang mic prin descompunerea în valori singulare sau reducerea dimensiunii datelor prin analiza componentelor principale.

Scopul cursului nu este memorarea unei colecții de algoritmi. Pentru fiecare metodă ne interesează patru întrebări: *de unde provine metoda?*, *de ce funcționează?*, *cât costă?* și *cât de mult putem avea încredere în rezultatul obținut?* Aceste întrebări vor însoți metodele prezentate pe parcursul cursului.

De la formulă la algoritm

În matematică, două formule echivalente descriu același obiect. Pe calculator, două proceduri algebric echivalente pot avea costuri și comportamente numerice foarte diferite. De exemplu, pentru sistemul $Ax = b$, formula $x = A^{-1}b$ este corectă matematic, dar calcularea explicită a inversei este în general o alegere numerică mai slabă decât rezolvarea sistemului printr-o factorizare adecvată. În calculul științific contează, așadar, nu numai formula, ci și modul în care ea este transformată într-un algoritm.

Manualul este destinat studenților anului al II-lea și presupune cunoștințe de bază de algebră liniară și analiză. Expunerea combină demonstrația matematică cu interpretarea numerică, exemple complet calculate, exerciții, sarcini de programare și mini-proiecte.

Cum se citește o secțiune

La prima lectură urmăriți ideea metodei și rolul fiecărei ipoteze. La a doua lectură refaceți demonstrațiile și calculele. În final, implementați algoritmul și testați-l pe exemple pentru care cunoașteți răspunsul. O metodă numerică este înțeleasă cu adevărat atunci când puteți explica de ce funcționează, când poate eșua, ce eroare produce și cum verificați rezultatul.

Cum lucrăm la seminar și laborator

În acest manual, calculul manual și programarea nu sunt activități separate. O implementare este considerată înțeleasă numai dacă studentul poate explica formulele din care provine, ordinea operațiilor, ipotezele în care algoritmul funcționează și un criteriu prin care rezultatul este verificat.

Pentru fiecare algoritm nou se recomandă aceeași succesiune. Mai întâi se execută cu pixul pe hârtie unul sau două cazuri mici, fără a sări etape. Apoi se scrie pseudocodul și se identifică dependențele dintre pași. Numai după aceea se trece la MATLAB. Funcțiile MATLAB de bibliotecă sunt folosite în primul rând ca referință pentru verificare; scopul laboratorului este construirea algoritmului, nu înlocuirea lui printr-o singură comandă.

Ce înseamnă “a explica un algoritm”?

Studentul trebuie să poată răspunde la întrebări precum: de ce această buclă începe de la indicele $k + 1$? de ce este necesar pivotul nenul? ce mărime scade de la o iterație la alta? de ce folosim QR și nu ecuațiile normale? ce se întâmplă dacă o valoare singulară este foarte mică? Un cod care produce un număr, dar ale cărui etape nu pot fi justificate, nu este încă o soluție completă.

Proiectele au nivel mediu și urmăresc integrarea noțiunilor. Nu se cere construirea unui sistem software complex. Se cere însă ca modelul, algoritmul, implementarea MATLAB, verificarea și interpretarea rezultatului să formeze un argument coerent.

Bibliografie de bază și materiale didactice consultate

Expunerea folosește rezultate standard din calcul științific și algebră liniară numerică și a fost calibrată didactic și prin consultarea unor suporturi moderne de curs. Principalele repere sunt:

- Lucian Maticiuc, *Introducere în MATLAB*, Iași, 2019, material disponibil online la https://www.math.uaic.ro/~maticiuc/didactic/MATLAB_Course.pdf.
- A. Quarteroni, R. Sacco, F. Saleri, *Numerical Mathematics*, Springer.
- W. Gander, M. J. Gander, F. Kwok, *Scientific Computing – An Introduction Using Maple and MATLAB*, Springer.
- A. Beck, *Introduction to Nonlinear Optimization – Theory, Algorithms, and Applications with MATLAB*, SIAM.
- L. Page, S. Brin, R. Motwani, T. Winograd, *The PageRank Citation Ranking: Bringing Order to the Web*, Stanford Digital Library Technologies Project, 1998/1999.
- Y. Koren, R. Bell, C. Volinsky, “Matrix Factorization Techniques for Recommender Systems”, *Computer*, 42(8), 30–37, 2009.
- J. Calder, *Math 5467: Introduction to the Mathematics of Image and Data Analysis*, University of Minnesota, School of Mathematics, 2022.
- J. Gallier, J. Quaintance, *Linear Algebra for Computer Vision, Robotics, and Machine Learning*, University of Pennsylvania, 2025.
- O. Çeliker, *MIT 6.057: Introduction to MATLAB*, IAP 2019.

Organizarea și evaluarea cursului

Evaluarea la această disciplină este pe parcurs și este alcătuită din 2 componente de evaluare continuă. Evaluarea continuă constă în

- 70% verificarea înțelegerii rezolvărilor unor probleme la laborator, asemănătoare cu cele din fișele propuse săptămânal. Această verificare se va face cerând la laborator ca studenții să rezolve individual în 20 minute o sarcină de laborator. Aceste activități au caracter continuu și vor fi notați, desigur, doar studenții care le vor îndeplini. Nu se repetă, fiind considerate activități curente de laborator. **Nu se repetă pe parcursul semestrului sau în sesiunea de restanțe!**
- 30% verificarea cunoștințelor teoretice printr-un test scurt de 20 minute în ultima săptămână, la începutul ultimului curs. **Se poate repeta în sesiunea de restanțe.**

Ce se evaluează

- Accentul nu este pus numai pe obținerea unui rezultat numeric. Studentul trebuie să poată justifica alegerea metodei, să explice ipotezele teoretice, să estimeze costul și eroarea și să interpreteze rezultatul unui experiment numeric.
- Nu se punctează prezența, însă absentarea de la orele de laborator vor conduce la imposibilitatea îndeplinirii sarcinilor de lucru, neîndeplinirea lor notându-se cu nota 1/10.
- Prezența la curs nu este obligatorie, însă multe explicații teoretice și de îndeplinire a sarcinilor de laborator vor fi date doar la curs și vor fi foarte utile, evaluarea insistând pe aceste aspecte comunicate doar în sala de curs.

Notatii

Unele litere sunt folosite cu sensuri diferite în capitole diferite. Pentru a evita confuziile, convențiile principale sunt reunite mai jos.

Simbol	Sens	Capitol
P	matrice de permutare	metode directe / pivotare
P	matrice de splitting sau preconditionare	metode iterative
P_L	proiector ortogonal pe subspațiul L	PCA
P	matrice stocastică de tranziție	PageRank
U	factor superior triunghiular în LU	metode directe
U	matricea vectorilor singulari stânga	SVD
$r = b - Ax$	reziduul pentru sisteme liniare	convenție folosită în manual

Sensul unui simbol este precizat din nou acolo unde poate apărea o confuzie.

