Numerická lineární algebra

- mnoho úloh ve fyzice (zvláště pokud je systém popsán parciálními diferenciálními rovnicemi) vede nakonec k numerickému řešení jedné ze tří základních úloh lineární algebry

  1) řešení soustavy lineárních rovnic $Ax=b$

  2) problém hledání vlastních čísel a vektorů

     $Ax=\lambda x$, neboli hledáme rozklad $A=V\Lambda V^{-1}$

  3) řešení přeurčené soustavy $Ax=b$ vedoucí na metodu nejmenších čtverců

- při řešení těchto úloh se s výhodou používají různé faktorizace (rozklady, dekompozice) příslušné matice A na součin 2-3 speciálních matic, pomocí kterých už pak může být určitá úloha snadno (a rychle) řešitelná

- základní rozklady, kterými se budeme podrobněji zabývat, jsou

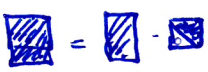
  a) LU rozklad (dekompozice) používaná k řešení 1)

  $A = L \cdot U$, kde L je dolní trojúhelníková a U je horní trojúhelníková matice

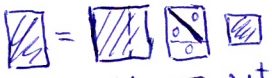
  b) QR rozklad používaný pro 2) a 3), ale lze i pro 1), jen je 2x pomalejší než LU rozklad

  $A = Q \cdot R$

  , kde Q je ortogonální (obecně $m \times n$) matice a R je horní trojúhelníková matice

  c) SVD - singulární rozklad matice (singular value decomposition)

  $A = U \cdot \Sigma \cdot V^{+}$

  kde Σ je diagonální matice obsahující tzv. singulární hodnoty a U a V jsou ortogonální (unitární) matice

  - lze použít téměř na cokoli, jen je mnohem pomalejší pro praktické výpočty než předchozí rozklady, ovšem nejstabilnější

- v praktických problémech rozhoduje především rychlost (efektivita) určité metody řešení a dále stabilita (správnost a přesnost řešení jde většinou ruku v ruce se stabilitou)
  - bohužel velmi často jsou stabilnější metody méně efektivní a je nutný určitý kompromis založený na zkušenosti z praxe
  - např. pro řešení $Ax=b$ je pořadí efektivity

    $$LU < QR < SVD$$

    ale pokud jde o stabilitu, je to naopak
    v praxi se ale ukazuje, že pro většinu systémů $Ax=b$ vzešlých např. z diskretizace PDR je bezpečné z hlediska přesnosti použít LU rozklad
- přestože si ukážeme, jak některé metody fungují a jak je lze využít k řešení problémů 1), 2) a 3), tak v praxi velmi doporučuji použít některou z dostupných knihoven, případně programy jako Mathematica nebo Matlab, kde jsou všechny výše uvedené metody a problémy vyladěné z hlediska efektivity a i přesnosti (viz pár odkazů na mé webové stránce)
  - zvláště upozorňuji na knihovnu LAPACK a její implementaci v Intel MKL knihovně
    - optimalizace nejen z hlediska algoritmů, ale i z hlediska hardwaru (využití rychlé paměti na procesorech)
  - též je potřeba uvážit, zda váš konkrétní problém nevede na jeden z řady speciálních typů matic (tridiagonální, pásové, řídké apod.) pro které existují specializované metody, často řádově efektivnější než pro obecné matice
  - někdy (zvláště pro tzv. řídké matice) se s výhodou používají iterační metody (implementované např. v knihovnách PARDISO či JADAMILU) probírané v letním semestru

Gaussova eliminace a LU dekompozice (rozklad)

- Gauss. eliminace převádí systém $Ax=b$, kde $A\in\mathbb{C}^{m\times m}$ na systém $Ux=\bar{b}$, kde U je horní $\triangle$ matice a $\bar{b}$ je modifikované b, pomocí operací eliminujících prvky pod diagonálou

