

Del software di questo mese non pubblichiamo «il testo» in quanto eccessivamente lungo... Scherzi a parte una volta tanto i listati sono brevissimi (addirittura molto più brevi di quanto ci si immagini) mentre il testo dell'articolo è veramente completo; purtroppo se è sempre possibile eliminare i listati (almeno in parte) non è mai bello tagliare i testi. In questo caso poi il testo è veramente ben fatto e molto didattico e non era assolutamente il caso di eliminarne nemmeno una riga. Buona lettura a tutti...

I sistemi di equazioni lineari

di Maurizio Sichera - Milano

Il metodo di Gauss

Il metodo che gode di maggior popolarità è quello di Gauss. Si tratta del famigerato «metodo di sostituzione» che tutti abbiamo studiato alle scuole medie: mediante successive eliminazioni delle incognite dalle equazioni, si arriva ad un sistema in cui nell'ultima equazione compare solo l'ultima incognita, nella penultima equazione le ultime due incognite, e così via (sistema ridotto a forma triangolare); a questo punto si parte dal fondo, ricavando il valore dell'ultima incognita e sostituendolo nelle equazioni precedenti, poi si ricava la penultima incognita e si risale così fino alla prima. Il listato GAUSS1 implementa questo algoritmo: la matrice A(NEQ, NEQ) contiene inizialmente i coefficienti ed il vettore B(NEQ) i termini noti; al ritorno dal programma il vettore B contiene le soluzioni del sistema.

Questo metodo, nella sua forma elementare, presenta un grosso svantaggio: la divisione alla linea 10210 ci impone che gli elementi della diagonale principale della matrice dei coefficienti si mantengano diversi da zero durante le successive trasformazioni (poiché intorno a questi elementi ruota tutto l'algoritmo, vengono comunemente chiamati **pivot**). L'analisi numerica ci assicura che questa restrizione è rispettata

quando la matrice è simmetrica e definita positiva, o anche quando è a diagonale dominante (la prima condizione è molto complessa da verificare, mentre per la seconda basta una semplice ispezione visiva). L'analisi numerica però ci assicura anche che è sempre possibile permutare le righe della matrice in modo tale che i successivi pivot siano diversi da zero, a patto che la matrice dei coefficienti non sia singolare.

Le tecniche di scelta del pivot in modo che sia diverso da zero e quindi non arresti l'algoritmo sono chiamate, in genere, tecniche di pivotizzazione. La più semplice è la pivotizzazione parziale: nel generico stadio K della trasformazione, tra gli elementi della K-esima colonna A(K,K), A(K+1,K),..., A(NEQ,K) se ne sceglie come pivot uno diverso da zero: se questo elemento non giace sulla riga K, ma — ad esempio — sulla riga F, si scambiano fra di loro la K-esima e la F-esima equazione prima di proseguire con la trasformazione.

Implementando questo algoritmo, non conviene eseguire immediatamente lo scambio delle righe della matrice, ma memorizzare gli indici di permutazione in un vettore ausiliario LR e poi riordinare le soluzioni alla fine (vedi listato GAUSS2). Tra i tanti modi possibili di codificare il vettore delle permutazioni ho scelto il seguente: se — ad esempio — **LR(4)=7**, questo significa che la settima equazione del sistema originale si sarebbe spostata al quarto posto per effetto degli scambi; ma siccome gli scambi non sono stati effettivamente eseguiti, nel quarto passo di un generico loop sulle righe devo utilizzare i valori contenuti nella settima riga della matrice. Il criterio di scelta del pivot è semplicemente quello di prendere l'elemento che ha il massimo valore assoluto tra i possibili candidati: questa scelta ha il vantaggio non indifferente di ridurre la propagazione degli errori e quindi produce di solito risultati più accurati. Il vettore W viene usato per permutare in modo efficiente le soluzioni contestualmente al loro calcolo, mentre la variabile U# serve ad evitare il ricalcolo di una sottoespressione comune.

L'algoritmo GAUSS2 richiede ancora un commento: concettualmente ogni pivot dovrebbe essere scelto all'inizio del loop principale, immediatamente dopo la linea 10130, e questo comporterebbe un loop di ricerca. Poiché an-

Figura 1
Errore di macchina.

