Dispense del corso di

LABORATORIO DI PROGRAMMAZIONE E CALCOLO

Marco Marfurt

Parte II: Risoluzione numerica di sistemi di equazioni lineari

0.1 Matrici, vettori e sistemi lineari

Nel seguito considereremo sempre matrici ad elementi reali, che indicheremo sistematicamente con lettere latine maiuscole in carattere grassetto; per esempio, se $\mathbf{A}$ è una matrice ad m righe ed n colonne, scriveremo

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ ... & ... & ... & ... \\ a_{m1} & a_{m2} & ... & a_{mn} \end{pmatrix}$$

intendendo così che $\mathbf{A}$ è il nome della matrice (appunto in carattere grassetto), mentre a_{ij} indica l'elemento di $\mathbf{A}$ che si trova sulla riga $i-esima$ e sulla colonna $j-esima$ e che, essendo un numero reale, viene indicato in caratteri normali. Questa convenzione, peraltro abbastanza diffusa, ci sarà comoda per distinguere subito nelle formule quali oggetti sono matrici e quali sono numeri reali. In particolare osserviamo che un vettore ad n componenti può essere pensato come una matrice in due modi un pò diversi fra loro: cioè si può pensarlo come una matrice ad una riga ed n colonne, oppure come una matrice ad n righe ed una colonna; un'altra convenzione che adotteremo è quella di indicare con lettere latine minuscole in carattere grassetto un vettore ad n componenti *pensandolo sempre come una matrice ad n righe ed una colonna*; per esempio, se $\mathbf{u}$ è il nome del vettore di componenti $u_1, u_2, \ldots, u_n$, ciò significa che noi pensiamo ad $\mathbf{u}$ come una matrice al modo seguente:

$$\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \\ ... \\ u_n \end{pmatrix}$$

Sia ora $\mathbf{A}$ la matrice precedentemente definita e sia

$$\mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & ... & b_{1n} \\ b_{21} & b_{22} & ... & b_{2n} \\ ... & ... & ... & ... \\ b_{m1} & b_{m2} & ... & b_{mn} \end{pmatrix}$$

un'altra matrice ad m righe ed n colonne (d'ora in poi scriveremo brevemente che $\mathbf{A}$ e $\mathbf{B}$ sono matrici mxn); chiameremo *matrice somma* di $\mathbf{A}$ e $\mathbf{B}$, la matrice $\mathbf{C}$ (anch'essa mxn) ottenuta sommando $\mathbf{A}$ e $\mathbf{B}$ *elemento per elemento*, cioè, se è

$$\mathbf{C} = \mathbf{A} + \mathbf{B} = \begin{pmatrix} c_{11} & c_{12} & ... & c_{1n} \\ c_{21} & c_{22} & ... & c_{2n} \\ ... & ... & ... & ... \\ c_{m1} & c_{m2} & ... & c_{mn} \end{pmatrix}$$

allora dovrà essere

$$c_{ij} = a_{ij} + b_{ij} \quad per \quad i = 1, \ldots, m; j = 1, \ldots, n$$

Osserviamo che questa operazione di *somma fra matrici* che abbiamo ora definita è eseguibile solo se le due matrici da sommare sono delle stesse dimensioni, cioè sono entrambe mxn. Anche due vettori $\mathbf{u}$ e $\mathbf{v}$ che hanno lo stesso numero n di componenti, potranno essere sommati allo stesso modo *componente per componente*, dal momento che li consideriamo come due matrici ad n righe ed una colonna.

Un'altra operazione che riguarda le matrici e che utilizzeremo spesso, è il *prodotto di uno scalare per una matrice*, che si definisce al modo seguente: sia λ uno scalare (cioè, nel nostro caso, un numero reale) e sia $\mathbf{A}$ una matrice mxn i cui elementi indicheremo con a_{ij}, allora la matrice $\mathbf{C} = \lambda\mathbf{A}$ (*prodotto di* λ *per* $\mathbf{A}$) è ancora una matrice mxn i cui elementi si ottengono moltiplicando i corrispondenti elementi di $\mathbf{A}$ per λ, cioè, se è

$$\mathbf{C} = \lambda\mathbf{A} = \begin{pmatrix} c_{11} & c_{12} & \ldots & c_{1n} \\ c_{21} & c_{22} & \ldots & c_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ c_{m1} & c_{m2} & \ldots & c_{mn} \end{pmatrix}$$

allora dovrà essere

$$c_{ij} = \lambda a_{ij} \quad per \quad i = 1, \ldots, m \;\; e \;\; j = 1, \ldots, n$$

Siano ora $\mathbf{A}$ e $\mathbf{B}$ due matrici rispettivamente mxn e nxk, e quindi sarà

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & \ldots & b_{1k} \\ b_{21} & b_{22} & \ldots & b_{2k} \\ \ldots & \ldots & \ldots & \ldots \\ b_{n1} & b_{n2} & \ldots & b_{nk} \end{pmatrix}$$

Chiameremo *prodotto righe per colonne* di $\mathbf{A}$ per $\mathbf{B}$, e indicheremo con $\mathbf{AB}$, la matrice $\mathbf{C}$ mxk definita da

$$\mathbf{C} = \mathbf{AB} = \begin{pmatrix} c_{11} & c_{12} & \ldots & c_{1k} \\ c_{21} & c_{22} & \ldots & c_{2k} \\ \ldots & \ldots & \ldots & \ldots \\ c_{m1} & c_{m2} & \ldots & c_{mk} \end{pmatrix}$$

dove è

$$c_{ij} = \sum_{s=1}^{n} a_{is} b_{sj} \quad per \quad i = 1, \ldots, m; j = 1, \ldots, k$$