- postupné řádkové eliminace v jednom sloupci jsou ovšem ekvivalentní násobení matice A dolní $\triangle$ maticí:

$$A^{(1)} = L_1 A$$

neboli

$$\underbrace{\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{m1} & \cdots & & a_{mm} \end{pmatrix}}_{A=A^{(0)}} \xrightarrow{L_1} \underbrace{\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ 0 & \tilde{a}_{22} & \cdots & \tilde{a}_{2m} \\ \vdots & & & \\ 0 & \tilde{a}_{m2} & \cdots & \tilde{a}_{mm} \end{pmatrix}}_{A^{(1)}}, \text{ kde } L_1 = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ -l_{21} & 1 & & \vdots \\ \vdots & 0 & \ddots & \\ -l_{m1} & 0 & \cdots 0 & 1 \end{pmatrix}$$

$l_{j1} = a_{j1}/a_{11}$, $j>1$

a obdobně pro eliminaci k-tého sloupce použijeme

$$L_k = \begin{pmatrix} 1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \ddots & & 0 & & \vdots \\ \vdots & & 1 & & & \\ & 0 & & 1 & & \\ & & -l_{k+1,k} & & \ddots & 0 \\ 0 & \cdots 0 & -l_{mk} & 0 & & 1 \end{pmatrix}$$

k-tý sloupec

kde $l_{jk} = \dfrac{a_{jk}^{(k-1)}}{a_{kk}^{(k-1)}}$ pro $j>k$

a $a_{jk}^{(k-1)}$ jsou prvky matice $A^{(k-1)} = L_{k-1}\cdots L_1 A$

- dostáváme nakonec $L_{m-1}L_{m-2}\cdots L_1 A = U$

neboli

$$A = (L_{m-1}\cdots L_1)^{-1}U = L_1^{-1}\cdots L_{m-1}^{-1}U = LU$$

kde U je horní $\triangle$ matice a L je dolní $\triangle$ matice, neboť

$$L_k^{-1} = \begin{pmatrix} 1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots & & \vdots \\ & & 1 & 0 & & \\ & & 0 & 1 & & \\ & & & l_{k+1,k} & \ddots & 0 \\ 0 & \cdots & 0 & l_{mk} & 0 & 1 \end{pmatrix} \quad \text{a navíc} \quad L_1^{-1}L_2^{-1} = \begin{pmatrix} 1 & & & & \\ l_{21} & 1 & & 0 & \\ & l_{32} & 1 & & \\ & & & \ddots & \\ l_{m1} & l_{m2} & 0 & & 1 \end{pmatrix} \quad \text{atd.}$$

(pouze změna znamének v l_{jk})

Pozn.: 1) L_k lze zapsat pomocí „vnějšího součinu"

$$L_k = I - l_k e_k^T \qquad l_k e_k^T = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ l_{k+1,k} \\ \vdots \\ l_{m,k} \end{pmatrix} (0 \ldots 0\ 1\ 0 \ldots 0)$$

k-tý prvek

a protože skalární součin $e_k^T l_k = 0$,

bude $(I - l_k e_k^T)(I + l_k e_k^T) = I$ a tedy $L_k^{-1} = I + l_k e_k^T$

a dále $L_k^{-1}L_{k+1}^{-1} = (I + l_k e_k^T)(I + l_{k+1} e_{k+1}^T) = I + l_k e_k^T + l_{k+1} e_{k+1}^T$ atd.

2) protože má L na diagonále 1, lze L a U uložit do jediné matice, většinou se nahrazuje přímo matice A jejím LU rozkladem

3) takto přímočaře nelze použít (nestabilní), nutná tzv. pivotizace

Př. Nestabilita Gaussovy eliminace bez pivotizace

Řešme $Ax = b$ pro $A = \begin{pmatrix} 10^{-20} & 1 \\ 1 & 1 \end{pmatrix}$ a $b = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

Přímo:

$$10^{-20} x_1 + x_2 = 1 \quad \Rightarrow \quad x_1 = \frac{1}{10^{-20} - 1} \doteq -1 \quad , \quad x_2 = -x_1 \doteq 1$$
$$, \; x_1 + x_2 = 0$$

LU dekompozice

$$L = \begin{pmatrix} 1 & 0 \\ 10^{20} & 1 \end{pmatrix} \quad , \quad U = \begin{pmatrix} 10^{-20} & 1 \\ 0 & 1 - 10^{20} \end{pmatrix} \quad \Rightarrow \quad L \cdot U = A = \begin{pmatrix} 10^{-20} & 1 \\ 1 & 1 \end{pmatrix}$$