Cuprins

Prefață	i
Cum lucrăm la seminar și laborator	ii
Bibliografie de bază și materiale didactice consultate	iii
Organizarea și evaluarea cursului	iv
Notății	v
1 Introducere în calcul științific	1
1.1 Ce este calculul științific?	1
1.2 De ce apar vectorii și matricile în aplicații?	1
1.3 Problema model: sistemul liniar	4
1.4 Erori, stabilitate și condiționare	4
1.5 Cum este organizat cursul?	4
Test de capitol	7
2 Recapitulare: matrice, valori proprii și pozitiva definire	8
Extensie didactică: interpretarea spectrală a matricelor	11
Fișă de seminar: geometria matricelor simetrice	13
Test de capitol	18
3 Metode directe pentru sisteme liniare	19
3.1 Rezolvarea sistemelor triunghiulare	19
3.2 Eliminarea gaussiană și factorizarea LU	23
3.3 Forma compactă a factorizării LU	33
3.4 Factorizarea LDM^T	36
3.5 Pivotare	40
3.6 Factorizarea Cholesky	47
3.7 Matrice cu structură de bandă	52
Fișă de seminar și laborator: metode directe în ansamblu	57

Test de capitol	62
4 Norme, condiționare și analiza erorii	63
Aritmetica în virgulă mobilă	63
Extensie didactică: norme, perturbații și numărul de condiționare	73
4.1 Îmbunătățirea acurateței eliminării gaussiene	76
Fișă de seminar: eroare, condiționare și stabilitate	80
Test de capitol	85
5 Metode iterative pentru sisteme liniare	86
Extensie didactică: metode iterative ca probleme de punct fix	89
5.1 Jacobi, Gauss-Seidel și metode relaxate	92
5.2 Convergența metodelor Jacobi și Gauss-Seidel	96
Fișă de seminar și laborator: metode iterative	102
Test de capitol	107
6 Sisteme dreptunghiulare și aproximații	108
Extensie didactică: ce înseamnă un sistem dreptunghiular?	108
6.1 “Soluția” sistemelor supradeterminate	109
6.2 Problema celor mai mici pătrate și ajustarea datelor	114
6.3 Ajustarea datelor (data fitting)	117
Fișă de seminar: sisteme dreptunghiulare și modelarea datelor	122
Test de capitol	126
7 Regularizarea problemei celor mai mici pătrate și eliminarea zgomotului	127
Extensie didactică: regularizarea ca informație a priori	130
Test de capitol	135
8 Factorizarea QR	136
Extensie didactică: ortogonalizarea ca instrument numeric	139
8.1 Sisteme dreptunghiulare rezolvate cu factorizarea QR	142
Fișă de seminar și laborator: factorizarea QR	146
Test de capitol	150
9 Metode de descreștere și metoda gradientului	151
Extensie didactică: un sistem SPD ca problemă de minimizare	153
9.1 Metoda direcțiilor de descreștere	155
9.2 Metoda gradientului	159
9.3 Numărul de condiționare pentru funcții pătratice. Rescalarea problemei.	163
9.4 Rescalare	166

Fișă de seminar: metode de descreștere și gradient	174
Extensie: metoda gradientului conjugat	175
Test de capitol	179
10 Metoda Gauss–Newton pentru cele mai mici pătrate neliniare	180
Extensie didactică: Gauss–Newton ca secvență de probleme LS liniare	182
Fișă de seminar și laborator: Gauss–Newton	184
10.1 Problema celor mai mici pătrate pentru modele neliniare: ajustarea cercurilor și problema localizării	185
Test de capitol	194
11 Descompunerea în valori singulare și pseudoinversa	195
Extensie didactică: geometria și puterea descompunerii SVD	200
Fișă de seminar: SVD și pseudoinversa	202
Test de capitol	206
12 Aplicații SVD în compresia imaginilor	207
12.1 Descompunerea în valori singulare (SVD) și tensorii de ordinul al doilea	207
12.2 Compresia imaginilor în tonuri de gri	213
12.3 Compresia imaginilor color în spațiul RGB	217
12.4 Compresia imaginilor color în spațiul YCbCr	222
Test de capitol	228
13 Analiza componentelor principale (PCA)	229
13.1 Motivație: cum descriem date de dimensiune mare prin puține direcții?	229
13.2 Centrare, varianță și covarianță	229
13.3 Cel mai bun subspațiu afin în sensul celor mai mici pătrate	230
13.4 Principiul Rayleigh–Ritz și alegerea direcțiilor principale	232
13.5 PCA ca maximizare a varianței	234
13.6 Legătura exactă dintre PCA și SVD	234
13.7 Reducerea dimensiunii și reconstrucția	235
13.8 Câte componente păstrăm? Varianța explicată	236
13.9 Exemplu complet calculat	237
13.10Centrare sau standardizare?	238
13.11Algoritmul PCA în MATLAB	238
13.12Aplicație: o colecție de imagini ca nor de puncte	240
13.13Limitări și situații în care PCA trebuie folosită cu atenție	241
13.14Exerciții	242
13.15Rezumatul capitolului	243

Test de capitol	244
14 Aplicații integratoare: PageRank și sisteme de recomandare	245
14.1 De ce introducem aceste aplicații abia acum?	245
14.2 PageRank ca problemă de calcul științific	245
14.2.1 De la un graf orientat la o matrice stocastică	245
14.2.2 Modelul surferului aleator și teleportarea	246
14.2.3 Forma spectrală	247
14.2.4 Exemplu complet cu patru pagini	247
14.3 Un model de recomandare prin factorizare de rang mic	250
14.4 Ce noțiuni din curs au fost folosite?	252
Test de capitol	253
În loc de încheiere	254
Tabel de referință: costuri dominante	254
Fișă finală: compresia imaginilor ca proiect de calcul științific	255

Introducere în calcul științific

1.1 Ce este calculul științific?

Calculul științific studiază transformarea unui model matematic într-un calcul efectiv, controlabil și reproductibil. Într-o problemă reală nu este suficient să știm că o soluție există sau că poate fi scrisă formal printr-o formulă. Trebuie să construim un algoritm, să estimăm costul lui, să înțelegem efectul erorilor de rotunjire și al perturbațiilor din date și, în final, să putem verifica dacă rezultatul calculat este relevant pentru problema inițială.

Un exemplu elementar este sistemul liniar $Ax = b$. Formula $x = A^{-1}b$ este corectă atunci când A este inversabilă, dar nu descrie metoda numerică pe care ar trebui să o folosim. În practică, alegerea depinde de structură: un sistem algebric având o matrice triunghiulară se rezolvă prin substituție, un sistem algebric având o matrice generală prin eliminare Gauss și factorizare LU cu pivotare, un sistem algebric având o matrice simetrică pozitiv definită (adică $A = A^T$ și $x^T Ax > 0$ pentru orice $x \neq 0$) prin Cholesky, iar o problemă foarte mare cu o matrice rară – adică având majoritatea elementelor nule – poate favoriza o metodă iterativă. Aceeași problemă matematică poate conduce astfel la algoritmi foarte diferiți.

Trei niveluri care nu trebuie confundate

În calculul științific avem permanent *problema matematică*, *algoritmul* și *implementarea*. Problema poate fi bine sau prost condiționată; algoritmul poate fi stabil sau instabil; implementarea poate fi corectă sau poate conține erori. Un rezultat numeric bun cere înțelegerea tuturor celor trei niveluri menționate.

1.2 De ce apar vectorii și matricile în aplicații?

De exemplu, după discretizarea unei ecuații diferențiale, valorile necunoscute ale unei temperaturi, deplasări sau presiuni devin componentele unui vector, iar relațiile dintre ele sunt descrise de o matrice. În ajustarea datelor experimentale, liniile unei matrice conțin evaluările unui model în punctele măsurate. În procesarea imaginilor, o imagine în tonuri de gri este chiar o matrice de intensități, iar o imagine color poate fi privită ca un ansamblu de trei astfel de matrici. În probleme de rețea, matricile descriu conexiunile dintre noduri. În sisteme de recomandare, matricile pot codifica interacțiunile dintre utilizatori și obiecte.

Pentru a vedea mai concret cum apar aceste matrice, să considerăm câteva situații simple. Dacă discretizăm problema de ordinul întâi

$$u'(x) = g(x), \quad u(0) = 0,$$

pe o rețea uniformă $x_i = ih$, atunci aproximația cu diferență înainte

$$\frac{u_i - u_{i-1}}{h} \simeq g(x_i), \quad i = 1, \dots, n,$$

conduce, pentru necunoscutele $u_1, \dots, u_n$, la sistemul

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ -1 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & 0 \\ 0 & \cdots & -1 & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} = h \begin{pmatrix} g(x_1) \\ g(x_2) \\ \vdots \\ g(x_n) \end{pmatrix}.$$

Astfel, matricea apare direct din relațiile dintre valorile vecine ale soluției pe rețea.