Come risulta chiaro da questa definizione, l'operazione di *prodotto righe per colonne* (che d'ora in poi chiameremo semplicemente *prodotto*) è eseguibile solo se il numero delle colonne della prima matrice coincide con il numero delle righe della seconda matrice (ovvero, equivalentemente, se le righe della prima matrice e le colonne della seconda matrice hanno lo stesso numero di elementi); quindi, se il prodotto **AB** è eseguibile (perchè **A** è mxn e **B** è nxk), non sarà invece eseguibile il prodotto **BA** (a meno che non sia $m = k$). Osserviamo comunque che, anche nel caso in cui **A** sia una matrice mxn e **B** una matrice nxm (per cui sono eseguibili entrambi i prodotti **AB** e **BA**), mentre il prodotto **AB** sarà una matrice mxm, il prodotto **BA** sarà una matrice nxn e quindi non potrà essere **AB** = **BA**, a meno che non sia $m = n$. Vediamo infine che cosa succede nel caso in cui sia **A** che **B** sono matrici nxn (o, come spesso si dice, sono matrici *quadrate*): in questo caso sia il prodotto **AB** che il prodotto **BA** sono eseguibili e danno luogo in entrambi i casi a una matrice nxn, tuttavia non è detto che sia **AB** = **BA**; ecco un semplice controesempio:

$$\mathbf{A} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

Il prodotto fra matrici è quindi un esempio di operazione *non commutativa*; tuttavia è bene anche tenere presente che talvolta può accadere che il prodotto di due particolari matrici *commuti*, nel senso che possono esistere due particolari matrici **A** e **B** tali che

$$\mathbf{AB} = \mathbf{BA}.$$

Un semplice esempio di matrici che commutano è il seguente: sia **D** una matrice nxn della forma:

$$\mathbf{D} = \begin{pmatrix} d_{11} & 0 & \ldots & 0 \\ 0 & d_{22} & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & d_{nn} \end{pmatrix}$$

cioè tale che

$$d_{ij} = 0 \quad per \quad i \neq j$$

(diremo in questo caso che $\mathbf{D}$ è una matrice *diagonale*), allora

$$\mathbf{ED} = \mathbf{DE}$$

per qualsiasi coppia di matrici diagonali $\mathbf{E}$ e $\mathbf{D}$ nxn, cioè le matrici diagonali nxn *commutano* nel prodotto fra di loro.

In particolare, se $\mathbf{A}$ è una matrice nxn e $\mathbf{u}$ è un vettore ad n componenti (e quindi, per la nostra convenzione, una matrice $nx1$), sarà eseguibile il prodotto $\mathbf{Au}$ ed il risultato del prodotto sarà una matrice $nx1$, cioè di nuovo un vettore; quindi, se

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ ... & ... & ... & ... \\ a_{n1} & a_{n2} & ... & a_{nn} \end{pmatrix}$$

e

$$\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \\ ... \\ u_n \end{pmatrix}$$

il vettore $\mathbf{v} = \mathbf{Au}$ avrà componenti

$$v_i = a_{i1}u_1 + a_{i2}u_2 + \ldots + a_{in}u_n \quad per \quad i = 1, \ldots, n$$

Il prodotto fra matrici è invece una operazione *associativa*, cioè, se $\mathbf{A}$ è una matrice mxn, $\mathbf{B}$ una matrice nxk e $\mathbf{C}$ una matrice kxr, allora si avrà

$$(\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC})$$

e inoltre gode della *proprietà distributiva rispetto alla somma*, cioè, se $\mathbf{A}$, $\mathbf{B}$ e $\mathbf{C}$ sono rispettivamente mxn, nxk e nxk (ovvero nxk, mxn e mxn), allora si avrà

$$\mathbf{A}(\mathbf{B} + \mathbf{C}) = \mathbf{AB} + \mathbf{AC} \quad (ovvero \quad (\mathbf{B} + \mathbf{C})\mathbf{A} = \mathbf{BA} + \mathbf{CA} \ )$$

(Si osservi che occorrono *due proprietà distributive* perchè il prodotto non è commutativo).

Sia ora $\mathbf{A}$ la seguente matrice mxn

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ ... & ... & ... & ... \\ a_{m1} & a_{m2} & ... & a_{mn} \end{pmatrix}$$

chiameremo *matrice trasposta* di **A** e indicheremo con **A**' la matrice nxm le cui righe coincidono con le colonne di **A**, cioè sarà quindi:

$$\mathbf{A}' = \begin{pmatrix} a_{11} & a_{21} & ... & a_{m1} \\ a_{12} & a_{22} & ... & a_{m2} \\ ... & ... & ... & ... \\ a_{1n} & a_{2n} & ... & a_{mn} \end{pmatrix}$$

Come abbiamo detto, se **A** è una matrice mxn,la sua trasposta **A**' è una matrice nxm, che sarà quindi, in generale, diversa da **A**; tuttavia, nel caso in cui $m = n$, cioè se **A** è una matrice nxn, la sua matrice trasposta **A**' sarà ancora una matrice nxn; diremo che una matrice **A** nxn è *simmetrica* se essa coincide con la sua trasposta, cioè se

$$\mathbf{A} = \mathbf{A}';$$

è immediato verificare che se **A** è una matrice nxn simmetrica allora

$$a_{ij} = a_{ji} \quad per \quad i, j = 1, ..., n$$

Nel caso di un vettore **u** ad n componenti avremo in particolare che

$$\mathbf{u}' = \begin{pmatrix} u_1 & u_2 & ... & u_n \end{pmatrix}$$

e quindi, utilizzando l'operatore di trasposizione, possiamo rappresentare un vettore come una matrice ad una riga ed n colonne.

In particolare osserviamo il seguente fatto: se **u** e **v** sono due vettori ad n componenti, il prodotto righe per colonne di **u** per **v** non sarà eseguibile in quanto essi sono entrambi matrici $nx1$; tuttavia il prodotto **u**'**v** sarà eseguibile perchè **u**' è una matrice $1xn$ e **v** è una matrice $nx1$; il risultato di questo prodotto righe per colonne sarà una matrice $1x1$, cioè quindi un numero reale. Se applichiamo la definizione di prodotto (righe per colonne) al prodotto **u**'**v**, otteniamo

$$\mathbf{u}'\mathbf{v} = \sum_{i=1}^{n} u_i v_j$$

che è la formula che definisce l'ordinario *prodotto scalare* dei vettori **u** e **v**.

Consideriamo ora un sistema di n equazioni lineari in n incognite:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n \end{aligned} \tag{1}$$

Esso è *caratterizzato* dalla sua *matrice dei coefficienti*

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ ... & ... & ... & ... \\ a_{n1} & a_{n2} & ... & a_{nn} \end{pmatrix}$$

e dai suoi *termini noti* $b_1, b_2, \ldots, b_n$, che possono essere pensati complessivamente come un vettore le cui componenti sono $b_1, b_2, \ldots, b_n$; chiameremo brevemente $\mathbf{b}$ questo vettore; in base alla nostra convenzione sui vettori, sarà allora:

$$\mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ ... \\ b_n \end{pmatrix}$$

Se ora definiamo anche il vettore $\mathbf{x}$ che ha come componenti le incognite $x_1, x_2, \ldots, x_n$, cioè