Hardware/Software	e	Formato	Base	Cifre	Formula	EM
MSX BASIC		SWG	10	6	0.5*10 ⁻⁰⁵	5.000E-6
Mainframe IBM		F	16	6	0.5*16 ⁻⁰⁵	4.768E-7
DEC PDP-11 (mod. vecchi)		F	2	24	2 ⁻²³	1.192E-7
DEC PDP-11 (mod. recenti)		F	2	24	0.5* 2 ⁻²³	5.960E-8
DEC VAX		F	2	24	0.5* 2 ⁻²³	5.960E-8
MS/DOS GWBASIC e IBM BASICA		SWG	2	24	0.5* 2 ⁻²³	5.960E-8
Standard IEEE P754		S	2	24	0.5* 2 ⁻²³	5.960E-8
Pocket Computer SHARP PC-1500			10	10	0.5*10 ⁻⁰⁹	5.000E-10
MSX BASIC		DBL	10	14	0.5*10 ⁻¹³	5.000E-14
DEC VAX		G	2	53	0.5* 2 ⁻⁵²	1.110E-16
Mainframe IBM		D	16	14	0.5*16 ⁻¹³	1.110E-16
Standard IEEE P754		D	2	53	0.5* 2 ⁻⁵²	1.110E-16
DEC PDP-11 (mod. vecchi)		D	2	56	2 ⁻⁵⁵	2.776E-17
DEC PDP-11 (mod. recenti)		D	2	56	0.5* 2 ⁻⁵⁵	1.388E-17
DEC VAX		D	2	56	0.5* 2 ⁻⁵⁵	1.388E-17
MS/DOS GWBASIC e IBM BASICA		DBL	2	56	0.5* 2 ⁻⁵⁵	1.388E-17
Standard IEEE P754		X	2	64	0.5* 2 ⁻⁶³	5.421E-20
DEC VAX		H	2	112	0.5*2 ⁻¹¹¹	1.926E-34

che i loop consumano tempo di calcolo, soprattutto nei linguaggi interpretati, un loop solo con più istruzioni è più efficiente di due loop con poche istruzioni; per questo motivo ho preferito scegliere il primo pivot al di fuori del loop principale (linee 10030-10110) e scegliere i successivi contestualmente alla trasformazione degli elementi della matrice (linee 10220-10230).

Per migliorare l'accuratezza dei risultati, si può, ad ogni stadio della trasformazione, scegliere come pivot l'elemento che ha massimo valore assoluto in tutta la sottomatrice restante, da $A(K,K)$ a $A(NEQ,NEQ)$ (pivotizzazione totale): questo comporta uno scambio sia delle righe che delle colonne, per cui introduciamo un vettore LC per memorizzare gli indici di permutazione di colonna (vedi listato GAUSS3). Con questo algoritmo la tecnica di ricercare il prossimo pivot contestualmente ad altre operazioni porta ad un risparmio di tempo piuttosto significativo.

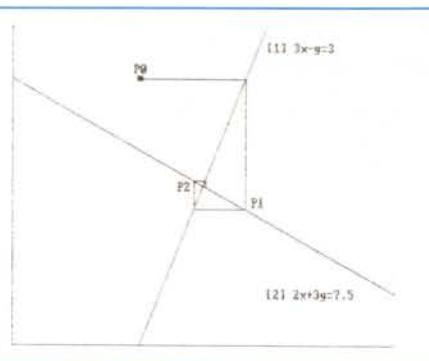
C'è ancora un accorgimento che si può adottare per migliorare ulteriormente l'accuratezza dei risultati. Supponiamo infatti di avere un sistema in cui tutti i coefficienti di una equazione siano molto maggiori dei coefficienti delle altre equazioni; supponiamo anche che, durante il procedimento di trasformazioni, tutti i coefficienti di questa equazione assumano valori molto più piccoli di quelli originali, ma comunque sempre maggiori di quelli delle altre equazioni. Se esaminiamo la linea 10210, vediamo subito che un risultato piccolo può derivare solo dalla sottrazione di due numeri con lo stesso segno e valori abbastanza vicini, e che quindi questo risultato è probabilmente affetto da un notevole errore; ma un coefficiente che presenta questo comportamento è un pessimo candidato a svolgere il ruolo di pivot, in quanto propagherebbe il suo errore su tutto il resto della matrice. Vediamo allora che il criterio di scegliere come pivot l'elemento di massimo valore assoluto (nella colonna o in tutta la matrice) va un po' corretto, facendo il rapporto tra il valore assoluto di ogni elemento candidato ed il massimo valore assoluto nella corrispondente riga della matrice originale e prendendo il candidato per cui questo rapporto è massimo. Questo accorgimento prende il nome di equilibratura della matrice e può essere accoppiato sia alla pivotizzazione parziale che a quella totale.

