

5.4. Feluppskattning vid lösning av ekvationssystem

Vi har tidigare påpekat, att "pivot"-elementen bör vara olika noll, för att man skall kunna tillämpa Gauss' elimineringsmetod. Men det kan också uppstå problem, om något av pivot-elementen är mycket litet. Vi skall se på följande exempel:

$$\begin{pmatrix} 10 & -7 & 0 \\ -3 & 2.099 & 6 \\ 5 & -1 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 3.901 \\ 6 \end{pmatrix} .$$

Den exakta lösningen till detta ekvationssystem är $(0, -1, 1)^T$, men om vi gör beräkningarna på en (hypotetisk) maskin som räknar med fem siffrors noggrannhet (avkortning) enligt Gauss' metod blir resultatet $(-0.35, -1.50, 0.99993)^T$! Orsaken till detta dåliga resultat är att ett av pivot-elementen var alltför litet, och detta resulterade i att de återstående ekvationerna blev multiplicerade med en stor faktor.

Man kan visa, att Gauss' elimineringsprocess är stabil, ifall alla faktorerna till sitt absoluta värde är mindre eller lika med 1. Genom att använda den partiella pivoteringsmetoden kan man se till att detta villkor alltid är uppfyllt. Metoden innebär, som vi redan sett, att man vid det k :te steget i elimineringsproceduren väljer det *största* elementet i den oreducerade delen av matrisens k :te kolonn som pivot-element.

Raden, som innehåller detta element, bytes därpå ut mot den k :te raden, så att pivot-elementet kommer i positionen (k, k) . Samma procedur tillämpas även på kolumnvektorn b . På grund av att A -matrisens kolonner inte permuteras, ändras inte heller ordningsföljden för de obekanta.

Om matrisen är nästan singular (determinanten mycket liten), så kan de beräknade värdena av de obekanta ha stora fel, fastän restvektorn (avvikelserna) är små. En karaktäristisk egenskap för Gauss' elimineringsmetod med partiell pivotering är just de små avvikelserna.

För att få en bättre uppfattning om felet av vektorstorheter, skall vi införa begreppet **vektornorm**, i analogi med funktionsnormerna, som vi tidigare studerat. Allmänt brukar man definiera (l_p) -normen av en vektor x i ett n -dimensionellt vektorrum som

$$\|x\|_p = (|x_1|^p + |x_2|^p + \dots + |x_n|^p)^{1/p}, \quad 1 \leq p < \infty.$$

Om $p = 2$, har vi speciellt en **euklidisk** norm, och om $p \rightarrow \infty$ får vi **maximum-normen**:

$$\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|.$$

En vektornorm uppfyller följande villkor (analogt med villkoren för avstånd mellan punkter i rummet):

$$\|x\| > 0 \quad \text{om} \quad x \neq 0; \quad \|x\| = 0 \quad \text{implikerar} \quad x = 0.$$

$$\|\alpha x\| = |\alpha| \|x\| \quad (\text{om } \alpha \text{ är en (komplex) skalar}).$$

$$\|x + y\| \leq \|x\| + \|y\| \quad \text{"triangelolikheten"}.$$

Också för matriser kan man definiera en norm med motsvarande egenskaper:

$$\|A\| > 0 \quad \text{om } A \neq 0; \quad \|A\| = 0 \quad \text{implikerar } A = 0.$$

$$\|\alpha A\| = |\alpha| \|A\| \quad (\text{om } \alpha \text{ är en (komplex) skalar}).$$

$$\|A + B\| \leq \|A\| + \|B\|; \quad \|AB\| \leq \|A\| \|B\|.$$

En matrisnorm och en vektornorm sägs vara **konsistenta** ifall $\|Ax\| \leq \|A\| \|x\|$ gäller för alla A och x . Mot varje vektornorm svarar en konsistent matrisnorm. Man kan tex definiera en matrisnorm utgående från en vektornorm på följande sätt:

$$\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|} = \max_{\|x\|=1} \|Ax\| \quad (\text{den naturliga normen}),$$

som också kan uttryckas

$$\|A\|_{\infty} = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}|,$$

eftersom den är relaterad till $\|x\|_\infty$. Dessa normer är enkla att handskas med, emedan

$$\| |A| \| = \|A\|, \quad \text{om} \quad |A| = (|a_{ij}|).$$

En annan matrisnorm är **spektralnomen**, som är baserad på l_2 -normen:

$$\|A\|_2 = (\text{max. egenvärde av } A^\dagger A)^{1/2} = \max_{\|x\|_2 \neq 0} \frac{\|Ax\|_2}{\|x\|_2}.$$

Motsvarigheten till l_2 -normen av en vektor är den *euklidiska* normen (eller **Frobenius**-normen)

$$\|A\|_E = \left(\sum_{i,j} |a_{ij}|^2 \right)^{1/2},$$

som har den fördelen, att den är lätt att beräkna. Jämför även `help norm` i MATLAB.