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ ... \\ x_n \end{pmatrix}$$

è facile verificare che il sistema (1) può scriversi nella *forma compatta*

$$\mathbf{Ax} = \mathbf{b} \tag{2}$$

0.2 Metodo di eliminazione di Gauss

Cominciamo con il considerare un sistema di n equazioni lineari in n incognite che abbia la seguente particolare struttura:

$$\begin{aligned} u_{11}x_1 + u_{12}x_2 + u_{13}x_3 + \cdots\cdots\cdots + u_{1n}x_n &= b_1 \\ u_{22}x_2 + u_{23}x_3 + \cdots\cdots\cdots + u_{2n}x_n &= b_2 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & \\ u_{n-1,n-1}x_{n-1} + u_{n-1,n}x_n &= b_{n-1} \\ u_{nn}x_n &= b_n \end{aligned} \tag{3}$$

ovvero, in forma compatta

$$\mathbf{Ux} = \mathbf{b}$$

dove $\mathbf{U}$ è la matrice

$$\mathbf{U} = \begin{pmatrix} u_{11} & u_{12} & u_{13} & ... & u_{1,n-1} & u_{1n} \\ 0 & u_{22} & u_{23} & ... & u_{2,n-1} & u_{2n} \\ ... & ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & u_{n-1,n-1} & u_{n-1,n} \\ 0 & 0 & 0 & ... & 0 & u_{nn} \end{pmatrix}$$

(diremo in questo caso che $\mathbf{U}$ è una matrice *triangolare superiore*); in questa situazione particolare, la soluzione del sistema (3) si ottiene molto facilmente al modo seguente: prima di tutto, l'ultima equazione del sistema

$$u_{nn}x_n = b_n$$

è un'equazione nella sola incognita x_n e la sua soluzione è data immediatamente da

$$x_n = \frac{b_n}{u_{nn}};$$

se passiamo ora alla penultima equazione del sistema (3), cioè

$$u_{n-1,n-1}x_{n-1} + u_{n-1,n}x_n = b_{n-1}$$

vediamo che anche questa è un'equazione in una sola incognita, in quanto la x_n è già stata calcolata in precedenza e rimane quindi la sola incognita x_{n-1} il cui valore si ottiene facilmente come

$$x_{n-1} = \frac{b_{n-1} - u_{n-1,n}x_n}{u_{n-1,n-1}};$$

è chiaro a questo punto, che possiamo procedere *a ritroso* in questo modo, risalendo dall'ultima equazione fino alla prima e risolvendo ogni volta una singola equazione in una singola incognita; non è difficile vedere che la formula relativa alla generica incognita sarà la seguente:

$$x_i = \frac{b_i - \sum_{j=i+1}^{n} u_{i,j}x_j}{u_{i,i}} \quad per \;\; i = n-1, n-2, \ldots, 1. \tag{4}$$

Si osservi che, se tutti gli *elementi diagonali* $u_{i,i}$ per $i = 1, \ldots, n$ della matrice triangolare $\mathbf{U}$ sono diversi da zero, allora il *rango* della matrice $\mathbf{U}$ è n (ovvero il *determinante* di $\mathbf{U}$ è diverso da zero) e quindi il sistema triangolare $\mathbf{Ux} = \mathbf{b}$ ammette una e una sola soluzione; d'altra parte la condizione che gli elementi diagonali di $\mathbf{U}$ siano diversi da zero garantisce anche che l'algoritmo da noi descritto per il calcolo delle soluzioni sia effettivamente eseguibile,

infatti l'unica condizione che deve essere soddisfatta nella (4) perchè x_i sia *effettivamente calcolabile* è che sia $u_{i,i} \neq 0$.

Passiamo ora a considerare un generico sistema di n equazioni lineari in n incognite

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n \end{aligned} \tag{5}$$

ovvero, in forma compatta

$$\mathbf{A}\mathbf{x} = \mathbf{b};$$

d'ora in poi supporremo sempre che $\mathbf{A}$ sia una matrice non singolare (cioè di *rango* n), per cui il sistema ammette una e una sola soluzione.

Il *metodo di eliminazione di Gauss* per risolvere il sistema (5) consiste nel costruire una sequenza di sistemi lineari

$$\mathbf{A}^{(k)}\mathbf{x} = \mathbf{b}^{(k)} \quad per \;\; k = 1, \ldots, n$$

in modo tale che ciascun sistema della sequenza sia *equivalente* al sistema (5) ed infine che l'ultimo sistema della sequenza

$$\mathbf{A}^{(n)}\mathbf{x} = \mathbf{b}^{(n)}$$

sia *triangolare superiore*; in questo modo saremmo ritornati al caso precedente e potremmo risolvere l'ultimo sistema con l'algoritmo appena descritto.

Come sappiamo, due sistemi di equazioni lineari si dicono *equivalenti* se essi hanno le stesse soluzioni; un modo molto semplice per passare da un sistema di equazioni lineari ad un altro sistema ad esso equivalente, è il seguente:

dato un sistema di equazioni lineari, se si somma ad una qualsiasi equazione del sistema una qualsiasi altra equazione moltiplicata (o divisa) per una costante diversa da zero, si ottiene un sistema equivalente al sistema di partenza.

è appunto utilizzando questa regola che noi costruiremo la sequenza di sistemi lineari sopra definita.

Prima di tutto porremo

$$\mathbf{A}^{(1)} = \mathbf{A} \quad e \quad \mathbf{b}^{(1)} = \mathbf{b}$$

in modo tale che il sistema di partenza coincida con il sistema assegnato; dopo di che osserviamo che, se $a_{11}^{(1)} \neq 0$, definendo

$$m_{i1} = \frac{a_{i1}^{(1)}}{a_{11}^{(1)}} \quad per \quad i = 2, 3, \ldots, n$$

e sottraendo alla i-esima equazione del sistema la prima equazione moltiplicata per m_{i1}, si otterrà un nuovo sistema, equivalente a quello di partenza, nel quale il coefficiente della prima incognita nella i-esima equazione sarà nullo; ripetendo questo procedimento per l'indice i che varia da 2 ad n, otterremo quindi alla fine un sistema, sempre equivalente a quello di partenza, nel quale i coefficienti della prima incognita sono nulli in tutte le equazioni che vanno dalla seconda fino all'ultima. Se chiamiamo

$$\mathbf{A}^{(2)}\mathbf{x} = \mathbf{b}^{(2)}$$

il sistema così ottenuto, possiamo dire che la matrice $\mathbf{A}^{(2)}$ ha tutti gli elementi della prima colonna nulli, eccetto il primo. Ecco quindi come saranno le formule che forniscono gli elementi di $\mathbf{A}^{(2)}$ e di $\mathbf{b}^{(2)}$ in funzione degli elementi di $\mathbf{A}^{(1)}$ e di $\mathbf{b}^{(1)}$:

$$a_{ij}^{(2)} = a_{ij}^{(1)} - m_{i1}a_{1j}^{(1)} \quad per \quad i, j = 2, \ldots, n$$

$$b_{i}^{(2)} = b_{i}^{(1)} - m_{i1}b_{1}^{(1)} \quad per \quad i = 2, \ldots, n$$