Ovšem v double precision:

$$\tilde{L} = \begin{pmatrix} 1 & 0 \\ 10^{20} & 1 \end{pmatrix} \quad \tilde{U} = \begin{pmatrix} 10^{-20} & 1 \\ 0 & -10^{20} \end{pmatrix} \Rightarrow \tilde{L}\tilde{U} = \begin{pmatrix} 10^{-20} & 1 \\ 1 & 0 \end{pmatrix} \neq A$$

Pro některé pravé strany (např. $b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$) to stále dá správné řešení, ale ne vždy! Právě pro $b = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ dostaneme:

$$\tilde{L}^{-1} = \begin{pmatrix} 1 & 0 \\ -10^{20} & 1 \end{pmatrix} \quad \tilde{U}^{-1} = \begin{pmatrix} 10^{20} & 1 \\ 0 & -10^{-20} \end{pmatrix} \Rightarrow \tilde{x} = \tilde{U}^{-1}\tilde{L}^{-1} b = \tilde{U}^{-1} \begin{pmatrix} 1 \\ -10^{20} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \neq \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

neboť Gauss. eliminace dělá něco jiného než dosazení za $x_2 = -x_1$:

$$\begin{array}{l|l} \cdot 10^{20} & 10^{-20} x_1 + x_2 = 1 \\ & x_1 + x_2 = 0 \\ \hline & 10^{-20} x_1 + x_2 = 1 \\ & (1 - 10^{20}) x_2 = -10^{20} \end{array} \quad \xrightarrow{\text{zaokrouhlení}} \quad -10^{20} x_2 = -10^{20} \Rightarrow x_2 = 1 \; !$$

a tedy $10^{-20} x_1 + 1 = 1 \Rightarrow x_1 = 0$!

Ovšem s pivotizací bude:

$$\begin{array}{l|l} \cdot 10^{-20} & x_1 + x_2 = 0 \\ & 10^{-20} x_1 + x_2 = 1 \\ \hline & x_1 + x_2 = 0 \\ & 0 + (1 - 10^{-20}) x_2 = 1 \end{array} \quad \xrightarrow{\text{zaokrouhlení}} \quad 1 x_2 = 1 \Rightarrow x_2 = 1$$

a tedy nyní $x_1 = -x_2 = -1$

- <u>pivotizace</u> - výběr největšího prvku, který dělíme
  - nejen pokud se na diagonále objeví nula, ale i tehdy když je tam velmi malé číslo relativně k ostatním, je nutné kvůli stabilitě přehodit řádky (a případně i sloupce) a pak pokračovat v Gauss. eliminaci
  - <u>řádková pivotizace</u> = hledáme největší $a_{jk}^{(k-1)}$ pro $j \geq k$ a přehazujeme řádky

    vs.

    <u>plná pivotizace</u> = hledáme největší $a_{jl}^{(k-1)}$ pro $j \geq k$ a $l \geq k$ a přehazujeme nejen řádky, ale i sloupce
  - ovšem plná pivotizace potřebuje $O(m^3)$ operací porovnání oproti $O(m^2)$ operací u řádkové pivotizace, proto se používá řádková pivotizace

    Pozn: zde $O(m^3)$ je celkový počet operací v průběhu celé Gaussovy eliminace

- <u>řádková pivotizace a LU rozklad</u>
  - prohazování řádků lze provést pomocí permutačních matic, např. $P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$ přehodí 2. a 4. řádek matice 4x4
  - lze tedy psát $L_{m-1} P_{m-1} L_{m-2} P_{m-2} \dots L_1 P_1 A = U$ s příslušnými perm. maticemi P_k pro řádkovou pivotizaci v k-tém kroku

    avšak toto lze přepsat pomocí

    $$L_k' = P_{m-1} \dots P_{k+1} L_k P_{k+1}^{-1} \dots P_{m-1}^{-1}, \quad L_{m-1}' = L_{m-1}$$