Vi skall nu tillämpa detta på beräkning av felet vid lösning av ekvationssystem. Antag, att A och b är kända i ekvationen $Ax = b$, och att dessa storheter utsätts för små störningar. Om b perturberas med en störningsterm δb , fås då

$$A(x + \delta x) = b + \delta b,$$

som ger

$$\delta x = A^{-1} \delta b.$$

Om vi tillämpar normens egenskaper, följer härav

$$\|\delta x\| \leq \|A^{-1}\| \|\delta b\|.$$

Genom perturbation av A fås motsvarigt

$$(A + \delta A)(x + \delta x) = b,$$

varav i sin tur följer

$$A\delta x + \delta A(x + \delta x) = 0.$$

Således gäller $\delta x = -A^{-1}\delta A(x + \delta x)$, och felvektorns norm kan då uppskattas till

$$\|\delta x\| \leq \|A^{-1}\| \|\delta A\| \|x + \delta x\|,$$

som kan skrivas om i formen

$$\frac{\|\delta x\|}{\|x + \delta x\|} \leq \text{cond}(A) \frac{\|\delta A\|}{\|A\|},$$

där $\text{cond}(A) = \|A\| \cdot \|A^{-1}\|$ betecknar **konditionstalet** för matrisen A med avseende på den givna normen. Se också `help cond` i MATLAB. Konditionstalet är alltid åtminstone 1. Den tvåradiga matrisen

$$\begin{pmatrix} 1.2969 & 0.8648 \\ 0.2161 & 0.1441 \end{pmatrix}$$

t.ex. har konditionstalet $3.27 \cdot 10^8$ i avs. på maximinormen (determinanten är 10^{-8}), vilket visar att konditionen är mycket dålig. Ett stort konditionstal innebär att matrisen är nästan singular. Konditionstalet är ett bättre mått på matrisens kondition än determinanten. Matlab har en funktion `cond`, som beräknar konditionstalet i avseende på den euklidiska normen.

Av olikheterna $\|\delta x\| \leq \|A^{-1}\| \|\delta b\|$ och $\|b\| = \|Ax\| \leq \|A\| \|x\|$ följer ytterligare, att

$$\frac{\|\delta x\|}{\|x\|} \leq \text{cond}(A) \frac{\|\delta b\|}{\|b\|}.$$

Dessa olikheter visar, att om $\text{cond}(A)$ är stort, så kan relativt små störningar i A och b åstadkomma stora störningar i x , och problemet är då inte välartat.

5.5. Iterativa metoder för glesa ekvationssystem

Om antalet obekanta är mycket stort, kan det vara omöjligt att lagra hela matrisen A i datorns minne. Ofta är matrisen dock gles, så att det räcker med att lagra endast de element som skiljer sig från 0, vilket kan vara betydligt lättare. För att lösa dylika glesa ekvationssystem kan man använda sig av *iterativa* metoder. Sådana ekvationssystem förekommer ofta i fysiken.

Betrakta systemet $Ax = b$, och antag, att $a_{ii} \neq 0$, $i = 1, 2, \dots, n$. Ekvationerna kan då skrivas i formen

$$x_i = \frac{-\sum_{j \neq i} a_{ij} x_j + b_i}{a_{ii}}, \quad i = 1, 2, \dots, n.$$

Enligt en metod, som härrör sig från *Jacobi*¹, så kan man beräkna en serie approximationer $(x^{(1)}, x^{(2)}, \dots)$ till Lösningsvektorn ur formeln

$$x_i^{(k+1)} = \frac{-\sum_{j \neq i} a_{ij} x_j^{(k)} + b_i}{a_{ii}}, \quad i = 1, 2, \dots, n.$$

¹Über eine neue Auflösungsart der bei der Methode der kleinsten Quadrate vorkommenden linearen Gleichungen, Astron. Nachr. XXXII, 297-306 (1845)

Utgångsvektorn är vanligen $x^{(0)} = 0$. Om man bildar gränsvärdet av vartdera medlemmarna då $k \rightarrow \infty$, så ser man att $\lim_{k \rightarrow \infty} x^{(k)} = x$ är en lösning till den ursprungliga ekvationen.