În ajustarea unei drepte $y \simeq a + bt$ la datele (t_i, y_i) , fiecare observație furnizează relația $a + bt_i \simeq y_i$. Reunind toate relațiile obținem

$$\begin{pmatrix} 1 & t_1 \\ 1 & t_2 \\ \vdots & \vdots \\ 1 & t_m \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} \simeq \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix},$$

deci matricea apare direct prin evaluarea funcțiilor 1 și t în punctele măsurate.

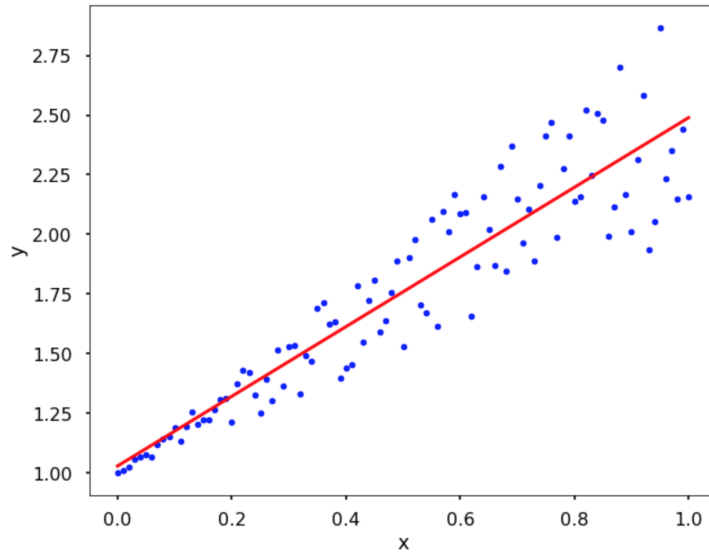


Figura 1.1: Date experimentale și o dreaptă de ajustare. Fiecare observație contribuie cu o relație liniară aproximativă pentru parametrii dreptei.

Într-o rețea cu n noduri, o descriere elementară este matricea de adiacență $A = (a_{ij})$, unde $a_{ij} = 1$ dacă nodurile i și j sunt conectate și $a_{ij} = 0$ în caz contrar. Astfel, chiar structura conexiunilor este codificată într-o matrice.

Aceste exemple au o caracteristică comună: dimensiunea problemei poate fi mare, datele pot conține zgomot, iar structura matricei este adesea la fel de importantă ca valorile numerice ale elementelor sale. De aceea, cursul nu tratează algebra liniară doar ca pe un set de identități, ci ca pe un limbaj pentru construirea și analiza algoritmilor.

Un exemplu practic important: o imagine ca matrice

O imagine în tonuri de gri cu $m \times n$ pixeli poate fi reprezentată printr-o matrice $A \in \mathbb{R}^{m \times n}$. Elementul a_{ij} descrie intensitatea pixelului din poziția (i, j) : valori mici corespund unor tonuri mai închise, iar valori mari unor tonuri mai deschise. Dacă această matrice admite o aproximare bună de rang mic, ea poate fi descrisă cu mult mai puține date. Mai târziu vom demonstra că descompunerea în valori singulare furnizează, într-un sens precis, cea mai bună aproximație de rang fix. Pentru moment este suficient să observăm ordinul de mărime: o imagine 4000×3000 conține 12 000 000 de valori, în timp ce o reprezentare de rang 50 necesită aproximativ $50(4000 + 3000 + 1)$ valori, adică aproximativ 350 000.

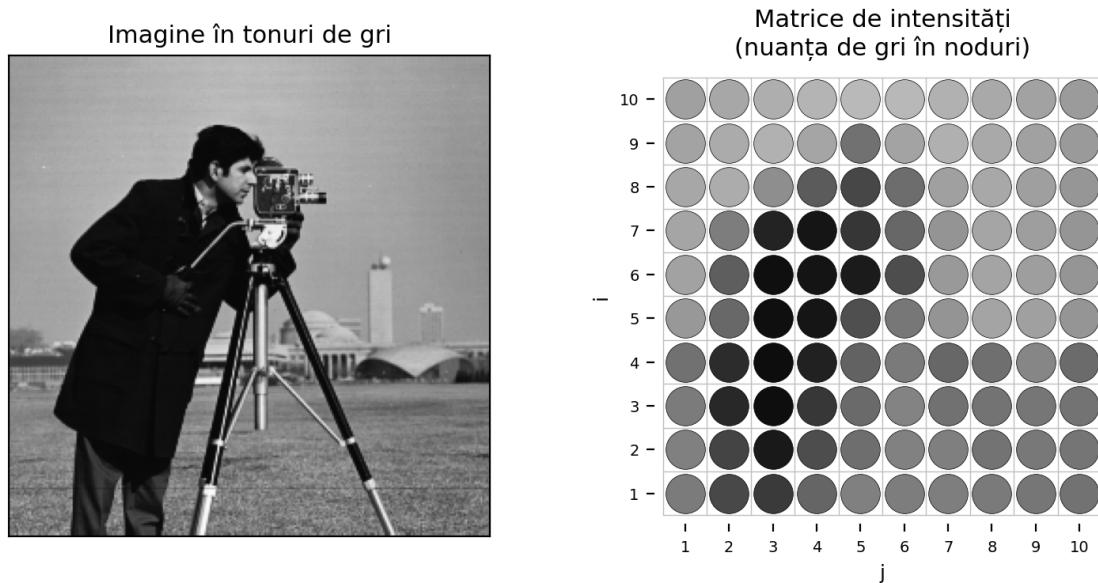


Figura 1.2: O imagine alb-negru și o reprezentare matricială grosieră. Fiecărui nod din grilă îi corespunde o nuanță de gri, deci o valoare a matricei de intensități.

Acest exemplu arată de ce noțiuni aparent abstracte precum rangul, ortogonalitatea și valorile singulare au consecințe practice directe. În capitolele finale vom reveni la imagini și vom construi efectiv algoritmi MATLAB de compresie.

Alte aplicații care vor reapărea în curs

Problema celor mai mici pătrate apare în calibrare, ajustarea unor modele la date și identificarea parametrilor. Regularizarea permite stabilizarea problemelor sensibile la zgomot. Metodele de gradient și Gauss–Newton apar în optimizare și ajustări neliniare. Matricile rare și metodele iterative apar în discretizări ale ecuațiilor diferențiale și în probleme de rețea de dimensiuni foarte mari.

Spre sfârșitul manualului vom reveni la mai multe dintre aceste noțiuni prin algoritmul PageRank pentru analiza rețelelor. El va fi introdus numai după ce vom avea la dispoziție norme induse, raza spectrală, metodele iterative, analiza convergenței și reprezentarea rară a matricelor. În acel moment vom putea nu doar să scriem o iterație, ci să demonstrăm de ce converge, să explicăm de ce este eficientă și să o implementăm corect în MATLAB. În aceeași parte finală vom discuta și un model simplificat de recomandare bazat pe factorizări de rang mic și pe problema celor mai mici pătrate.

1.3 Problema model: sistemul liniar

Unul dintre ideile centrale ale cursului este sistemul $Ax = b$, unde $A \in \mathbb{R}^{m \times n}$, $x \in \mathbb{R}^n$ și $b \in \mathbb{R}^m$. În cazul pătratic, $m = n$, iar dacă A este inversabilă, soluția este unică. Din punct de vedere teoretic putem scrie $x = A^{-1}b$ sau putem invoca regula lui Cramer. Din punct de vedere numeric, întrebarea relevantă este însă cum obținem x cu un cost rezonabil și cu o eroare controlată.