Abbiamo così compiuto il primo passo; ora non ci resta che procedere allo stesso modo sulla matrice $\mathbf{A}^{(2)}$, questa volta cercando di rendere nulli tutti gli elementi della seconda colonna, eccetto i primi due; non è difficile vedere che questa volta, se $a_{22}^{(2)} \neq 0$, i coefficienti che dovremo usare saranno

$$m_{i2} = \frac{a_{i2}^{(2)}}{a_{22}^{(2)}} \quad per \quad i = 3, 4, \ldots, n$$

e sottraendo alla i-esima equazione la seconda equazione moltiplicata per m_{i2} e ripetendo ciò per i che varia da 3 ad n, si ottiene alla fine il sistema equivalente

$$\mathbf{A}^{(3)}\mathbf{x} = \mathbf{b}^{(3)}$$

dove la matrice $\mathbf{A}^{(3)}$ avrà ora nulli tutti gli elementi della prima colonna eccetto il primo e tutti gli elementi della seconda colonna eccetto i primi due. Ecco quindi come saranno le formule che forniscono gli elementi di $\mathbf{A}^{(3)}$ e di $\mathbf{b}^{(3)}$ in funzione degli elementi di $\mathbf{A}^{(2)}$ e di $\mathbf{b}^{(2)}$:

$$a_{ij}^{(3)} = a_{ij}^{(2)} - m_{i2}a_{2j}^{(2)} \quad per \quad i,j = 3,\ldots,n$$

$$b_{i}^{(3)} = b_{i}^{(2)} - m_{i2}b_{2}^{(2)} \quad per \quad i = 3,\ldots,n$$

Non ci resta ora che scrivere le formule del passaggio generico a partire dal sistema

$$\mathbf{A}^{(k)}\mathbf{x} = \mathbf{b}^{(k)}$$

dove la matrice $\mathbf{A}^{(k)}$ sarà una matrice che ha tutti nulli gli elementi al di sotto della diagonale principale nelle colonne da 1 a $k-1$. Poichè vogliamo annullare questa volta gli elementi al disotto della diagonale principale nella k-esima colonna, i coefficienti che dovremo ora usare, se $a_{kk}^{(k)} \neq 0$, saranno

$$m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \quad per \quad i = k+1, k+2, \ldots, n \tag{6}$$

e sottraendo alla i-esima equazione la k-esima equazione moltiplicata per m_{ik} e ripetendo ciò per i che varia da $k+1$ ad n, si ottiene alla fine il sistema equivalente

$$\mathbf{A}^{(k+1)}\mathbf{x} = \mathbf{b}^{(k+1)}$$

dove $\mathbf{A}^{(k+1)}$ sarà una matrice che ha tutti nulli gli elementi al di sotto della diagonale principale nelle colonne da 1 a k. Ed ecco infine come saranno le formule che forniscono gli elementi di $\mathbf{A}^{(k+1)}$ e di $\mathbf{b}^{(k+1)}$ in funzione degli elementi di $\mathbf{A}^{(k)}$ e di $\mathbf{b}^{(k)}$:

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik}a_{kj}^{(k)} \quad per \quad i,j = k+1,\ldots,n \tag{7}$$

$$b_{i}^{(k+1)} = b_{i}^{(k)} - m_{ik}b_{k}^{(k)} \quad per \quad i = k+1,\ldots,n \tag{8}$$

Le (6),(7) e (8) permettono quindi di definire completamente il metodo di eliminazione di Gauss; l'algoritmo completo richiederà tre cicli ognuno annidato all'interno del precedente e si potrebbe descrivere sinteticamente al modo seguente:

inizio del ciclo $k = 1, ..., n-1$

inizio del ciclo $i = k+1, ..., n$

calcola $m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$

inizio del ciclo $j = k+1, ..., n$

calcola $a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)}$

fine del ciclo su j

calcola $b_i^{(k+1)} = b_i^{(k)} - m_{ik} b_k^{(k)}$

fine del ciclo su i

fine del ciclo su k

L'algoritmo di eliminazione di Gauss che abbiamo ora descritto ha però un inconveniente su cui fino ad ora abbiamo sorvolato: poichè per ogni k dobbiamo eseguire la divisione per l'elemento $a_{kk}^{(k)}$ (detto *elemento pivot*), quest'ultimo dovrà essere diverso da zero, infatti, in caso contrario, l'algoritmo si arresterebbe senza poter più proseguire. Un modo molto semplice di ovviare a questo inconveniente è il seguente: prima di iniziare *il ciclo su i*, si esaminano tutti gli elementi del tipo

$$a_{ik}^{(k)} \quad per \quad i = k, k+1, ..., n$$

(cioè tutti gli elementi della k-esima colonna a partire da quello sulla diagonale principale e scendendo verso il basso) e si determina il più grande di essi in valore assoluto; se $a_{rk}^{(k)}$ è questo elemento di massimo modulo si effettua lo scambio della riga k-esima della matrice con la riga r-esima e inoltre si scambia il termine noto k-esimo con il termine noto r-esimo. Tutto ciò ha come effetto di mantenere equivalente il sistema (infatti abbiamo semplicemente scambiato fra loro due equazioni) e contemporaneamente di avere ora come *elemento pivot* un elemento che è più grande in modulo di tutti quelli della k-esima colonna che stanno sotto di lui; in questo modo l'elemento pivot sarà ora certamente diverso da zero perchè altrimenti il rango della matrice del sistema sarebbe inferiore ad n e quindi il sistema non sarebbe univocamente risolubile (contro l'ipotesi iniziale).