I Jacobis metod används inte de nya approximationerna för de obekanta innan ett iterationssteg blivit slutfört. I *Gauss'–Seidels* metod² används de nya approximationerna genast:

$$x_i^{(k+1)} = \frac{-\sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)} + b_i}{a_{ii}}, \quad i = 1, 2, \dots, n.$$

För varje obekant behöver man alltså endast lagra *en* approximation vid varje iteration, vilket sparar utrymme. Gauss'–Seidels metod är dubbelt snabbare än Jacobis, men detta gäller inte alltid.

Vi skall nu studera konvergensen. Man kan lätt visa, att Jacobis och Gauss'–Seidels algoritmer kan framställas i den kompakta formen

$$x^{(k+1)} = Bx^{(k)} + c, \quad k = 0, 1, 2, \dots,$$

som gäller oförändrad från iteration till iteration (stationär iterativ metod).

²Gauss' metod beskrevs av Gerling i arbetet: *Die Ausgleichungs–Rechnungen der praktischen Geometrie oder die Methode der kleinsten Quadrate mit ihrer Anwendungen für geodätische Aufgaben* (1843). Seidel var Jacobis elev, och publicerade metoden 1874: *Über ein Verfahren, die Gleichungen, auf welche die Methode der kleinsten Quadrate führt, sowie lineäre Gleichungen überhaupt, durch successive Annäherung aufzulösen*, Münch. Abh. II, 81–108 (1874)

Om A uppdelas på följande sätt:

$$A = D(L + I + U),$$

där $D = \text{diag}(a_{ii})$, L är en undre triangulär matris, och U en övre triangulär matris, så inser vi, att Jacobis metod kan uttryckas

$$x^{(k+1)} = -(L + U)x^{(k)} + D^{-1}b,$$

och Gauss'-Seidels metod

$$x^{(k+1)} = -Lx^{(k+1)} - Ux^{(k)} + D^{-1}b.$$

Båda dessa ekvationer har det önskade utseendet om vi väljer $B_J = -(L + U)$ och $B_{GS} = -(I + L)^{-1}U$. I MATLAB sätter man $x = (I+L) \setminus (-U*x+D^{-1}*b)$ för Gauss-Seidel, och löser ekvationen iterativt.

En relation mellan felen i de successiva approximationerna fås genom att subtrahera $x = Bx + c$ från $x^{(k+1)} = Bx^{(k)} + c$:

$$x^{(k+1)} - x = B(x^{(k)} - x) = \dots = B^{k+1}(x^{(0)} - x).$$

Om vi antar att B har egenvärdena $\lambda_1, \lambda_2, \dots, \lambda_n$, (alltså $Bu_i = \lambda_i u_i$) och att de motsvarande egenvektorerna $u_i, i = 1, 2, \dots, n$ är linjärt oberoende, så kan det ursprungliga felet $x^{(0)} - x$ framställas med hjälp av dessa egenvektorer som basvektorer

$$x^{(0)} - x = \alpha_1 u_1 + \dots + \alpha_n u_n, \quad x^{(1)} - x = B(x^{(0)} - x) = \alpha_1 \lambda_1 u_1 + \dots + \alpha_n \lambda_n u_n$$

och vi finner slutligen att

$$x^{(k)} - x = \alpha_1 \lambda_1^k u_1 + \alpha_2 \lambda_2^k u_2 + \dots + \alpha_n \lambda_n^k u_n.$$

Vi ser således att iterationsprocessen konvergerar från en godtycklig utgångspunkt *endast och endast om* $|\lambda_i| < 1, i = 1, 2, \dots, n$.

Man kan göra en enkel modifikation av Gauss'–Seidels metod som avsevärt förbättrar konvergensthastigheten.