Il listato GAUSS4 mostra appunto un

$$\begin{array}{ccccccc} 1 & 0 & \dots & -A(1,K)/A(K,K) & \dots & 0 & \\ 0 & 1 & \dots & -A(2,K)/A(K,K) & \dots & 0 & \\ \dots & \dots & \dots & \dots & \dots & \dots & \\ 0 & 0 & \dots & 1/A(K,K) & \dots & 0 & \\ \dots & \dots & \dots & \dots & \dots & \dots & \\ 0 & 0 & \dots & -A(N,K)/A(K,K) & \dots & 1 & \end{array}$$

Figura 2 - Matrice usata nel metodo di Gauss-Jordan.

Figura 3 - Rappresentazione geometrica del metodo di Gauss-Seidel.



algoritmo per risolvere un sistema con il metodo di Gauss, applicando sia la pivotizzazione totale che l'equilibratura: in questo esempio il vettore W viene prima utilizzato per memorizzare i massimi di riga e poi per il riordinamento delle soluzioni. Una volta costruito il vettore dei massimi di riga, applicare la tecnica dell'equilibratura è di una banalità sconcertante: basta confrontare la linea 10220 con la stessa linea del listato GAUSS3 per rendersene conto.

Stima degli errori con il metodo di Gauss

Poiché la precisione con cui i calcolatori eseguono i calcoli non è infinita, i risultati di qualunque algoritmo sono, in genere, affetti da errori; la conoscenza dell'ordine di grandezza di questi errori è spesso molto importante, in quanto permette di sapere quanto sono affidabili i risultati. Se questo è vero in genere, diventa ancora più vero parlando del metodo Gauss, che è l'unico praticamente applicabile nel caso di sistemi «brutti», e che quindi può restituire risultati affetti da errori enormi senza fare una piega...

Dato che qui stiamo parlando di sistemi di equazioni, in cui la soluzione non è un singolo numero ma un intero vettore, occorre definire che cosa si intende per errore: detto $err(i)$ l'errore sulla singola componente, si definisce l'errore assoluto sull'intera soluzione come la norma del vettore errore.

$$EA = \max \{ \text{abs}(err(i)) \}$$

e l'errore relativo come rapporto tra EA e la norma del vettore soluzione

$$ER = EA / \max \{ \text{abs}(x(i)) \}.$$

Gli errori, prodotti da un metodo diretto come quello di Gauss sono dovuti

all'accumulo degli errori di arrotondamento introdotti durante l'esecuzione delle singole operazioni aritmetiche. Data una certa matrice, il fatto che gli errori di arrotondamento si compensino fra loro oppure si esaltino a vicenda è un fatto puramente casuale: esistono però certe combinazioni di valori nella matrice dei coefficienti (per esempio, la presenza contemporanea di elementi molto piccoli ed elementi molto grandi nella stessa riga e/o nella stessa colonna) per cui gli errori si esaltano molto più facilmente. Questo comportamento viene misurato mediante il numero di condizione: valori piccoli indicano un buon condizionamento (infatti la matrice identità ha numero di condizione pari ad 1), mentre valori grandi (1000 e più) indicano malcondizionamento e quindi errori probabilmente grandi.

Esistono diverse formule che permettono di calcolare l'errore, e tutte si basano sul numero di condizione: poiché il calcolo esatto di quest'ultimo è piuttosto laborioso (tra l'altro richiede il calcolo della matrice inversa), ci si può accontentare di una sua approssimazione; da qui si ricava una stima dell'errore, come mostra il listato GAUSS4ER. In questo sottoprogramma si costruisce una seconda colonna in termini noti, tale da produrre una soluzione il cui valore esatto sarebbe