L'algoritmo di eliminazione di Gauss così modificato prende il nome di *metodo di Gauss con il pivot* o semplicemente di *metodo del pivot*; ecco come potremmo descriverlo sinteticamente:

inizio del ciclo $k = 1, ..., n-1$

trova l'indice r *per cui* $|a_{rk}^{(k)}| \geq |a_{ik}^{(k)}|$ *per* $i = k, ..., n$

se $r \neq k$ *scambia l'equazione k-esima con la r-esima*

inizio del ciclo $i = k+1, ..., n$

calcola $m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$

inizio del ciclo $j = k+1, ..., n$

calcola $a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)}$

fine del ciclo su j

calcola $b_i^{(k+1)} = b_i^{(k)} - m_{ik} b_k^{(k)}$

fine del ciclo su i

fine del ciclo su k

Per concludere con il metodo di eliminazione di Gauss, cerchiamo ora di capire quale è il *costo di calcolo* di questo metodo, dove per *costo di calcolo di un metodo* intenderemo il tempo che richiede l'esecuzione completa dell'algoritmo; poichè questo tempo si può considerare proporzionale al numero di operazioni aritmetiche che devono essere eseguite, ciò che dobbiamo fare ora è contare appunto queste operazioni. Prima di tutto cominciamo a contare le operazioni di prodotto e divisione: nella definizione che abbiamo data dell'algoritmo di Gauss abbiamo un *ciclo* interno su j nel quale dobbiamo eseguire gli $(n-k)$ prodotti $m_{ik} a_{kj}^{(k)}$ a questi dobbiamo aggiungere il prodotto $m_{ik} b_k^{(k)}$ e, inizialmente , la divisione

$$\frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$$

e quindi in totale abbiamo già $(n-k+2)$ operazioni di prodotto e divisione; queste operazioni vanno ripetute nel ciclo per i che va da $k+1$ ad n e quindi in totale $(n-k)$ volte. In definitiva, per ogni valore di k dovremo eseguire $(n-k)(n-k+2)$ operazioni di prodotto e divisione e quindi, dovendo ripetere questi cicli per k che assume i valori da 1 ad $n-1$, ne segue che il numero totale di operazioni di prodotto e divisione per completare l'algoritmo di eliminazione di Gauss, sarà

$$\sum_{k=1}^{n-1} (n-k)(n-k+2);$$

con qualche calcolo si può verificare che questo numero è

$$\frac{1}{3}n^3 + \frac{1}{2}n^2 - \frac{5}{6}n$$

e quindi possiamo concludere che, per valori di n non troppo piccoli, il numero totale di prodotti e divisioni richiesto dal metodo di eliminazione di Gauss è di circa $\frac{1}{3}n^3$. In effetti ci sarebbe da tener conto anche delle operazioni di prodotto e divisione che servono per risolvere il sistema ridotto in forma triangolare, tuttavia un analogo semplice conteggio mostra che questo numero è solo dell'ordine di n^2 e quindi non modifica qualitativamente il risultato precedente.

In maniera del tutto analoga si può procedere per contare le operazioni di somma e sottrazione, pervenendo anche qui allo stesso risultato di circa $\frac{1}{3}n^3$ operazioni.

0.3 Metodo di Jacobi

Abbiamo visto che per risolvere un sistema di n equazioni in n incognite si può ricorrere al metodo di eliminazione di Gauss; questo procedimento appartiene alla categoria dei cosiddetti *metodi diretti*, nel senso che esso fornisce (in teoria) la soluzione esatta dopo un numero finito di operazioni di somma e prodotto (che, come abbiamo visto, è circa $\frac{1}{3}n^3$ prodotti e altrettante somme).

In alternativa ai *metodi diretti* (ne esistono ovviamente molti altri oltre al metodo di Gauss), è possibile usare metodi di tipo *iterativo*; i metodi di questo tipo consistono nel costruire opportunamente una successione di vettori ad n componenti in modo tale che convergano ad un *vettore limite* le cui componenti costituiscono proprio la soluzione del sistema. Naturalmente in questo modo verrà calcolata solo una soluzione approssimata, in quanto, in pratica, nel costruire la successione di vettori dovremo accontentarci di costruirne solo un segmento finito, arrestandoci dopo un opportuno numero (finito) di iterazioni (in effetti, come vedremo, uno dei problemi che ci si porrà sarà proprio quello di decidere quando arrestarci).

Detta così, la faccenda sembra complicata, ma in pratica le cose sono abbastanza semplici; vediamo per esempio come si può costruire un classico metodo iterativo, noto come *metodo di Jacobi*. Sia dato quindi il sistema di n equazioni lineari in n incognite

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\
&\cdots\cdots\cdots\cdots \\
&\cdots\cdots\cdots\cdots \\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n
\end{aligned}
\tag{9}
$$

che supporremo ammetta una e una sola soluzione che indicheremo con $\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n$; supporremo inoltre che il sistema (9) sia a *diagonale strettamente dominante* , cioè che valgano le seguenti disuguaglianze

$$
\sum_{\substack{j=1 \\ (j\neq i)}}^{n} |a_{ij}| < |a_{ii}| \quad per \quad (i = 1, 2, \ldots, n) \tag{10}
$$

(Questa è ovviamente una ipotesi restrittiva, nel senso che non sarà verificata in generale da ogni sistema di equazioni lineari con matrice dei coefficienti non singolare; tuttavia senza di essa il metodo di Jacobi potrebbe non funzionare, come vedremo meglio in seguito.)

Immaginiamo ora di risolvere la prima equazione rispetto alla prima incognita, la seconda equazione rispetto alla seconda incognita, ecc. fino a risolvere l'ultima equazione rispetto all'ultima incognita; avremo

$$
\begin{aligned}
x_1 &= -\frac{a_{12}}{a_{11}}x_2 - \frac{a_{13}}{a_{11}}x_3 - \cdots\cdots - \frac{a_{1n}}{a_{11}}x_n + \frac{b_1}{a_{11}} \\
x_2 &= -\frac{a_{21}}{a_{22}}x_1 - \frac{a_{23}}{a_{22}}x_3 - \cdots\cdots - \frac{a_{2n}}{a_{22}}x_n + \frac{b_2}{a_{22}} \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
x_n &= -\frac{a_{n1}}{a_{nn}}x_1 - \frac{a_{n2}}{a_{nn}}x_2 - \cdots - \frac{a_{nn-1}}{a_{nn}}x_{n-1} + \frac{b_n}{a_{nn}}
\end{aligned}
\tag{11}
$$

Poichè il sistema (11) è equivalente al sistema (9), se sostituiamo i valori $\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n$ nei secondi membri del sistema (11), nei primi membri si dovranno riottenere rispettivamente $\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n$, cioè avremo che

$$
\begin{aligned}
\overline{x}_1 &= -\frac{a_{12}}{a_{11}}\overline{x}_2 - \frac{a_{13}}{a_{11}}\overline{x}_3 - \cdots\cdots\cdots - \frac{a_{1n}}{a_{11}}\overline{x}_n + \frac{b_1}{a_{11}} \\
\overline{x}_2 &= -\frac{a_{21}}{a_{22}}\overline{x}_1 - \frac{a_{23}}{a_{22}}\overline{x}_3 - \cdots\cdots\cdots - \frac{a_{2n}}{a_{22}}\overline{x}_n + \frac{b_2}{a_{22}} \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
\overline{x}_n &= -\frac{a_{n1}}{a_{nn}}\overline{x}_1 - \frac{a_{n2}}{a_{nn}}\overline{x}_2 - \cdots - \frac{a_{nn-1}}{a_{nn}}\overline{x}_{n-1} + \frac{b_n}{a_{nn}}
\end{aligned}
\tag{12}
$$