Regula lui Cramer ilustrează bine diferența dintre formulă și algoritm. Calcularea determinantilor prin dezvoltare Laplace are un cost care crește factorial cu dimensiunea. Eliminarea Gauss și factorizarea LU au, pentru matrice dense, cost de ordin n^3 și permit apoi rezolvarea eficientă pentru mai mulți termeni liberi. Pentru matrici rare sau structurate costul poate fi redus substanțial.

Direct sau iterativ?

Metodele directe, precum LU , Cholesky și QR , produc în aritmetică exactă rezultatul după un număr finit de pași. Metodele iterative construiesc o succesiune $x^{(0)}, x^{(1)}, \dots$ care trebuie să tindă la soluție și sunt potrivite pentru multe probleme foarte mari și rare. Alegerea nu se face numai după dimensiunea matricei: contează simetria, pozitivă definită, raritatea, condiționarea și numărul de sisteme care trebuie rezolvate cu aceeași matrice.

1.4 Erori, stabilitate și condiționare

Calculatorul nu efectuează operații cu numere reale exacte. Numerele sunt reprezentate cu precizie finită, iar fiecare operație poate introduce o eroare de rotunjire. În plus, datele unei probleme provin adesea din măsurători și au propriile erori. Din acest motiv vom distinge două întrebări.

Prima este o proprietate a problemei: *cât de mult se modifică soluția dacă datele se modifică puțin?* Aceasta este problema condiționării. A doua este o proprietate a algoritmului: *cât de mult amplifică procedura numerică erorile inevitabile ale calculului?* Aceasta este problema stabilității. Un algoritm foarte bun nu poate transforma o problemă extrem de prost condiționată într-una insensibilă, dar un algoritm instabil poate degrada inutil chiar și o problemă bine condiționată.

1.5 Cum este organizat cursul?

Ordinea capitolelor urmărește dependența logică dintre noțiuni. Începem cu recapitularea matricelor simetrice, a valorilor proprii și a pozitivei definiri. Continuăm cu metode directe pentru sisteme liniare: substituție, eliminare Gauss, pivotare, factorizările LU , LDL^T și Cholesky, precum și matricile cu structură de bandă. Abia apoi introducem normele, raza spectrală, condiționarea și analiza erorilor.

Pe această bază studiem metodele iterative Jacobi, Gauss–Seidel, JOR și SOR și demonstrăm criterii de convergență. Urmează sistemele dreptunghiulare, problema celor mai mici pătrate, regularizarea și factorizarea QR . Problemele liniare sunt apoi reinterpretate ca probleme de optimizare, ceea ce conduce la metode de descreștere, metoda gradientului și metoda Gauss–Newton. Ultimele capitole dezvoltă descompunerea în valori singulare, pseudoinversa și aproximațiile de rang mic, cu aplicații în compresia imaginilor. Pe această bază introducem analiza componentelor principale (PCA), unde aceleași instrumente spectrale sunt folosite pentru

reducerea dimensiunii, aproximarea datelor prin subspații afine și analiza variației. Numai după parcurgerea acestor instrumente revenim la aplicații integratoare de tipul analizei rețelelor și al recomandării.

Ce trebuie să poată face studentul

La finalul cursului, studentul trebuie să poată trece în ambele direcții între matematică și algoritm: să deducă pașii unei metode din relațiile matematice, dar și să explice matematic un algoritm scris în MATLAB. La problemele mici, pașii trebuie să poată fi executați cu pixul pe hârtie; la problemele mai mari, implementarea trebuie însoțită de criterii de oprire, verificări prin reziduu, estimări de eroare și o discuție privind costul și stabilitatea.

Întrebări de început

1. De ce formula $x = A^{-1}b$ nu specifică încă un algoritm numeric?
2. Dați un exemplu în care structura matricei ar trebui să schimbe succesiunea de calcule folosită pentru rezolvarea sistemului.
3. De ce trebuie să distingem între problema matematică, algoritm și implementare?
4. De ce o matrice rară nu ar trebui, în general, stocată ca și cum toate elementele sale ar fi nenule?
5. Ce informații despre date, dimensiune și structură ar trebui notate înainte de a începe implementarea MATLAB?

Sarcini de aprofundare și orientare

Probleme de început: model, date și calcul

1. Alegeți una dintre aplicațiile discutate în capitol (procesare de imagini, recomandări, analiza rețelelor, ajustarea datelor). Identificați separat: datele de intrare, necunoscutele, rezultatul urmărit și etapa în care apare calculul numeric.
2. Pentru un sistem liniar simplu $Ax = b$, explicați diferența dintre a scrie o formulă matematică pentru soluție și a descrie o succesiune efectivă de operații care o calculează.
3. Dați câte un exemplu de matrice diagonală, triunghiulară, simetrică și rară. Pentru fiecare, indicați ce informație structurală este vizibilă înainte de efectuarea calculelor.
4. Comparați stocarea unei matrice 1000×1000 dense cu stocarea unei matrice de aceeași dimensiune având numai trei diagonale nenule. Numărați elementele care trebuie memorate în fiecare caz.
5. Construiți un exemplu în care aceeași problemă poate fi formulată cu 10, 100 și 1000 de necunoscute. Discutați ce se schimbă în volumul de calcul și ce rămâne neschimbat în model.
6. Formulați, în maximum cinci rânduri, diferența dintre model matematic, date, algoritm și implementare.

Laborator de orientare: aceeași structură la dimensiuni diferite

- Pași obligatorii de lucru.** 1. Refaceți mai întâi pe hârtie relațiile și cel puțin un pas numeric.
2. Scrieți algoritmul în pseudocod și precizați datele de intrare, ieșire și criteriul de oprire, dacă

este cazul. **3.** Implementați în MATLAB fără a apela direct rutina care rezolvă exact sarcina studiată, exceptând etapa de verificare. **4.** Testați pe un caz cu rezultat cunoscut. **5.** Comparați cu o rutină MATLAB de referință folosind numai mărimi și criterii definite până în acel punct al cursului. **6.** Formulați în câteva propoziții ce a făcut algoritmul și de ce rezultatul este credibil.

Obiectiv. Observați rolul structurii și al dimensiunii înainte de introducerea instrumentelor de analiză a erorilor.

1. Construiți în MATLAB matricea tridiagonală cu 2 pe diagonală și -1 pe diagonalele vecine, pentru $n = 10, 100, 1000$.
2. Stocați matricea atât dens, cât și cu `sparse` și comparați memoria raportată de `whos`.
3. Reprezentați structura cu `spy(A)` și explicați ce informație oferă imaginea.
4. Alegeți vectorul $x_\star = \mathbf{1}$, formați $b = Ax_\star$ și folosiți `A\b` pentru a obține un vector calculat $\hat{x}$.
5. Comparați direct componentele lui $\hat{x}$ cu cele ale lui $x_\star$ și raportați cea mai mare diferență absolută dintre componente, fără a introduce încă noțiuni de normă sau condiționare.
6. Măsurați timpul pentru forma densă și forma rară și explicați diferențele observate.

Test de capitol

Test – Introducere în calcul științific

1. Analiză și justificare. Pentru fiecare dintre situațiile următoare precizați dacă afirmația se referă în primul rând la *problema matematică*, la *algoritm* sau la *implementare* și justificați răspunsul:

- a) o mică modificare a datelor produce o modificare mare a soluției;
- b) două programe care implementează aceeași metodă furnizează rezultate diferite deoarece unul conține o eroare de indexare;
- c) pentru aceeași problemă, o metodă exploatează faptul că matricea este triunghiulară, iar alta tratează matricea ca fiind generală.