$$y(i) = x(i)+1$$

(dove x è la soluzione del sistema originale). A causa degli errori, la differenza tra gli $x(i)$ e gli $y(i)$ effettivamente calcolati non sarà esattamente uguale a 1: basandosi su questa differenza, si arriva ad una stima del numero di condizione e quindi dell'errore relativo ed assoluto sul risultato, restituiti nelle variabili CN, ER ed EA rispettivamente. La formula utilizzata per la stima del-

```

10000 REM GAUSS1 - Metodo di Gauss - Algoritmo base
10010 REM -----
10130 FOR K = 1 TO NEQ-1
10180 FOR I = K+1 TO NEQ
10200 FOR J = K+1 TO NEQ
10210 A(I,J) = A(I,J) - A(I,K) * A(K,J) / A(K,K)
10240 NEXT J
10250 B(I) = B(I) - A(I,K) * B(K) / A(K,K)
10260 NEXT I
10270 NEXT K
10280 REM -----
10290 FOR K = NEQ TO 1 STEP -1
10310 FOR J = K+1 TO NEQ
10320 B(K) = B(K) - A(K,J) * B(J)
10330 NEXT J
10340 B(K) = B(K) / A(K,K)
10350 NEXT K
10380 RETURN

10000 REM GAUSS2 - Metodo di Gauss con pivotizzazione parziale
10010 REM -----
10020 DIM LR(NEQ), W(NEQ)
10030 WMAX = 0
10040 FOR I = 1 TO NEQ
10050 LR(I) = I
10070 TMP = ABS(A(I,1))
10090 IF WMAX < TMP THEN WMAX = TMP: IM = I
10110 NEXT I
10120 REM -----
10130 FOR K = 1 TO NEQ
10140 IF WMAX = 0 THEN PRINT "Matrice singolare": STOP
10150 IF IM <> K THEN SWAP LR(IM), LR(K)
10170 WMAX = 0
10180 FOR I = K+1 TO NEQ
10190 UM = A(LR(I),K) / A(LR(K),K)
10200 FOR J = K+1 TO NEQ
10210 A(LR(I),J) = A(LR(I),J) - UM * A(LR(K),J)
10220 IF I > K+1 THEN 10240
10230 TMP = ABS(A(LR(I),J)): IF WMAX < TMP THEN WMAX = TMP: IM = I
10240 NEXT J
10250 B(LR(I)) = B(LR(I)) - UM * B(LR(K))
10260 NEXT I
10270 NEXT K
10280 REM -----
10290 FOR K = NEQ TO 1 STEP -1
10300 UM = 0
10310 FOR J = K+1 TO NEQ
10320 UM = UM + A(LR(K),J) * W(J)
10330 NEXT J
10340 W(K) = (B(LR(K)) - UM) / A(LR(K),K)
10350 NEXT K
10360 FOR I = 1 TO NEQ: B(I) = W(I): NEXT I
10370 ERASE LR, W
10380 RETURN

10000 REM GAUSS4 - Metodo di Gauss con piv. totale ed equilibratura
10010 REM -----
10020 DIM LR(NEQ), LC(NEQ), W(NEQ)
10030 WMAX = 0
10040 FOR I = 1 TO NEQ
10050 LR(I) = I: LC(I) = 1: W(I) = 0
10060 FOR J = 1 TO NEQ
10070 TMP = ABS(A(I,J))
10080 IF W(I) < TMP THEN W(I) = TMP
10090 IF WMAX < TMP THEN WMAX = TMP: IM = I: JM = J
10100 NEXT J
10110 NEXT I
10120 REM -----
10130 FOR K = 1 TO NEQ
10140 IF WMAX = 0 THEN PRINT "Matrice singolare": STOP
10150 IF IM <> K THEN SWAP LR(IM), LR(K)
10160 IF JM <> K THEN SWAP LC(JM), LC(K)
10170 WMAX = 0
10180 FOR I = K+1 TO NEQ
10190 UM = A(LR(I),LC(K)) / A(LR(K),LC(K)): A(LR(I),LC(K)) = UM
10200 FOR J = K+1 TO NEQ
10210 A(LR(I),LC(J)) = A(LR(I),LC(J)) - UM * A(LR(K),LC(J))
10220 TMP = ABS(A(LR(I),LC(J))) / W(LR(I))
10230 IF WMAX < TMP THEN WMAX = TMP: IM = I: JM = J
10240 NEXT J
10250 B(LR(I)) = B(LR(I)) - UM * B(LR(K))
10260 NEXT I
10270 NEXT K
10280 REM -----
10290 FOR K = NEQ TO 1 STEP -1
10300 UM = 0
10310 FOR J = K+1 TO NEQ
10320 UM = UM + A(LR(K),LC(J)) * W(LC(J))
10330 NEXT J
10340 W(LC(K)) = (B(LR(K)) - UM) / A(LR(K),LC(K))
10350 NEXT K
10360 FOR I = 1 TO NEQ: B(I) = W(I): NEXT I
10370 ERASE LR, LC, W
10380 RETURN