Se ora invece sostituiamo nei secondi membri di (11) n valori qualsiasi $x_1^{'}, x_2^{'}, \ldots, x_n^{'}$ (diversi da $\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n$), nei primi membri si determineranno n numeri $x_1^{''}, x_2^{''}, \ldots, x_n^{''}$ (diversi sia dai $\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n$ che dai $x_1^{'}, x_2^{'}, \ldots, x_n^{'}$); cioè si avrà

$$
\begin{aligned}
x_1^{''} &= -\frac{a_{12}}{a_{11}}x_2^{'} - \frac{a_{13}}{a_{11}}x_3^{'} - \cdots\cdots\cdots - \frac{a_{1n}}{a_{11}}x_n^{'} + \frac{b_1}{a_{11}} \\
x_2^{''} &= -\frac{a_{21}}{a_{22}}x_1^{'} - \frac{a_{23}}{a_{22}}x_3^{'} - \cdots\cdots\cdots - \frac{a_{2n}}{a_{22}}x_n^{'} + \frac{b_2}{a_{22}} \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
x_n^{''} &= -\frac{a_{n1}}{a_{nn}}x_1^{'} - \frac{a_{n2}}{a_{nn}}x_2^{'} - \cdots - \frac{a_{nn-1}}{a_{nn}}x_{n-1}^{'} + \frac{b_n}{a_{nn}}
\end{aligned}
\tag{13}
$$

Se ora definiamo le quantità

$$\epsilon^{'} = \max_{i=1,\ldots,n} |x_i^{'} - \overline{x}_i|$$

$$\epsilon^{''} = \max_{i=1,\ldots,n} |x_i^{''} - \overline{x}_i|$$

e sottraiamo le (12) dalle (13), con facili passaggi otteniamo le disuguaglianze

$$
\begin{aligned}
|x_1^{''} - \overline{x}_1| &\leq |\frac{a_{12}}{a_{11}}|\epsilon^{'} + |\frac{a_{13}}{a_{11}}|\epsilon^{'} + \cdots\cdots + |\frac{a_{1n}}{a_{11}}|\epsilon^{'} \\
|x_2^{''} - \overline{x}_2| &\leq |\frac{a_{21}}{a_{22}}|\epsilon^{'} + |\frac{a_{23}}{a_{22}}|\epsilon^{'} + \cdots\cdots + |\frac{a_{2n}}{a_{22}}|\epsilon^{'} \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
|x_n^{''} - \overline{x}_n| &\leq |\frac{a_{n1}}{a_{nn}}|\epsilon^{'} + |\frac{a_{n2}}{a_{nn}}|\epsilon^{'} + \cdots + |\frac{a_{nn-1}}{a_{nn}}|\epsilon^{'}
\end{aligned}
\tag{14}
$$

e poichè dalle disuguaglianze (10) segue che

$$\sum_{\substack{j=1 \\ (j\neq i)}}^{n} |\frac{a_{ij}}{a_{ii}}| < 1$$

posto

$$\lambda = \max_{i=1,\ldots,n} \sum_{\substack{j=1 \\ (j\neq i)}}^{n} |\frac{a_{ij}}{a_{ii}}|$$

le (14) forniscono semplicemente:

$$\epsilon^{''} \leq \lambda \epsilon^{'}. \tag{15}$$

Poichè dalle (10) segue anche facilmente che $\lambda < 1$, la disuguaglianza (15) ci dice che *la massima differenza in valore assoluto fra le componenti del vettore* $x_1^{''}, x_2^{''}, \ldots, x_n^{''}$ *e le componenti del vettore soluzione* $\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n$ *è minore della massima differenza in valore assoluto fra le componenti del vettore* $x_1^{'}, x_2^{'}, \ldots, x_n^{'}$ *e le componenti del vettore soluzione.*

In altre parole la disuguaglianza (15) significa questo: se noi partiamo da un vettore $\mathbf{x}^{'}$ di componenti $x_1^{'}, x_2^{'}, \ldots, x_n^{'}$ e lo trasformiamo, mediante le (13) nel vettore $\mathbf{x}^{''}$ di componenti $x_1^{''}, x_2^{''}, \ldots, x_n^{''}$, allora il vettore $\mathbf{x}^{''}$ è più *vicino* al vettore soluzione $\overline{\mathbf{x}}$ di componenti $\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n$ di quanto non lo fosse il vettore di partenza $\mathbf{x}^{'}$.

Tutto ciò suggerisce il seguente procedimento: si comincia con il fissare un vettore iniziale

$$x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)}$$

e quindi si calcola la successione di vettori

$$x_1^{(k)}, x_2^{(k)}, \ldots, x_n^{(k)}$$

definita dalla *regola iterativa*

$$
\begin{aligned}
x_1^{(k+1)} &= -\frac{a_{12}}{a_{11}}x_2^{(k)} - \frac{a_{13}}{a_{11}}x_3^{(k)} - \cdots\cdots - \frac{a_{1n}}{a_{11}}x_n^{(k)} + \frac{b_1}{a_{11}} \\
x_2^{(k+1)} &= -\frac{a_{21}}{a_{22}}x_1^{(k)} - \frac{a_{23}}{a_{22}}x_3^{(k)} - \cdots\cdots - \frac{a_{2n}}{a_{22}}x_n^{(k)} + \frac{b_2}{a_{22}} \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
x_n^{(k+1)} &= -\frac{a_{n1}}{a_{nn}}x_1^{(k)} - \frac{a_{n2}}{a_{nn}}x_2^{(k)} - \cdots\cdots - \frac{a_{nn-1}}{a_{nn}}x_{n-1}^{(k)} + \frac{b_n}{a_{nn}}
\end{aligned}
\tag{16}
$$

Se ora definiamo

$$\epsilon_k = \max_{i=1,\ldots,n} |x_i^{(k)} - \overline{x}_i|$$

e sottraiamo le (12) dalle (16), con gli stessi passaggi con cui abbiamo ottenuto la (15), otterremo ora

$$\epsilon_{k+1} \leq \lambda \epsilon_k \tag{17}$$

da cui segue

$$\epsilon_{k+1} \leq \lambda\epsilon_k \leq \lambda^2\epsilon_{k-1} \leq \lambda^3\epsilon_{k-2} \leq \ldots \leq \lambda^{k+1}\epsilon_0$$

e quindi in generale

$$\epsilon_k \leq \lambda^k \epsilon_0; \tag{18}$$

poichè è $\lambda < 1$, dalla (18) si deduce

$$\lim_{n\to\infty} \epsilon_k = 0$$

e in particolare avremo quindi che

$$\lim_{n\to\infty} x_i^{(k)} = \overline{x}_i \quad per \quad i = 1, 2, \ldots, n$$