2. Selectați afirmația/afirmațiile corecte și justificați.

- a) Pentru un sistem $Ax = b$, formula $x = A^{-1}b$ este:
 - A) întotdeauna un algoritm numeric complet;
 - B) o formulă matematică ce nu precizează în sine succesiunea efectivă de calcule;
 - C) o formulă care exploatează automat faptul că A poate fi rară sau structurată;
 - D) aplicabilă și dacă A nu este inversabilă.
- b) O implementare numerică serioasă trebuie să țină seama de:
 - A) structura datelor;
 - B) dimensiunea problemei;
 - C) verificarea rezultatului;
 - D) numai numărul de linii ale programului.

Recapitulare: matrice, valori proprii și pozitivă definită

Matrice simetrice pozitiv definite

Pozitivă definire a unei matrice

- Spunem că matricea $A \in \mathbb{R}^{n \times n}$ este simetrică dacă $A = A^T$,
- Spunem că matricea simetrică $A \in \mathbb{R}^{n \times n}$ este **pozitiv semidefinită**, notăm $A \succeq 0$, dacă $\langle Ax, x \rangle \geq 0$ pentru orice $x \in \mathbb{R}^n$, unde $\langle \cdot, \cdot \rangle$ reprezintă produsul scalar standard din $\mathbb{R}^n$.
- Spunem că matricea simetrică $A \in \mathbb{R}^{n \times n}$ este **pozitiv definită**, notăm $A \succ 0$, dacă $\langle Ax, x \rangle > 0$ pentru orice $x \in \mathbb{R}^n$, $x \neq 0$.

Valori proprii și vectori proprii

Fie $A \in \mathbb{R}^{n \times n}$. Un vector nenul $v \in \mathbb{C}^n$ se numește **vector propriu** pentru A dacă există $\lambda \in \mathbb{C}$ astfel încât

$$Av = \lambda v.$$

Scalarul λ se numește **valoarea proprie** corespunzătoare vectorului propriu v . În general, o matrice reală poate avea valori proprii complexe. Pentru o matrice simetrică reală, toate valorile proprii sunt însă reale.

Teoremă 2.0.1. Dacă $A \in \mathbb{R}^{n \times n}$ este simetrică și $Av = \lambda v$ pentru un vector $v \in \mathbb{C}^n \setminus \{0\}$, atunci $\lambda \in \mathbb{R}$.

Demonstrație. Pe $\mathbb{C}^n$ folosim produsul scalar hermitian standard

$$\langle z, w \rangle_{\mathbb{C}^n} = z^H w = \sum_{j=1}^n \bar{z}_j w_j,$$

unde $z^H = \bar{z}^T$ notează transpusa conjugată (sau adjuncta hermitiană) a vectorului z . Mai general, pentru o matrice complexă M , notația $M^H = \bar{M}^T$ reprezintă transpusa conjugată a lui M . Deoarece A este reală și simetrică, avem $A^H = A$.

Din $Av = \lambda v$ rezultă

$$v^H Av = \lambda v^H v.$$

Scalarul $v^H Av$ este real: într-adevăr,

$$\overline{v^H Av} = v^H A^H v = v^H Av.$$

Cum $v^H v = \|v\|_2^2 > 0$, obținem

$$\lambda = \frac{v^H Av}{v^H v} \in \mathbb{R}.$$

□

Teoremă 2.0.2 (Teorema spectrală pentru matrice reale simetrice). *Fie $A \in \mathbb{R}^{n \times n}$ simetrică. Atunci există o matrice ortogonală $Q \in \mathbb{R}^{n \times n}$ și o matrice diagonală reală $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$ astfel încât*

$$A = Q\Lambda Q^T.$$

Echivalent, $\mathbb{R}^n$ admite o bază ortonormală formată din vectori proprii ai lui A .

Demonstrație. Demonstrăm prin inducție după n . Pentru $n = 1$ afirmația este evidentă. Presupunem rezultatul adevărat în dimensiunea $n - 1$. Polinomul caracteristic al lui A are o rădăcină în $\mathbb{C}$, deci A are un vector propriu complex $v \neq 0$. Teorema precedentă arată că valoarea proprie corespunzătoare λ este reală. Scriind $v = x + iy$, cu $x, y \in \mathbb{R}^n$, din $Av = \lambda v$ obținem $Ax = \lambda x$ și $Ay = \lambda y$; cel puțin unul dintre x, y este nenul. Există deci un vector propriu real unitar q_1 .

Considerăm subspațiul ortogonal $q_1^\perp$. Acesta este invariant la acțiunea lui A : dacă $z \perp q_1$, atunci, folosind simetria,

$$q_1^T Az = (Aq_1)^T z = (\lambda q_1)^T z = 0,$$

deci $Az \in q_1^\perp$. Restricția lui A la $q_1^\perp$ este din nou simetrică. Prin ipoteza de inducție există o bază ortonormală $q_2, \dots, q_n$ a lui $q_1^\perp$ formată din vectori proprii ai restricției, deci și ai lui A . Matricea $Q = (q_1 | \dots | q_n)$ este ortogonală, iar $Q^T A Q$ este diagonală cu valorile proprii pe diagonală. Rezultă $A = Q\Lambda Q^T$. □

Valorile proprii ale unei matrice simetrice $A \in \mathbb{R}^{n \times n}$ vor fi notate cu

$$\lambda_1(A) \geq \lambda_2(A) \geq \dots \geq \lambda_n(A).$$

Cea mai mare valoare proprie va fi notată cu $\lambda_{\max}(A) = \lambda_1(A)$ și cea mai mică valoare proprie va fi notată cu $\lambda_{\min}(A) = \lambda_n(A)$.

Teoremă 2.0.3 (Pozitiva definire și valorile proprii). *Fie $A \in \mathbb{R}^{n \times n}$ simetrică. Atunci A este pozitiv semidefinită dacă și numai dacă toate valorile sale proprii sunt nenegative, iar A este pozitiv definită dacă și numai dacă toate valorile sale proprii sunt strict pozitive.*

Demonstrație. Prin teorema spectrală, $A = Q\Lambda Q^T$, cu Q ortogonală și $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$. Pentru $x \in \mathbb{R}^n$ punem $y = Q^T x$. Deoarece Q este ortogonală, $x \neq 0$ dacă și numai dacă $y \neq 0$, iar

$$x^T Ax = y^T \Lambda y = \sum_{i=1}^n \lambda_i y_i^2.$$

Dacă toate $\lambda_i \geq 0$, suma este nenegativă pentru orice x , deci A este pozitiv semidefinită; dacă toate $\lambda_i > 0$, suma este strict pozitivă pentru orice $x \neq 0$, deci A este pozitiv definită. Reciproc, alegând $x = q_i$, coloana a i -a a lui Q , obținem $x^T Ax = \lambda_i$. Prin urmare pozitiv semidefinire implică $\lambda_i \geq 0$, iar pozitiv definită implică $\lambda_i > 0$ pentru orice i . □

Teoremă 2.0.4 (Criteriul lui Sylvester). *Fie $A \in \mathbb{R}^{n \times n}$ simetrică și notați $A_k = A(1:k, 1:k)$. Atunci A este pozitiv definită dacă și numai dacă*

$$\det A_k > 0, \quad k = 1, \dots, n.$$

Demonstrație. Dacă A este pozitiv definită, atunci fiecare submatrice principală A_k este pozitiv definită: pentru $z \in \mathbb{R}^k \setminus \{0\}$ completăm z cu zerouri până la un vector $x \in \mathbb{R}^n$ și avem $z^T A_k z = x^T A x > 0$. Valorile proprii ale lui A_k sunt deci pozitive, iar $\det A_k$, produsul lor, este pozitiv.