```

```

10000 REM GAUSS3 - Metodo di Gauss con pivotizzazione totale
10010 REM -----
10020 DIM LR(NEQ), LC(NEQ), W(NEQ)
10030 WMAX = 0
10040 FOR I = 1 TO NEQ
10050 LR(I) = I: LC(I) = I
10060 FOR J = 1 TO NEQ
10070 TMP = ABS(A(I,J))
10090 IF WMAX < TMP THEN WMAX = TMP: IM = I: JM = J
10100 NEXT J
10110 NEXT I
10120 REM -----
10130 FOR K = 1 TO NEQ
10140 IF WMAX = 0 THEN PRINT "Matrice singolare": STOP
10150 IF IM <> K THEN SWAP LR(IM), LR(K)
10160 IF JM <> K THEN SWAP LC(JM), LC(K)
10170 WMAX = 0
10180 FOR I = K+1 TO NEQ
10190 UM = A(LR(I),LC(K)) / A(LR(K),LC(K))
10200 FOR J = K+1 TO NEQ
10210 A(LR(I),LC(J)) = A(LR(I),LC(J)) - UM * A(LR(K),LC(J))
10220 TMP = ABS(A(LR(I),LC(J)))
10230 IF WMAX < TMP THEN WMAX = TMP: IM = I: JM = J
10240 NEXT J
10250 B(LR(I)) = B(LR(I)) - UM * B(LR(K))
10260 NEXT I
10270 NEXT K
10280 REM -----
10290 FOR K = NEQ TO 1 STEP -1
10300 UM = 0
10310 FOR J = K+1 TO NEQ
10320 UM = UM + A(LR(K),LC(J)) * W(LC(J))
10330 NEXT J
10340 W(LC(K)) = (B(LR(K)) - UM) / A(LR(K),LC(K))
10350 NEXT K
10360 FOR I = 1 TO NEQ: B(I) = W(I): NEXT I
10370 ERASE LR, LC, W
10380 RETURN

```

```

10000 REM Inversione di matrice
10010 REM Metodo di Gauss-Jordan con pivotizzazione totale
10020 REM -----
10030 DIM LR(N), LC(N), W(N)
10040 WMAX = 0
10050 FOR I = 1 TO N
10060 LR(I) = -1: LC(I) = -1
10070 FOR J = 1 TO N
10080 TMP = ABS(A(I,J))
10090 IF WMAX < TMP THEN WMAX = TMP: IM = I: JM = J
10100 NEXT J
10110 NEXT I
10120 REM -----
10130 FOR K = 1 TO N
10140 IF WMAX = 0 THEN PRINT "Matrice singolare": STOP
10150 KR = IM: LR(JM) = KR
10160 KC = JM: LC(IM) = KC
10170 UP# = A(KR,KC)
10180 WMAX = 0
10190 FOR J = 1 TO N: IF J=KC THEN 10270
10200 UQ# = A(KR,J) / UP#: A(KR,J) = UQ#
10210 FOR I = 1 TO N: IF I=KR THEN 10260
10220 A(I,J) = A(I,J) - UQ# * A(I,KC)
10230 IF LR(J) > 0 OR LC(I) > 0 THEN 10260
10240 TMP = ABS(A(I,J))
10250 IF WMAX < TMP THEN WMAX = TMP: IM = I: JM = J
10260 NEXT I
10270 NEXT J
10280 FOR I = 1 TO N
10290 IF I <> KR THEN A(I,KC) = -A(I,KC) / UP#
10300 NEXT I
10310 A(KR,KC) = 1 / UP#
10320 NEXT K
10330 REM -----
10340 FOR J = 1 TO N
10350 FOR I = 1 TO N: W(LC(I)) = A(I,J): NEXT I
10360 FOR I = 1 TO N: A(I,J) = W(I): NEXT I
10370 NEXT J
10380 FOR I = 1 TO N
10390 FOR J = 1 TO N: W(LR(J)) = A(I,J): NEXT J
10400 FOR J = 1 TO N: A(I,J) = W(J): NEXT J
10410 NEXT I
10420 ERASE LR, LC, W
10430 RETURN