Som vi lätt inser, kan Gauss'–Seidels ekvation skrivas i formen $x_i^{(k+1)} = x_i^{(k)} + r_i^{(k)}$, där $r_i^{(k)}$ är resttermen för den i :te ekvationen:

$$r_i^{(k)} = \frac{-\sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i}^n a_{ij} x_j^{(k)} + b_i}{a_{ii}}.$$

Genom att multiplicera resttermen med en **relaxationsparameter** ω får man en ny metod, den sk **successiva överrelaxationsmetoden** (SOR):

$$x_i^{(k+1)} = x_i^{(k)} + \omega r_i^{(k)}.$$

Man strävar efter att välja ett sådant värde av ω , att konvergensthastigheten blir så stor som möjligt. Iterationsmatrisen för överrelaxationsmetoden har formen

$$B_\omega = (I + \omega L)^{-1}[(1 - \omega)I - \omega U].$$

Metoden konvergerar endast om $0 < \omega < 2$. Metoder som har $0 < \omega < 1$ kallas ibland *underrelaxerade*, och metoder med $1 < \omega < 2$ *överrelaxerade* (om $\omega = 1$ får vi Gauss'-Seidels ekvation).

Emedan determinanten för en triangulär matris är lika med produkten av dess diagonala element får man

$$\det(B_\omega) = \det(I + \omega L)^{-1} \det((1 - \omega)I - \omega U) = (1 - \omega)^n.$$

Emedan å andra sidan $\det(B_\omega) = \lambda_1 \lambda_2 \dots \lambda_n$, så gäller $\max_i |\lambda_i| \geq |1 - \omega|$, varav i sin tur följer $|1 - \omega| < 1$. Konvergenzvillkoret gäller alltså. För vissa speciella matriser A kan man t.o.m. beräkna det optimala värdet av ω .

För att lösa stora glesa linjära ekvationssystem, då koefficientmatrisen A är symmetrisk och positivt definit (dvs $x^T Ax > 0$ för alla $x \neq 0$) kan man använda den **konjugerade gradientmetoden**, som introducerades av Magnus Hestenes och Eduard Stiefel år 1952. Fördelen med metoden är, att matrisen endast multipliceras med en vektor, vilket lätt kan programmeras för glesa matriser.

Att lösa systemet $Ax = b$ är ekvivalent med minimering av funktionen $f(x) = \frac{1}{2}x^T Ax - x^T b$, vilket inses på följande sätt. Om man bildar gradienten av $f(x)$ fås $\nabla f(x) = \frac{1}{2}(A + A^T)x - b = Ax - b$ (då $A = A^T$). Vi ser då att villkoret för minimum $\nabla f(x) = 0$ är ekvivalent med $Ax = b$.

Enklaste sättet att finna ett minimum av $f(x)$ är med **steepest descent**-metoden. Det betyder att om x_0 är en approximation till minimet, så kan man söka efter det i den riktning där $f(x)$ avtar snabbast, som är den negativa gradienten av f i punkten x_0 : $g_0 = -\nabla f(x_0) = b - Ax_0$.

Därpå löser man ett endimensionellt minimeringsproblem genom att söka det värde α för vilket funktionen $f(x_0 + \alpha g_0)$ har ett minimum, som leder till en ny punkt $x_1 = x_0 + \alpha g_0$. Proceduren kan upprepas iterativt, men konvergerar tyvärr mycket långsamt mot minimet.

En betydligt snabbare metod får man genom att konstruera *konjugerade riktningar*. En mängd av vektorer p_1, \dots, p_n , som inte är nollvektorer, sägs vara *konjugerade* med avseende på A , om $p_i^T A p_j = 0$ då $i \neq j$. Vektorerna p_i kallas för konjugerade riktningar. Med hjälp av de konjugerade riktningarna kan man lätt lösa ekvationssystemet.

Om $x^* = \alpha_1 p_1 + \dots + \alpha_n p_n$ så följer av villkoret ovan att $p_k^T A x^* = \alpha_k p_k^T A p_k$, dvs

$$\alpha_k = \frac{p_k^T A x^*}{p_k^T A p_k} = \frac{p_k^T b}{p_k^T A p_k}, k = 1, 2, \dots, n$$

Om vi sätter $x_0 = 0$ och $x_k = \alpha_1 p_1 + \dots + \alpha_k p_k$, ($k = 1, \dots, n$) så kan resttermen skrivas $r_k = b - A x_k = -\nabla f(x_k)$. Då är $r_0 = b$ och vi får $x_k = x_{k-1} + \alpha_k p_k$ samt $r_k = r_{k-1} - \alpha_k A p_k$.