Pentru reciproca folosim inducția după n . Cazul $n = 1$ este imediat. Presupunem că toți minorii principali conducători ai lui A sunt pozitivi. Atunci, pentru blocul $B = A_{n-1}$, toți minorii principali conducători sunt pozitivi; prin ipoteza de inducție, B este pozitiv definită și, în particular, inversabilă. Scriem

$$A = \begin{pmatrix} B & c \\ c^T & a \end{pmatrix}.$$

Cantitatea

$$s = a - c^T B^{-1} c$$

se numește complementul Schur al blocului B în A . Pentru a vedea de unde apare, efectuăm o eliminare pe blocuri:

$$\begin{pmatrix} I & 0 \\ -c^T B^{-1} & 1 \end{pmatrix} \begin{pmatrix} B & c \\ c^T & a \end{pmatrix} = \begin{pmatrix} B & c \\ 0 & s \end{pmatrix}.$$

Matricea din stânga cu care am înmulțit are determinantul 1, iar matricea din dreapta este triunghiulară pe blocuri. Prin urmare,

$$\det A = \det B s = \det B (a - c^T B^{-1} c).$$

Din ipoteză, $\det A > 0$ și $\det B > 0$, deci

$$s = a - c^T B^{-1} c > 0.$$

Rămâne să arătăm că această informație implică pozitivă definirea lui A . Pentru $x = (y, t)^T \in \mathbb{R}^n$ avem

$$x^T A x = y^T B y + 2t c^T y + a t^2.$$

Completând pătratul și folosind simetria lui B , obținem

$$\begin{aligned} (y + t B^{-1} c)^T B (y + t B^{-1} c) + s t^2 &= y^T B y + 2t c^T y + t^2 c^T B^{-1} c \\ &\quad + t^2 (a - c^T B^{-1} c) \\ &= x^T A x. \end{aligned}$$

Deoarece B este pozitiv definită, primul termen este nenegativ, iar $s > 0$. Dacă $t \neq 0$, termenul $s t^2$ este strict pozitiv. Dacă $t = 0$ și $x \neq 0$, atunci $y \neq 0$, iar $y^T B y > 0$. Așadar

$$x^T A x > 0 \quad \text{pentru orice } x \neq 0,$$

deci A este pozitiv definită. □

O matrice $A \in \mathbb{R}^{n \times n}$ se numește diagonal dominantă pe linii dacă

$$|a_{ii}| \geq \sum_{j=1, j \neq i}^n |a_{ij}|, \quad \text{pentru } i = 1, \dots, n. \quad (2.1)$$

O matrice $A \in \mathbb{R}^{n \times n}$ se numește diagonal dominantă pe coloane dacă

$$|a_{ii}| \geq \sum_{j=1, j \neq i}^n |a_{ji}|, \quad \text{pentru } i = 1, \dots, n, \quad (2.2)$$

Dacă inegalitățile sunt stricte, spunem că A este strict diagonal dominantă (pe linii, respectiv, pe coloane).

Teoremă 2.0.5. *O matrice simetrică strict diagonal dominantă cu elemente strict pozitive pe diagonală este pozitiv definită.*

Demonstrație. Fie $A = (a_{ij})$ simetrică, cu $a_{ii} > \sum_{j \neq i} |a_{ij}|$ pentru fiecare i . Pentru $x \in \mathbb{R}^n$ avem, prin simetrie,

$$x^T Ax = \sum_{i=1}^n a_{ii} x_i^2 + 2 \sum_{1 \leq i < j \leq n} a_{ij} x_i x_j.$$

Folosind $2|x_i x_j| \leq x_i^2 + x_j^2$, obținem

$$2a_{ij} x_i x_j \geq -2|a_{ij}| |x_i x_j| \geq -|a_{ij}|(x_i^2 + x_j^2).$$

Prin însumare,

$$x^T Ax \geq \sum_{i=1}^n \left(a_{ii} - \sum_{j \neq i} |a_{ij}| \right) x_i^2.$$

Fiecare coeficient dintre paranteze este strict pozitiv. Pentru $x \neq 0$ există cel puțin un $x_i \neq 0$, deci partea dreaptă este strict pozitivă. Prin urmare A este pozitiv definită. $\square$

Reciproca nu este valabilă. De exemplu,

$$A = \begin{pmatrix} 1 & 2 \\ 2 & 5 \end{pmatrix}$$

este simetrică și pozitiv definită, deoarece minorii principali conducători sunt $1 > 0$ și $\det A = 1 > 0$, dar prima linie nu este diagonal dominantă: $|a_{11}| = 1 < 2 = |a_{12}|$. Așadar, diagonal-dominanța strictă cu diagonală pozitivă este o condiție suficientă, nu necesară, pentru pozitivă definire.

Extensie didactică: interpretarea spectrală a matricelor

Pentru metodele numerice, valorile proprii nu sunt doar obiecte algebrice. Ele descriu direcții în care acțiunea unei matrice are o formă simplă și vor reapărea în capitolele următoare în analiza mai multor algoritmi.

Matrice simetrică pozitiv definită

Pentru o matrice simetrică A , condiția $x^T Ax > 0$ pentru orice $x \neq 0$ înseamnă că forma pătratică asociată are curbura pozitivă în orice direcție. Prin teorema spectrală, $A = Q\Lambda Q^T$, unde Q este ortogonală și $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$. Dacă scriem $y = Q^T x$, atunci $x^T Ax = \sum_i \lambda_i y_i^2$. De aici se vede imediat echivalența dintre pozitivă definire și condiția $\lambda_i > 0$ pentru toate valorile proprii.

Raportul Rayleigh $R_A(x) = \frac{x^T A x}{x^T x}$, definit pentru $x \neq 0$, verifică $\lambda_{\min}(A) \leq R_A(x) \leq \lambda_{\max}(A)$ pentru orice matrice simetrică. El permite astfel să legăm proprietățile geometrice ale formei pătratice de spectrul matricei. În cazul unei funcții pătratice $f(x) = \frac{1}{2}x^T A x - b^T x$, aceste două valori proprii extreme vor determina cât de alungite sunt curbele de nivel și, ulterior, cât de repede converge metoda gradientului.

Exemplu ghidat

Fie $A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$. Vectorii $q_1 = (1, 1)^T/\sqrt{2}$ și $q_2 = (1, -1)^T/\sqrt{2}$ sunt vectori proprii pentru valorile proprii 3, respectiv 1. Pentru $x = \alpha q_1 + \beta q_2$ avem $x^T A x = 3\alpha^2 + \beta^2$. Pozitivitatea este transparentă în aceste coordonate. În plus, raportul dintre cea mai mare și cea mai mică valoare proprie este 3; pentru moment îl interpretăm numai ca pe o măsură geometrică a diferenței dintre cele două axe ale elipsei.

De ce contează ortogonalitatea?

Transformările ortogonale păstrează produsul scalar și lungimile euclidiene. Această proprietate geometrică va fi exploatată ulterior de factorizări numerice bazate pe transformări ortogonale; aici ne limităm la interpretarea ei algebrică și geometrică.

Laborator: spectru și formă pătratică

Pași obligatorii de lucru. 1. Refaceți mai întâi pe hârtie relațiile și cel puțin un pas numeric. 2. Scrieți algoritmul în pseudocod și precizați datele de intrare, ieșire și criteriul de oprire, dacă este cazul. 3. Implementați în MATLAB fără a apela direct rutina care rezolvă exact sarcina studiată, exceptând etapa de verificare. 4. Testați pe un caz cu rezultat cunoscut. 5. Comparați cu o rutină MATLAB de referință folosind numai mărimi și criterii definite până în acel punct al cursului. 6. Formulați în câteva propoziții ce a făcut algoritmul și de ce rezultatul este credibil.