Il metodo iterativo che abbiamo ora descritto e che è fondato sull'uso delle (16), va sotto il nome di *metodo di Jacobi*; esso risulterà *convergente* (cioè la successione di vettori $x_1^{(k)}, x_2^{(k)}, \ldots, x_n^{(k)}$ convergerà al vettore soluzione $\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n$) se valgono le (10), cioè se il sistema di equazioni lineari da risolvere è a diagonale strettamente dominante. Se invece il sistema non fosse a diagonale strettamente dominante il metodo di Jacobi potrebbe convergere in alcuni casi, ma non in tutti e l'analisi della convergenza diventerebbe più complicata.

Può essere utile, a questo punto, provare a ridescrivere il metodo di Jacobi utilizzando il simbolismo compatto che abbiamo imparato ad usare per le

operazioni fra matrici e vettori; prima di tutto, se noi definiamo la successione di vettori

$$\mathbf{x}^{(k)} = \begin{pmatrix} x_1^{(k)} \\ x_2^{(k)} \\ \dots \\ x_n^{(k)} \end{pmatrix}$$

la matrice

$$\mathbf{M} = \begin{pmatrix} 0 & -\frac{a_{12}}{a_{11}} & -\frac{a_{13}}{a_{11}} & \dots\dots & -\frac{a_{1n}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 & -\frac{a_{23}}{a_{22}} & \dots\dots & -\frac{a_{2n}}{a_{22}} \\ -\frac{a_{31}}{a_{33}} & -\frac{a_{32}}{a_{33}} & 0 & \dots\dots & -\frac{a_{3n}}{a_{33}} \\ \dots\dots & \dots\dots & \dots & \dots\dots & \dots \\ \dots\dots & \dots\dots & \dots & \dots\dots & \dots \\ -\frac{a_{n,1}}{a_{n,n}} & -\frac{a_{n,2}}{a_{n,n}} & -\frac{a_{n,3}}{a_{n,n}} & \dots\dots & 0 \end{pmatrix}$$

e il vettore

$$\mathbf{d} = \begin{pmatrix} \frac{b_1}{a_{11}} \\ \frac{b_2}{a_{22}} \\ \dots \\ \frac{b_n}{a_{nn}} \end{pmatrix}$$

le (16) si possono scrivere brevemente nella forma compatta

$$\mathbf{x}^{(k+1)} = \mathbf{M}\mathbf{x}^{(k)} + \mathbf{d}. \tag{19}$$

Se poi definiamo il vettore soluzione

$$\overline{\mathbf{x}} = \begin{pmatrix} \overline{x}_1 \\ \overline{x}_2 \\ \dots \\ \overline{x}_n \end{pmatrix}$$

le (12) si potranno scrivere analogamente nella forma compatta

$$\overline{\mathbf{x}} = \mathbf{M}\overline{\mathbf{x}} + \mathbf{d}. \tag{20}$$

Sottraendo ora (20) da (19) e definendo il vettore

$$\mathbf{e}^{(k)} = \mathbf{x}^{(k)} - \overline{\mathbf{x}}$$

otteniamo

$$\mathbf{e}^{(k+1)} = \mathbf{M}\mathbf{e}^{(k)}. \tag{21}$$

Il vettore $\mathbf{e}^{(k)}$ viene chiamato *vettore degli errori*, e infatti esso rappresenta (componente per componente) l'*errore che commetteremmo se ci arrestassimo alla k-esima iterazione e assumessimo* $\mathbf{x}^{(k)}$ *come approssimazione della soluzione esatta* $\overline{\mathbf{x}}$.

Osserviamo anche che l' ϵ_k che avevamo definito in precedenza, rappresenta semplicemente la massima componente in valore assoluto di $\mathbf{e}^{(k)}$, infatti è

$$\epsilon_k = \max_{i=1,\dots,n} |x_i^{(k)} - \overline{x}_i| = \max_{i=1,\dots,n} |e_i^{(k)}|$$

e quindi ϵ_k è semplicemente il *massimo errore* (su ciascuna componente) alla iterazione k-esima.

Dalla (21) e tenendo conto della definizione della matrice $\mathbf{M}$ si ricava poi subito la disuguaglianza (17) da cui segue la convergenza del metodo di Jacobi.

Ora che abbiamo capito come funziona questo metodo iterativo, ci sono però alcune domande a cui dobbiamo dare una risposta; la prima domanda è la seguente: *che convenienza possiamo avere ad usare un metodo iterativo invece di un metodo diretto come il metodo di eliminazione di Gauss?*.

Per rispondere a questa domanda, dobbiamo prima di tutto capire quale è il *costo di calcolo* del metodo di Jacobi e confrontarlo con quello del metodo di Gauss. Cominciamo col calcolare quante operazioni di prodotto o divisione dobbiamo eseguire per completare una singola iterazione: dal momento che per fare ciò occorre eseguire le operazioni previste dalle (16), è facile verificare che ad ogni iterazione del metodo, occorrerà eseguire $n(n-1)$ moltiplicazioni di coefficienti del tipo

$$\frac{a_{ij}}{a_{ii}}$$

per le componenti corrispondenti del tipo $x_j^{(k)}$; poichè le divisioni del tipo

$$\frac{a_{ij}}{a_{ii}} \quad o \quad del \quad tipo \quad \frac{b_i}{a_{ii}}$$

saranno invece sempre le stesse, noi possiamo supporre di averle eseguite e memorizzate alla prima iterazione e di non doverle quindi più ricalcolare. Se ne deduce quindi che il metodo di Jacobi richiede solo $n(n-1)$ moltiplicazioni ad ogni iterazione. Se quindi eseguiamo k iterazioni del metodo prima di arrestarci, il costo totale sarà di $kn(n-1)$ moltiplicazioni; a queste operazioni andrebbero aggiunte le divisioni per a_{ii} dei vari coefficienti che vengono eseguite solo alla prima iterazione, tuttavia, ragionando come abbiamo fatto nel caso del metodo di riduzione triangolare di Gauss, giungiamo facilmente

alla conclusione che il costo totale (in termini di operazioni di prodotto e divisione) del metodo di Jacobi sarà di circa kn^2 operazioni (dove k è il numero delle iterazioni eseguite). Poichè il costo del metodo di Gauss è invece di circa $\frac{1}{3}n^3$ operazioni di prodotto e divisione, ne deduciamo che il metodo di Jacobi risulterà *più conveniente* del metodo di Gauss se

$$kn^2 < \frac{1}{3}n^3$$

ovvero se

$$k < \frac{1}{3}n.$$

A questo punto, se vogliamo decidere quale fra i due metodi (di Gauss e di Jacobi) sia il più conveniente, dobbiamo decidere quanto vale k e quindi ci troviamo a dover rispondere ad un'altra domanda importante che è la seguente:*quante iterazioni dobbiamo eseguire con il metodo di Jacobi prima di fermarci?*