    jako

    $$\underbrace{(L_{m-1}' L_{m-2}' \dots L_1')}_{L^{-1}} \underbrace{(P_{m-1} P_{m-2} \dots P_1)}_{P} A = U$$

    neboli $PA = LU$ a kromě matic L a U je nutné ještě uložit celkovou permutační matici P (lze jako vektor)
  - matice L_k' má oproti L_k pouze přeuspořádané prvky l_{jk}, neboť $P_k^{-1} = P_k$ a P_k násobící A zprava přehazuje místo řádků příslušné sloupce, ovšem L_k násobíme pouze P_j pro $j > k$, takže se přehodí sloupce jednotkové podmatice, a pak to přehození řádků vrátí

[Pozn: obdobně lze přepsat i úplnou pivotizaci
jako $PAQ = LU$, kde P i Q jsou permutační matice
avšak v praxi se nepoužívá kvůli efektivitě]

- <u>algoritmus LU rozkladu s řádkovou pivotizací</u>

$U = A$, $L = I$, $P = I$ (inicializace)

pro $k = 1, m-1$ (eliminujeme $m-1$ sloupců)

najdi $i \geq k$, pro které je $|U_{ik}|$ maximální (pivotizace)

$$\left.\begin{array}{l} U_{k,k:m} \leftrightarrow U_{i,k:m} \\ \ell_{k,1:k-1} \leftrightarrow \ell_{i,1:k-1} \\ P_{k,:} \leftrightarrow P_{i,:} \end{array}\right\}$$ (přehození řádků, pokud $i \neq k$)

pro $j = k+1, m$

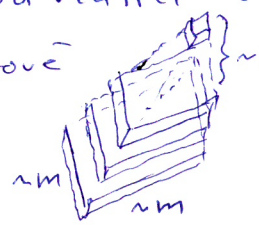
$\ell_{jk} = U_{jk} / U_{kk}$

$U_{j,k:m} = U_{j,k:m} - \ell_{jk} U_{k,k:m}$ (odečtení dvou řádků, časově nejnáročnější)

- <u>počet operací</u> ($+, -, *, /$ uvažujeme na stejné úrovni)

- když eliminujeme jeden sloupec (k-tý), tak musíme upravit podmatici $(m-k) \times (m-k)$ a sloupců je $m-1$

tedy obrazově



objem $\sim \frac{1}{3} m^3$

ovšem vždy musíme dělat 2 operace ($-, *$) viz posl. řádek

takže celkem je potřeba $\sim \frac{2}{3} m^3$ operací (tzv. flops = = floating point operations)

- řádková pivotizace je $\sim m^2$

znamená, že $\lim_{m \to \infty} \frac{\text{počet operací}}{\frac{2}{3} m^3} = 1$

- <u>řešení soustavy $Ax = b$ pomocí LU rozkladu</u>

- jakmile máme rozklad $PA = LU$, můžeme řešit ekvivalentní soustavu $PAx = Pb$ (tj. přeuspořádáme pravou stranu) a řešíme $Ly = Pb$ a pak $Ux = y$ přímo, neboť jde o trojúhelníkové matice (tzv. zpětná substituce)

- obě rovnice mají náročnost $\sim m^2$, proto se vyplatí udělat LU rozklad zvlášť tehdy, pokud se matice A nemění, ale máme různé pravé strany

Choleského rozklad

- jde o obdobu LU rozkladu, ale pro hermitovské, pozitivně definitní matice A, tj. $A = A^+$ a $x^+ A x > 0$ pro lib. $x \neq 0$

  (z pozitivní definitnosti např. plyne, že všechna vlastní čísla jsou kladná a též diagonální prvky)

- díky hermitovosti můžeme nejen nulovat při Gaussově eliminaci první sloupec násobením dolní Δ matice zleva, ale také první řádek násobením herm. sdružené matice zprava

  tj. A lze rozložit takto ($a_{11} > 0$)

$$A = \begin{pmatrix} a_{11} & w^+ \\ w & K \end{pmatrix} = \begin{pmatrix} \alpha & 0 \\ \frac{w}{\alpha} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \underbrace{K - \frac{ww^+}{\alpha^2}}_{\text{opět pozitivně definitní, hermitovská matice}} \end{pmatrix} \begin{pmatrix} \alpha & w^+/\alpha \\ 0 & 1 \end{pmatrix} = R_1^+ A_1 R_1, \text{ kde } \alpha = \sqrt{a_{11}}$$