```

```

10000 REM Metodo di Gauss-Jordan (algoritmo base)
10010 REM -----
10120 FOR K = 1 TO NEQ
10130 FOR J = K+1 TO NEQ+1
10140 A(K,J) = A(K,J) / A(K,K)
10150 FOR I = 1 TO NEQ
10160 IF I <> K THEN A(I,J) = A(I,J) - A(I,K) * A(K,J) / A(K,K)
10170 NEXT I
10180 NEXT J
10190 NEXT K
10200 RETURN

```

Risultati di alcune prove

Tutti i sottoprogrammi sono stati provati su una macchina MS-DOS basata su processore 8088 con clock a 4,77 MHz e senza coprocessore aritmetico, utilizzando l'interprete GWBasic e lavorando sistematicamente in precisione doppia (DEFDBL A-Z: DEFINT I-N). I sistemi di prova sono stati generati in modo da essere piuttosto ben condizionati, utilizzando numeri casuali.

```

10000 REM GAUSS4ER - Metodo di Gauss con stima degli errori
10010 REM -----
10020 DIM LR(NEQ), LC(NEQ), W(NEQ), Y(NEQ)
10030 WMAX = 0
10040 FOR I = 1 TO NEQ
10050 LR(I) = 1: LC(I) = 1: W(I) = 0: Y(I) = B(I)
10060 FOR J = 1 TO NEQ
10070 TMP = ABS(A(I,J)): Y(I) = Y(I) + A(I,J)
10080 IF W(I) < TMP THEN W(I) = TMP
10090 IF WMAX < TMP THEN WMAX = TMP: IM = I: JM = J
10100 NEXT J
10110 NEXT I
10120 REM -----
10130 FOR K = 1 TO NEQ
10140 IF WMAX = 0 THEN PRINT "Matrice singolare": STOP
10150 IF IM <> K THEN SWAP LR(IM), LR(K)
10160 IF JM <> K THEN SWAP LC(JM), LC(K)
10170 WMAX = 0
10180 FOR I = K+1 TO NEQ
10190 UM = A(LR(I),LC(K)) / A(LR(K),LC(K)): A(LR(I),LC(K)) = UM
10200 FOR J = K+1 TO NEQ
10210 A(LR(I),LC(J)) = A(LR(I),LC(J)) - UM * A(LR(K),LC(J))
10220 TMP = ABS(A(LR(I),LC(J))) / W(LR(I))
10230 IF WMAX < TMP THEN WMAX = TMP: IM = I: JM = J
10240 NEXT J
10250 B(LR(I)) = B(LR(I)) - UM * B(LR(K))
10255 Y(LR(I)) = Y(LR(I)) - UM * Y(LR(K))
10260 NEXT I
10270 NEXT K
10280 REM -----
10290 FOR K = NEQ TO 1 STEP -1
10300 UM = 0
10310 FOR J = K+1 TO NEQ
10320 UM = UM + A(LR(K),LC(J)) * W(LC(J))
10330 NEXT J
10340 W(LC(K)) = (B(LR(K)) - UM) / A(LR(K),LC(K))
10350 NEXT K
10360 FOR I = 1 TO NEQ: B(I) = W(I): NEXT I
10370 REM -----
10380 T1 = 0: T2 = 0: T3 = 0: EM = 2^-56
10390 FOR I = 1 TO NEQ
10400 TMP = ABS(Y(I)-B(I)-1): IF T1 < TMP THEN T1 = TMP
10410 TMP = ABS(Y(I)-B(I)): IF T2 < TMP THEN T2 = TMP
10420 TMP = ABS(B(I)): IF T3 < TMP THEN T3 = TMP
10430 NEXT I
10440 TMP = T1/T2: CN = TMP/(NEQ*EM)
10450 IF TMP >> 0.5 THEN PRINT "Errore troppo grande": STOP
10460 TMP = TMP*CN*NE*EC: ER = (TMP*CN*EN)/(1-TMP): EA = ER*T3
10470 ERASE LR, LC, W, Y
10480 RETURN