Abbiamo quindi bisogno di scegliere un *criterio di arresto* per il nostro metodo iterativo. Ecco come potremmo ragionare per ottenere un criterio del genere: prima di tutto consideriamo la seguente ovvia identità :

$$\mathbf{x}^{(k)} - \overline{\mathbf{x}} = \mathbf{x}^{(k)} - \mathbf{x}^{(k+1)} + \mathbf{x}^{(k+1)} - \overline{\mathbf{x}} = \mathbf{x}^{(k)} - \mathbf{x}^{(k+1)} + \mathbf{e}^{(k+1)}$$

da cui segue facilmente (ricordando la (21)) che

$$\mathbf{e}^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}^{(k+1)} + \mathbf{M}\mathbf{e}^{(k)};$$

da quest'ultima uguaglianza, applicando delle semplici disuguaglianze alle singole componenti, otteniamo con qualche facile passaggio

$$\max_{i=1,\dots,n} |e_i^{(k)}| \leq \max_{i=1,\dots,n} |x_i^{(k)} - x_i^{(k+1)}| + \lambda \max_{i=1,\dots,n} |e_i^{(k)}|$$

ovvero ancora

$$\epsilon_k \leq \delta_k + \lambda\epsilon_k$$

dove ϵ_k e λ sono gli stessi che abbiamo definito in precedenza, mentre per quanto riguarda δ_k esso non è altro che

$$\delta_k = \max_{i=1,\dots,n} |x_i^{(k)} - x_i^{(k+1)}|$$

e quindi δ_k rappresenta semplicemente la *massima differenza (in valore assoluto) fra le componenti delle ultime due iterazioni* .

Poichè sappiamo per ipotesi che $\lambda < 1$, possiamo dedurne allora che

$$\epsilon_k \leq \frac{1}{1-\lambda}\delta_k;$$

quest'ultima disuguaglianza e la (17) forniscono poi

$$\epsilon_{k+1} \leq \frac{\lambda}{1-\lambda}\delta_k \tag{22}$$

che è appunto la disuguaglianza che ci serve; infatti, supponiamo di conoscere il k-esimo vettore $\mathbf{x}^{(k)}$ della successione e calcoliamo il successivo $\mathbf{x}^{(k+1)}$ tramite le (19): a questo punto basta prendere la massima differenza (in valore assoluto) fra le componenti dei vettori $\mathbf{x}^{(k)}$ e $\mathbf{x}^{(k+1)}$ e moltiplicarla per il coefficiente $\frac{\lambda}{1-\lambda}$. Se il valore così ottenuto è minore della precisione che vogliamo raggiungere, ci fermeremo, altrimenti eseguiremo un'altra iterazione.

Il criterio di fermata che abbiamo ora proposto, è un criterio di fermata *a posteriori*, nel senso che esso ci consente di decidere di fermarci nel momento in cui viene raggiunta una certa precisione, tuttavia ciò avviene dopo un certo numero di iterazioni che non possiamo prevedere *a priori*; un criterio di fermata *a priori* dovrebbe consentirci di determinare il numero k di iterazioni che si dovranno eseguire per raggiungere una certa precisione *prima di iniziare a usare il metodo stesso.*

Se volessimo un criterio di fermata *a priori* potremmo ragionare al modo seguente: dalla disuguaglianza (18) sappiamo che dopo k iterazioni del metodo di Jacobi si avrà

$$\epsilon_k \leq \lambda^k \epsilon_0;$$

λ è un valore a noi noto, e quindi il problema sarà quello di valutare (o maggiorare) ϵ_0. Per fare ciò, eseguiamo la prima iterazione del metodo, cioè calcoliamo il vettore $\mathbf{x}^{(1)}$ a partire dal vettore iniziale $\mathbf{x}^{(0)}$ ed osserviamo che vale la seguente ovvia identità

$$\mathbf{x}^{(0)} - \overline{\mathbf{x}} = \mathbf{x}^{(0)} - \mathbf{x}^{(1)} + \mathbf{x}^{(1)} - \overline{\mathbf{x}} = \mathbf{x}^{(0)} - \mathbf{x}^{(1)} + \mathbf{e}^{(1)}$$

da cui segue facilmente (ricordando la (21)) che

$$\mathbf{e}^{(0)} = \mathbf{x}^{(0)} - \mathbf{x}^{(1)} + \mathbf{M}\mathbf{e}^{(0)};$$

da cui segue facilmente che

$$\epsilon_0 \leq \delta_0 + \lambda\epsilon_0$$

e quindi anche che

$$\epsilon_0 \leq \frac{1}{1-\lambda}\delta_0$$

che ci fornisce finalmente la maggiorazione di ϵ_0 che volevamo. Infatti, se ora torniamo a considerare la disuguaglianza (18), potremo maggiorarla con

$$\epsilon_k \leq \frac{\lambda^k}{1-\lambda}\delta_0$$

e quindi, se vogliamo che l'errore che noi commettiamo arrestandoci alla k-esima iterazione sia minore di un certo prefissato ϵ positivo assegnato, dovrà risultare che

$$\frac{\lambda^k}{1-\lambda}\delta_0 < \epsilon;$$

da quest'ultima disuguaglianza si deduce che il numero di iterazioni cercato è il minimo valore di k per cui vale

$$\lambda^k < \frac{\epsilon(1-\lambda)}{\delta_0};$$

se in quest'ultima disuguaglianza applichiamo il logaritmo ad entrambi i membri, otteniamo

$$k \log \lambda < \log \frac{\epsilon(1-\lambda)}{\delta_0}$$

che, poichè è $\lambda < 1$ e quindi è $\log \lambda < 0$, fornisce finalmente la disuguaglianza

$$k > \frac{\log \frac{\epsilon(1-\lambda)}{\delta_0}}{\log \lambda}; \tag{23}$$

il primo valore intero di k per cui vale la (23) sarà quindi il più piccolo numero di iterazioni con cui siamo certi di ottenere la precisione ϵ voluta.