Alegeți o matrice simetrică 2×2 pozitiv definită. Calculați valorile proprii cu `eig`, apoi reprezentați curbele de nivel ale lui $f(x) = \frac{1}{2}x^T A x$. Rotiți sistemul de coordonate cu matricea vectorilor proprii și verificați că în noile coordonate curbele de nivel sunt elipse cu axele aliniate cu axele coordonatelor. Modificați raportul $\lambda_{\max}/\lambda_{\min}$ și observați cum se alungesc elipsele.

Exerciții și verificarea înțelegerii

1. Demonstrați că o matrice simetrică A este pozitiv definită dacă și numai dacă toate valorile sale proprii sunt strict pozitive.
2. Pentru $A = \begin{pmatrix} 4 & -1 \\ -1 & 1 \end{pmatrix}$ verificați pozitivă definire prin două metode: criteriul minorilor principali și calculul valorilor proprii.
3. Arătați că, pentru $A = A^T$, maximul raportului Rayleigh este $\lambda_{\max}(A)$ și minimul este $\lambda_{\min}(A)$.
4. Construiți o matrice nesimetrică ale cărei valori proprii sunt pozitive, dar pentru care $x^T A x$ nu este pozitiv pentru orice $x \neq 0$. Explicați de ce ipoteza de simetrie este necesară în afirmația anterioară.

Mini-proiect: un model elementar cu arcuri

Etapele proiectului. 1. Modelare: definiți toate mărimile și termenii specifici aplicației înainte de a-i folosi și formulați problema matematică. **2. Derivare:** justificați metoda și executați manual un exemplu mic. **3. Algoritm:** prezentați pseudocodul și analizați costul. **4. MATLAB:** scrieți o implementare proprie, modulară. **5. Experimente:** variați cel puțin un parametru și comparați cu o metodă de referință. **6. Validare:** folosiți numai criterii definite până în acel punct al cursului. **7. Explicație:** includeți o concluzie matematică, nu doar grafice sau tabele.

Considerăm n puncte materiale care se pot deplasa numai pe o dreaptă, cu deplasări $x_1, \dots, x_n$, legate succesiv prin arcuri identice. Pentru un arc, presupunem legea liniară $F = -k\delta$: forța este opusă alungirii δ și proporțională cu ea. Capetele lanțului sunt fixate. Scrierea echilibrului forțelor pentru fiecare punct conduce la un sistem $Kx = f$. În acest context, numim *matrice de rigiditate* matricea K ce leagă deplasările de forțele elastice. După împărțirea prin k , ea este tridiagonală, cu 2 pe diagonală și -1 pe diagonalele vecine. Deduceți această matrice din ecuațiile de echilibru, apoi studiați numeric valorile proprii pentru mai multe valori ale lui n și verificați pozitivă definire. Interpretarea fizică a vectorilor proprii este opțională; din punct de vedere matematic, urmăriți forma lor și variația valorilor proprii.

Rezolvare orientativă**Pozitiva definire prin două metode**

Pentru $A = \begin{pmatrix} 4 & -1 \\ -1 & 1 \end{pmatrix}$, criteriul minorilor principali dă $\Delta_1 = 4 > 0$ și $\Delta_2 = 4 - 1 = 3 > 0$, deci A este SPD. Valorile proprii sunt rădăcinile lui $\lambda^2 - 5\lambda + 3 = 0$, adică $\lambda_{1,2} = (5 \pm \sqrt{13})/2$, ambele pozitive. Cele două verificări exprimă aceeași proprietate, dar din perspective diferite. Pentru $x = (x_1, x_2)^T$, forma pătratică este $4x_1^2 - 2x_1x_2 + x_2^2$. Completând pătratul, obținem $3x_1^2 + (x_2 - x_1)^2 > 0$ pentru $x \neq 0$, o a treia demonstrație. În aplicații, alegerea demonstrației depinde de informația disponibilă: structură explicită, spectru sau factorizare.

Situație practică: energie elastică discretă

În modelul elementar cu arcuri descris mai sus, x_i reprezintă deplasările punctelor, iar f_i forțele exterioare. Matricea K care apare în relația de echilibru $Kx = f$ a fost numită matrice de rigiditate. Pentru trei deplasări interioare și arcuri identice (după o scalare cu constanta arcului), energia poate avea forma $E(x) = \frac{1}{2}x^T Kx - f^T x$, unde K este tridiagonală cu 2 pe diagonală și -1 pe vecini. Termenul $x^T Kx$ se poate rescrie ca sumă de pătrate ale diferențelor dintre deplasări și ale deplasărilor față de capetele fixate. De aici se vede direct că K este SPD. Condiția de staționaritate a acestei funcții pătratice conduce la $Kx = f$. Deoarece K este pozitiv definită, sistemul are soluție unică. Cholesky și, ulterior, metoda gradientului vor exploata exact această structură.

Fișă de seminar: geometria matricelor simetrice

Problema rezolvată 1: pozitivă definire prin trei criterii

Considerăm $A = \begin{pmatrix} 4 & 2 \\ 2 & 3 \end{pmatrix}$. Putem verifica pozitivă definire în trei moduri. Prin criteriul Sylvester, minorii principali sunt $4 > 0$ și $\det A = 8 > 0$. Prin completare de pătrate, $x^T A x = 4x_1^2 + 4x_1x_2 + 3x_2^2 = (2x_1 + x_2)^2 + 2x_2^2 > 0$ pentru $x \neq 0$. Prin valori proprii, polinomul caracteristic este $\lambda^2 - 7\lambda + 8$, ale cărui rădăcini $(7 \pm \sqrt{17})/2$ sunt pozitive. Cele trei teste exprimă aceeași proprietate, dar sunt utile în contexte diferite.

Pentru calcul numeric, nu vom calcula de obicei valorile proprii doar pentru a decide dacă o matrice SPD admite Cholesky; algoritmul Cholesky însuși poate servi drept test practic. În demonstrații, însă, perspectiva spectrală explică direct geometria formei pătratice.

Problema rezolvată 2: o schimbare ortogonală de coordonate

Dacă $Q^T Q = I$, atunci produsul scalar este păstrat: $(Qx)^T (Qy) = x^T y$. În particular, unghiurile și lungimile euclidiene nu se schimbă. Această observație explică de ce diagonalizarea ortogonală a unei matrice simetrice păstrează fidel geometria problemei.

Laborator integrator 2: spectru și curbe de nivel

Pași obligatorii de lucru. **1.** Refaceți mai întâi pe hârtie relațiile și cel puțin un pas numeric. **2.** Scrieți algoritmul în pseudocod și precizați datele de intrare, ieșire și criteriul de oprire, dacă este cazul. **3.** Implementați în MATLAB fără a apela direct rutina care rezolvă exact sarcina studiată, exceptând etapa de verificare. **4.** Testați pe un caz cu rezultat cunoscut. **5.** Comparați cu o rutină MATLAB de referință folosind numai mărimi și criterii definite până în acel punct al cursului. **6.** Formulați în câteva propoziții ce a făcut algoritmul și de ce rezultatul este credibil.

Generați matrici simetrice 2×2 cu valori proprii prescrise și vectori proprii roțiți. Reprezentați curbele $x^T A x = 1$. Modificați raportul valorilor proprii de la 1 la 10^4 și observați elipsele. Conectați forma geometrică cu raportul dintre valorile proprii extreme.

Probleme pentru seminar

1. Arătați că o matrice SPD este nesingulară.
2. Demonstrați că suma a două matrici SPD este SPD.
3. Dacă A este SPD și Q ortogonală, arătați că $Q^T A Q$ este SPD și are aceleași valori proprii.
4. Construiți o matrice simetrică indefinită cu determinant pozitiv. De ce determinantul singur nu decide pozitivă definire în dimensiune mai mare?
5. Pentru $A = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$ determinați intervalul lui ρ pentru care A este SPD.
6. Calculați valorile proprii în problema precedentă și studiați ce se întâmplă cu cea mai mică valoare proprie când $|\rho| \rightarrow 1$.
7. Pentru o matrice simetrică diagonalizată ortogonal, explicați cum se citește acțiunea matricei pe fiecare direcție proprie.
8. Explicați de ce o matrice Gram $G = X^T X$ poate fi singulară chiar dacă toate elementele diagonale sunt pozitive.