(
$A_1 = (R_1^+)^{-1} A R_1^{-1}$ je též pozitivně definitní, herm. matice
a navíc platí, že lib. podmatice obsahující levý horní, nebo pravý dolní roh takovéto matice je též herm., pozit. definitní
– násobení – $X^+ A_1 X$, kde $X = \begin{pmatrix} 0 \\ 1_{n \times n} \end{pmatrix}$, vybereme takovou podmatici
)

a takto můžeme pokračovat dále rozklady podmatic až zůstane uprostřed jednotková matice, tj. máme

$$A = R_1^+ R_2^+ \dots R_m^+ R_m \dots R_2 R_1 = R^+ R, \text{ kde } r_{jj} > 0$$

a R je horní Δ matice

- tento rozklad je jednoznačný
- díky hermitovskosti matice A stačí uložit jen ∇ část a upravovat ji na R

$R = A$
pro $k = 1, m$
  pro $j = k+1, m$
    $R_{j,j:m} = R_{j,j:m} - R_{k,j:m} \overbrace{R^*_{kj} / R_{kk}}^{\text{toto se vyplatí uložit do pomocné proměnné, pak jen 2 operace, ale nyní je „objem" pouze } \sim \frac{1}{6} m^3}$
  $R_{k,k:m} = R_{k,k:m} / \sqrt{R_{kk}}$

takže celkem $\sim \frac{1}{3} m^3$

- stabilita Gaussovy eliminace, a LU rozkladu

- bez pivotizace: lze ukázat, že spočtené $\tilde{L}$ a $\tilde{U}$ splňují $\tilde{L}\tilde{U} = A + \delta A$, kde ovšem $\frac{\|\delta A\|}{\|L\|\|U\|} = \mathcal{O}(\varepsilon_M)$, pro určité δA

tj. není zde $\frac{\|\delta A\|}{\|A\|}$, ale záleží na velikosti L a U

pokud $\|A\| \ll \|L\|$ nebo $\|U\|$, pak může být $\|\delta A\|$ poměrně velké a spočtené $\tilde{L}\tilde{U}$ se může dost lišit od původní A

- říkáme, že LU rozklad není zpětně stabilní

a důsledkem je, že řešení $\tilde{L}\tilde{U}x = b$ pak může být velmi nepřesné

- pro LU rozklad spivotizací lze ukázat, že je zpětně stabilní,

tj. $\tilde{L}\tilde{U} = \tilde{P}A + \delta A$, kde $\frac{\|\delta A\|}{\|A\|} = \mathcal{O}(\rho\,\varepsilon)$

$\uparrow$ kvůli zaokrouhlovacím chybám nemusí jít nutně o P

- nyní již $\|\delta A\|$ porovnáváme s $\|A\|$, ale objevuje se tzv. růstový faktor (growth factor) $\rho = \frac{\max_{i,j}|u_{ij}|}{\max_{i,j}|a_{ij}|}$

který může být ve výjimečných případech až 2^m (viz ukázka v Mathematice), ale v praxi se s tímto případem s velkou pravděpodobností nesetkáte, u náhodně generovaných matic je spíše řádu $\sqrt{m}$

Pozn: Algoritmus $\tilde{f}$ je pro problém $f = f(x)$ zpětně stabilní, pokud pro $\forall x$ platí $\tilde{f}(x) = f(\tilde{x})$ pro určité $\tilde{x}$ splňující $\frac{\|\tilde{x} - x\|}{\|x\|} = \mathcal{O}(\varepsilon_M)$

neboli „Zpětně stabilní algoritmus dá správnou odpověď na skoro správnou otázku."