I konjugerade gradientmetoden väljer man steepest descent-riktningen som p_1 , sätter resttermen vid det k :te steget till $r_k = b - A x_k$. Därpå fordrar man att alla riktningar p_k är konjugerade, vilket leder till den iterativa proceduren

$$p_{k+1} = r_k - \frac{p_k^T A r_k}{p_k^T A p_k} p_k = r_k + \beta_k p_k$$

Man kan nu visa, att $\alpha_k = \frac{r_k^T r_k}{p_k^T A p_k}$ och $\beta_k = \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}}$, som gör metoden effektivare.

I MATLAB kan man implementera metoden i ett ganska kort program:

```
function [x,nrit]=konjgrad(A,b,tol)
    n=length(b); x=zeros(n,1); r=b; rn=r'*r;
    p=zeros(n,1); nrit=0; relfel=1;
    beta=0;
    while relfel > tol
        p=r+beta*p; Ap=A*p;
        alfa=rn/(p'*Ap);
        dx=alfa*p; x=x+dx; nrit=nrit+1;
        relfel=norm(dx,inf)/norm(x,inf);
        r=r-alfa*Ap; rn1=rn; rn=r'*r;
        beta=rn/rn1;
    end
```

Observera sättet att uttrycka det relativa felet med maximinormen. Ett lämpligt värde för toleransen `tol` är t.ex. $1e-10$.

5.6. Gauss' minsta kvadrat-metod

Den första praktiska tillämpningen av minsta kvadrat-metoden som blivit känd är Gauss' beräkning av småplaneten Ceres' bana. Denna småplanet, som var den första som upptäcktes, sågs av Giuseppe Piazzi första gången nyårsnatten 1801 i Palermo. Han följde den under 21 nätter, men tappade sedan bort den, när den kom alltför nära solen. Ingen säker efemerid hade hunnit beräknas, men med hjälp av minsta kvadrat-metoden kunde Gauss beräkna planetens banelement utgående från endast tre observationer, så att den kunde återfinnas i december av astronomen von Zach. Metoden beskrevs i detalj i Gauss' bok: "*Theoria Motus Corporum Coelestium in sectionibus conicis solem ambientium*" (en teori för rörelsen hos de himlakroppar, som rör sig i kägelsnitt runt solen), som utkom år 1809. Han säger där själv, att han använt metoden så tidigt som 1795. Samma metod hade redan tidigare publicerats av Legendre år 1806.

Gauss' lösningsmetod baserar sig på användningen av normalekvationer. Antag, att vi önskar anpassa en lineär modell till givna (fysikaliska) mätvärden. Vanligen är antalet mätningar större än antalet obekanta. Det gäller då att lösa ett s.k. **överdeterminerat** system, som i matrisform kan uttryckas på följande sätt:

Problem: A är en given $m \times n$ matris, där m (= antalet observationer) $\geq n$ (= antalet obekanta), och b är en kolonnvektor med m element. Bestäm vektorn x (som har n element) så, att Ax är den "bästa" approximationen till b .

Emedan ifrågavarande ekvationssystem är överdeterminerat, kan det inte lösas exakt, och man kan därför inte entydigt definiera en "bästa" lösning. En metod, som är statistiskt motiverad, och dessutom leder till relativt enkla räkningar, är den **minsta kvadratmetoden**.

Enligt minsta kvadratmetoden löses det överdeterminerade systemet $Ax = b$ genom att man söker en vektor x , som minimerar den euklidiska normen (jfr avsnitt 4.4) av restvektorn r , dvs man minimerar

$$\|r\|_2 = \sqrt{r^T r}, \quad r = b - Ax.$$

Om b 's element är fysikaliska mätvärden med slumpmässiga fel, så bör ekvationerna skalas (dvs multipliceras med vikter), så att varianserna av b_i blir lika stora.

Man kan tolka minsta kvadrat-metoden geometriskt så, att man försöker minimera summan av kvadraterna på de vertikala avstånden mellan mätpunkterna och de beräknade punkterna. Detta innebär, att vi antar att de oberoende variablerna (x_i) är felfria. Om också de har fel, är det lämpligare att försöka minimera summan av kvadraterna på de euklidiska avstånden mellan mätpunkterna och de beräknade punkterna. I detta fall kallas metoden **ortogonaldistansregression**. Metoder för att lösa minsta kvadratproblemet då båda variablerna har fel, har på senare tid ofta behandlats i litteraturen. En översikt av forskningen på detta område har skrivits av Macdonald och Thompson³.