Mini-proiect: atlas de matrici

Etapele proiectului. **1. Modelare:** definiți toate mărimile și termenii specifici aplicației înainte de a-i folosi și formulați problema matematică. **2. Derivare:** justificați metoda și executați manual un exemplu mic. **3. Algoritm:** prezentați pseudocodul și analizați costul. **4. MATLAB:** scrieți o implementare proprie, modulară. **5. Experimente:** variați cel puțin un parametru și comparați cu o metodă de referință. **6. Validare:** folosiți numai criterii definite până în acel punct al cursului. **7. Explicație:** includeți o concluzie matematică, nu doar grafice sau tabele.

Construiți un script MATLAB cu exemple pentru cel puțin opt clase: diagonală, ortogonală, SPD, semidefinită, indefinită, strict diagonal dominantă, singulară și aproape singulară. Pentru fiecare afișați determinantul, rangul și valorile proprii și verificați proprietățile structurale relevante folosind numai criteriile introduse în acest capitol.

Sarcini de aprofundare și sinteză

Probleme de aprofundare: spectru, forme pătraticе și geometrie

1. Pentru $A = \begin{pmatrix} 4 & 1 \\ 1 & 3 \end{pmatrix}$ calculați valorile proprii și o bază ortonormală de vectori proprii. Scrieți diagonalizarea ortogonală $A = Q\Lambda Q^T$.
2. Construiți o matrice simetrică 2×2 cu valori proprii prescrise 1 și 100, ale cărei direcții proprii sunt obținute printr-o rotație de 30° . Scrieți explicit Q , Λ și $A = Q\Lambda Q^T$.
3. Demonstrați că pentru orice matrice B cu rang coloană complet, matricea Gram $B^T B$ este pozitiv definită.
4. Dați o matrice reală 2×2 cu valori proprii complexe. Explicați de ce aceasta nu contrazice teorema spectrală pentru matrici reale simetrice.
5. Demonstrați, folosind teorema spectrală, că o matrice simetrică este pozitiv definită dacă și numai dacă toate valorile proprii sunt strict pozitive.
6. Pentru o matrice simetrică A , arătați că valorile expresiei $x^T A x$ pe vectori unitari sunt cuprinse între cea mai mică și cea mai mare valoare proprie.
7. Verificați criteriul lui Sylvester pentru trei matrici simetrice alese astfel încât una să fie pozitiv definită, una pozitiv semidefinită și una indefinită. Comparați cu semnele valorilor proprii.
8. Construiți o matrice simetrică strict diagonal dominantă cu diagonală pozitivă și verificați direct pozitivă definire. Construiți apoi o matrice pozitiv definită care nu este diagonal dominantă.
9. Considerați matricea

$$L = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix}.$$

Calculați valorile proprii și identificați vectorul propriu asociat valorii proprii zero. Explicați algebric de ce suma elementelor de pe fiecare linie fiind zero implică existența acestui vector propriu. Interpretări ale unei asemenea matrice prin grafuri vor fi introduse numai ulterior.

10. Comparați două matrici similare A și SAS^{-1} . Verificați că au aceleași valori proprii, dar vectorii proprii se transformă prin S .

Laborator: matrice simetrice cu spectru prescris

Pași obligatorii de lucru. 1. Refaceți mai întâi pe hârtie relațiile și cel puțin un pas numeric. 2. Scrieți algoritmul în pseudocod și precizați datele de intrare, ieșire și criteriul de oprire, dacă este cazul. 3. Implementați în MATLAB fără a apela direct rutina care rezolvă exact sarcina studiată, exceptând etapa de verificare. 4. Testați pe un caz cu rezultat cunoscut. 5. Comparați cu o rutină MATLAB de referință folosind numai mărimi și criterii definite până în acel punct al cursului. 6. Formulați în câteva propoziții ce a făcut algoritmul și de ce rezultatul este credibil.

Obiectiv. Construiți exemple pentru care spectrul este cunoscut dinainte și verificați teorema spectrală.

1. În dimensiunea 2, luați explicit matricea de rotație $Q(\theta)$ și $\Lambda = \text{diag}(1, 10)$. Formați $A = Q\Lambda Q^T$ pentru mai multe valori ale lui θ .
2. Verificați numeric că A este simetrică și că $AQ = Q\Lambda$, comparând element cu element cele două matrici.
3. Reprezentați curbele $x^T A x = 1$ pentru cel puțin trei rapoarte diferite între valorile proprii și explicați schimbarea geometrică.

4. Generați mai mulți vectori unitari x și calculați $x^T Ax$. Verificați experimental intervalul determinat de valorile proprii extreme.
5. Repetați construcția în dimensiunea 3 folosind o matrice ortogonală simplă construită din rotații plane succesive.
6. Formulați o concluzie despre relația dintre valorile proprii, direcțiile proprii și geometria formei pătratice.

Mini-proiect: axele unui nor de puncte

Etapele proiectului. **1. Modelare:** definiți toate mărimile și termenii specifici aplicației înainte de a-i folosi și formulați problema matematică. **2. Derivare:** justificați metoda și executați manual un exemplu mic. **3. Algoritm:** prezentați pseudocodul și analizați costul. **4. MATLAB:** scrieți o implementare proprie, modulară. **5. Experimente:** variați cel puțin un parametru și comparați cu o metodă de referință. **6. Validare:** folosiți numai criterii definite până în acel punct al cursului. **7. Explicație:** includeți o concluzie matematică, nu doar grafice sau tabele.

Nivel: mediu. Considerați punctele

$$(-3, -2), (-2, -1), (-1, 0), (1, 0), (2, 1), (3, 2),$$

al căror centru este originea, și formați matricea $X \in \mathbb{R}^{6 \times 2}$ având aceste puncte pe linii. Definiți $C = X^T X$.

1. Calculați C și verificați că este simetrică și pozitiv semidefinită. Justificați direct relația $z^T C z = \|Xz\|_2^2$.
2. Calculați valorile proprii și o bază ortonormală de vectori proprii ai lui C .
3. Reprezentați punctele în plan și trasați prin origine cele două axe determinate de vectorii proprii. Comparați orientarea lor cu forma norului de puncte.
4. Rotiți toate punctele cu același unghi, folosind o matrice ortogonală Q , și formați noua matrice $\tilde{C}$. Verificați că valorile proprii nu se schimbă, iar axele se rotesc odată cu punctele.
5. Modificați coordonatele astfel încât norul să fie mai alungit și urmăriți raportul dintre cea mai mare și cea mai mică valoare proprie.

Test de capitol

Test – Matrice, valori proprii și pozitivă definită

1. Calcul explicit. Se consideră

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}.$$

- a) Determinați valorile proprii și câte un vector propriu pentru fiecare dintre ele.
- b) Construiți o bază ortonormală de vectori proprii și scrieți descompunerea spectrală $A = Q\Lambda Q^T$.
- c) Decideți, justificat în două moduri diferite, dacă A este pozitiv definită.

2. Selectați afirmația/afirmațiile corecte și justificați. Pentru o matrice reală simetrică A :

- A) toate valorile proprii sunt reale;
- B) vectorii proprii corespunzători unor valori proprii distincte pot fi aleși ortogonali;
- C) A este pozitiv definită dacă și numai dacă toate valorile proprii sunt strict pozitive;
- D) orice matrice cu determinant pozitiv este pozitiv definită.