```

```

11000 REM Metodo di Cholesky (delle radici quadrate)
11010 REM -----
11020 FOR K = 1 TO NEQ
11030 UM = A(K,K)
11040 FOR J = 1 TO K-1: UM = UM - A(K,J) * A(K,J): NEXT J
11050 IF UM <= 0 THEN PRINT "Matrice non definita positiva": STOP
11060 A(K,K) = SQR(UM)
11070 FOR I = K+1 TO NEQ
11080 UM = A(I,K)
11090 FOR J = 1 TO K-1: UM = UM - A(I,J) * A(K,J): NEXT J
11100 A(I,K) = UM / A(K,K)
11110 NEXT I
11120 NEXT K
11130 REM -----
11140 FOR K = 1 TO NEQ
11150 UM = B(K)
11160 FOR J = 1 TO K-1: UM = UM - A(K,J) * B(J): NEXT J
11170 B(K) = UM / A(K,K)
11180 NEXT K
11190 REM -----
11200 FOR K = NEQ TO 1 STEP -1
11210 UM = B(K)
11220 FOR I = K+1 TO NEQ: UM = UM - A(I,K) * B(I): NEXT I
11230 B(K) = UM / A(K,K)
11240 NEXT K
11250 RETURN

```

```

12000 REM Metodo di Gauss-Seidel
12010 REM -----
12020 IF MOON <= 0 THEN MOON = 50 ' Valore arbitrario
12030 IF EPS <= 0 THEN EPS = 1E-5 ' Valore arbitrario
12040 FOR I = 1 TO NEQ: X(I) = B(I) / A(I,I): NEXT I
12050 FOR K = 1 TO MOON
12060 VMX = 0: DIF = 0
12070 FOR I = 1 TO NEQ
12080 UM = B(I): T = X(I)
12090 FOR J = 1 TO NEQ
12100 IF J <> I THEN UM = UM - A(I,J) * X(J)
12110 NEXT J
12120 X(I) = UM / A(I,I)
12130 T = ABS(X(I)-T): IF DIF < T THEN DIF = T
12140 T = ABS(X(I)): IF VMX < T THEN VMX = T
12150 NEXT I
12160 IF DIF <= VMX*EPS THEN OK = -1: RETURN
12170 NEXT K
12180 OK = 0: RETURN

```

l'errore non restituisce il massimo errore teoricamente possibile, ma piuttosto una stima «ragionevole»: in casi sfortunati l'errore effettivo potrebbe essere molto maggiore di quello stimato, fino a 2^a (NE-1) volte più grande!

Nella formula compare la variabile EM che rappresenta l'errore di macchina, cioè il massimo errore relativo possibile su una singola operazione aritmetica: si può anche dire che EM è il più piccolo numero per cui il risultato dell'operazione $(1+EM)$ è ancora diverso da 1; questo errore dipende dalla rappresentazione dei numeri in virgola mobile e quindi varia al variare del calcolatore e del software utilizzato. La figura 1 mostra i valori di EM per un certo numero di macchine.

Nella formula (alla linea 10460) compaiono poi le variabili EC ed EN, che rappresentano l'errore relativo da cui sono affetti i coefficienti ed i termini noti rispettivamente: se questi valori sono il risultato di elaborazioni precedenti, i loro errori possono essere stimati; in caso contrario possono valere zero o, meglio, possono avere lo stesso valore di EM.

Il metodo di Cholesky

Il metodo di Gauss è, in fondo, l'esemplare più famoso della famiglia dei

metodi LR: l'idea di base è quella di trasformare la matrice dei coefficienti A nel prodotto di una matrice triangolare inferiore L e di una triangolare superiore R. In questo modo il sistema

$$A \cdot x = b$$

diventa

$$L \cdot R \cdot x = b$$

e quindi può essere risolto risolvendo in sequenza i sistemi

$$L \cdot y = b$$

$$R \cdot x = y$$

che hanno entrambi matrice dei coefficienti triangolare, e quindi non presentano particolari difficoltà. Il problema è quindi diventato quello di trovare due matrici triangolari L ed R, il cui prodotto sia uguale alla matrice originale A.