³J.R. Macdonald och W.J. Thompson, Am. J. Phys. **60**, 66 (1992)

Minsta kvadratmetodens lösningskriterium: A må vara en reell $m \times n$ matris, och b en kolonnvektor med m element ($m > n$). Om x satisfierar ekvationen

$$A^T(b - Ax) = 0,$$

så gäller för alla vektorer y att

$$\|b - Ax\|_2 \leq \|b - Ay\|_2.$$

Bevis: Sätt $r_x = b - Ax$ och $r_y = b - Ay$. Då gäller att

$$r_y = (b - Ax) + (Ax - Ay) = r_x + A(x - y).$$

Genom kvadrering fås

$$r_y^T r_y = r_x^T r_x + r_x^T A(x - y) + (x - y)^T A^T r_x + (x - y)^T A^T A(x - y).$$

Emedan $A^T r_x = 0$ enligt antagandet, så gäller

$$\|r_y\|_2^2 = \|r_x\|_2^2 + \|A(x - y)\|_2^2 \geq \|r_x\|_2^2, \quad \text{vsb.}$$

Av ekvationen $A^T(b - Ax) = 0$ följer även för alla vektorer z att $(Az)^T(b - Ax) = 0$. Detta innebär, att restvektorn $r_x = b - Ax$ är ortogonal mot varje vektor i rummet $\mathcal{R}(A)$ som spänns av

kolonnerna i A . Av ekvationerna $A^T r_x = 0$ följer direkt, att

$$(A^T A)x = A^T b$$

(**normalekvationerna**). $C = A^T A$ är här en symmetrisk $n \times n$ matris med elementen

$$c_{ij} = a_i^T a_j, \quad A = (a_1, a_2, \dots, a_n).$$

Normalekvationssystemet består alltså av n linjära ekvationer med n obekanta x_i , $i = 1, 2, \dots, n$:

$$a_1^T a_1 x_1 + \dots + a_1^T a_n x_n = a_1^T b$$

...

$$a_n^T a_1 x_1 + \dots + a_n^T a_n x_n = a_n^T b.$$

Eftersom $A^T A$ är symmetrisk, så kan $A^T A$ och $A^T b$ beräknas med $\frac{n}{2}(n+1)m + nm = \frac{1}{2}mn(n+3)$ räkneoperationer. Om $A^T A$ inte är singulär, kan normalekvationerna lösas t. ex. med Gauss' elimineringsmetod, som approximativt kräver $n^3/6$ räkneoperationer. Största delen av räknearbetet åtgår alltså till att bilda normalekvationerna.

Matrisen $A^T A$ är icke-singulär om och endast om A 's kolonner är linjärt oberoende vektorer. Detta bevisas på följande sätt. Om kolonnerna är linjärt oberoende, så följer därav $x \neq 0 \Rightarrow Ax \neq 0$. Sålunda gäller $x^T (A^T A)x = (Ax)^T Ax = \|Ax\|_2^2 > 0$, dvs $A^T A$ är *positivt definit* (se avsn. 5.1). Då är $\det(A^T A) > 0$ (följer av *Sylvesters kriterium*, dvs alla underdeterminanter som kan bildas av en positivt definit matris är positiva) och $A^T A$ är alltså icke-singulär.

Av detta teorem följer, att om A 's kolonner är linjärt oberoende, så är lösningen *entydig*, och $x = A^I b$ där $A^I = (A^T A)^{-1} A^T$ betecknar matrisen A 's **pseudo-invers**, som uppfyller villkoret $A^I A = I$.

Om A 's kolonner är linjärt oberoende, så följer av $(Az)^T r = 0 \forall z$ att Ax är en *ortogonal projektion* av b på den rymd som spänns av A 's kolonner, och vi kan skriva

$$r = b - Ax = b - AA^I b = (I - P_A)b,$$

där $P_A = AA^I = A(A^T A)^{-1} A^T$ kallas för den **ortogonala projektionsoperatorn**. Man kan lätt visa, att P_A är symmetrisk, $P_A^2 = P_A$, $(I - P_A)^2 = I - P_A$, samt att $(I - P_A)P_A = 0$.