Navíc pro přesnost zpětně stab. alg. platí (viz např. Trefethen, Bau III: Numerical linear algebra)

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} = \mathcal{O}(\varkappa(x)\,\varepsilon_M)$$

kde $\varkappa(x)$ je číslo podmíněnosti, pro matice $\varkappa(A) = \|A\|\,\|A^{-1}\|$

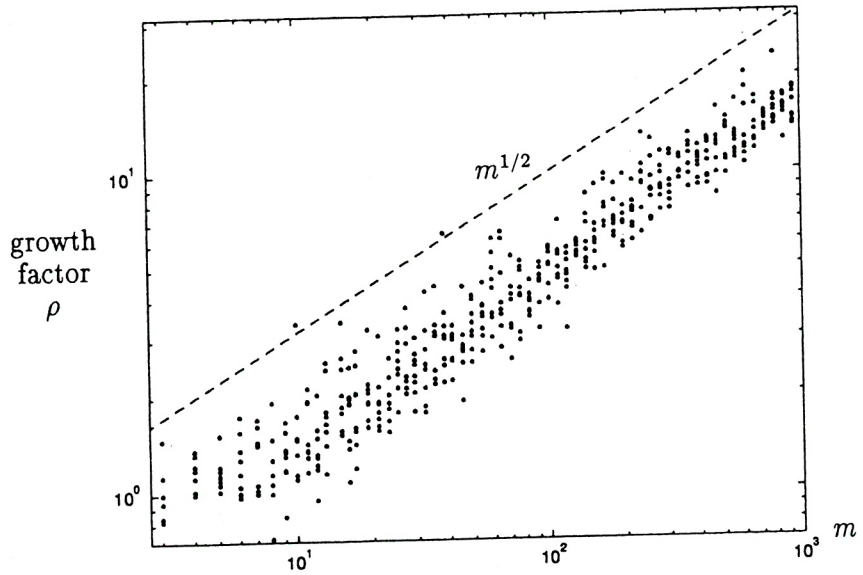


Figure 22.1. *Growth factors for Gaussian elimination with partial pivoting applied to* 496 *random matrices (independent, normally distributed entries) of various dimensions. The typical size of* ρ *is of order* $m^{1/2}$, *much less than the maximal possible value* 2^{m-1}.

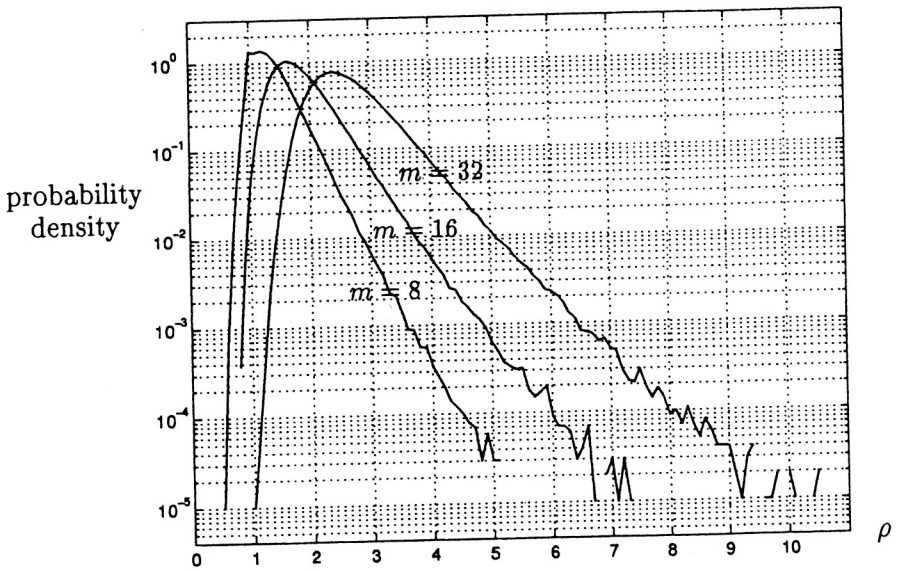


Figure 22.2. *Probability density distributions for growth factors of random matrices of dimensions* $m = 8, 16, 32$, *based on sample sizes of one million for each dimension. The density appears to decrease exponentially with* ρ. *The chatter near the end of each curve is an artifact of the finite sample sizes.*