Se nel metodo di Gauss la matrice L non è esplicitamente visibile, e le stesse operazioni che trasformano A in R trasformano anche b in y, altri metodi invece seguono letteralmente questa tecnica e quindi prima scompongono la matrice dei coefficienti nel prodotto di due matrici triangolari e poi risolvono successivamente due sistemi. In particolare, se la matrice dei coefficienti è simmetrica e definita positiva, le matrici L e R possono essere scelte in modo che una sia la trasposta dell'altra; su questo concetto si basa appunto il me-

todo di Cholesky (listato Cholesky). Data la simmetria della matrice dei coefficienti, della matrice A vengono utilizzati e modificati esclusivamente quegli elementi che giacciono sulla diagonale principale o sotto la diagonale, mentre quelli al di sopra vengono completamente ignorati.

Il grosso pregio di questo metodo è di essere veloce (almeno 4 volte più veloce del metodo di Gauss con pivotizzazione ed equilibratura e 2 volte più veloce del metodo di Gauss senza fronzoli); anche l'accuratezza dei risultati è buona, ma questo è principalmente dovuto al fatto che il metodo è applicabile sotto condizioni piuttosto restrittive. L'unico svantaggio sta proprio nella difficoltà di sapere in anticipo se la matrice dei coefficienti è definita positiva: poiché non esistono criteri semplici per stabilirlo, il metodo di Cholesky può essere usato con sicurezza solo quando la natura del problema permette di stabilire a priori che questa condizione è rispettata.

Avvertenza

Poiché questo sottoprogramma utilizza la funzione SQR, occorre fare attenzione alla precisione con cui un certo linguaggio esegue il calcolo delle funzioni. Se queste sono calcolate in precisione semplice (ad esempio, se si usa il

GWBasic senza specificare l'opzione /D), l'utilizzo della precisione doppia nel resto dei calcoli non porta ad un miglioramento della precisione dei risultati.

Il metodo di Gauss-Jordan

Un altro metodo che gode di discreto favore è quello di Gauss-Jordan (anche questo ben noto dalla scuola media, con il nome di «metodo di riduzione», o «metodo di somma e sottrazione»). Se con il metodo di Gauss la matrice dei coefficienti viene ridotta a forma triangolare, con quello di Gauss-Jordan viene addirittura ridotta a forma diagonale, per cui il sistema è praticamente già risolto. Questo metodo può essere accompagnato dagli stessi accorgimenti di pivotizzazione e di equilibratura che si usano con il metodo di Gauss ed offre la stessa accuratezza nel calcolo delle soluzioni, ma ha il difetto di essere notevolmente più lento.

Il listato Gauss-Jordan mostra l'algoritmo di base, senza fronzoli. Come si può osservare, l'algoritmo procede essenzialmente per colonne e quindi conviene memorizzare sia i coefficienti che i termini noti in un'unica matrice A(NEQ,NEQ+1); concettualmente, ad ogni stadio K della trasformazione la matrice A viene moltiplicata a sinistra per una matrice costruita in un certo modo (vedi fig. 2). L'effetto di questa moltiplicazione di matrici è quello di sostituire la colonna K della matrice con la corrispondente colonna della matrice identità, per cui, dopo NEQ passi di trasformazione, le prime NEQ colonne contengono la matrice identità e la (NEQ+1)-esima contiene le soluzioni.

Anche con il metodo di Gauss-Jordan si può invertire una matrice risolvendo simultaneamente NEQ sistemi, analogamente a quanto si è già visto per il metodo di Gauss; possiamo quindi costruire una matrice di NEQ righe e 2*NEQ colonne, che contiene inizialmente la matrice data nella metà di sinistra e la matrice identità nella metà di destra. Applicando il metodo di Gauss-Jordan ad una tale matrice, si vede che ad ogni passo dell'algoritmo una sola colonna della metà di destra viene sostituita con valori diversi da quelli originali; d'altra parte, si è già visto che ad ogni passo una colonna della metà di sinistra viene sostituita con una colonna della matrice identità, per cui le colonne che contengono valori significativi sono in ogni istante solo NEQ.

Nell'implementazione pratica le nuove colonne che si sviluppano nella metà di destra vengono memorizzate al posto di quelle che si liberano nella metà di sinistra, e quindi si riesce ad invertire

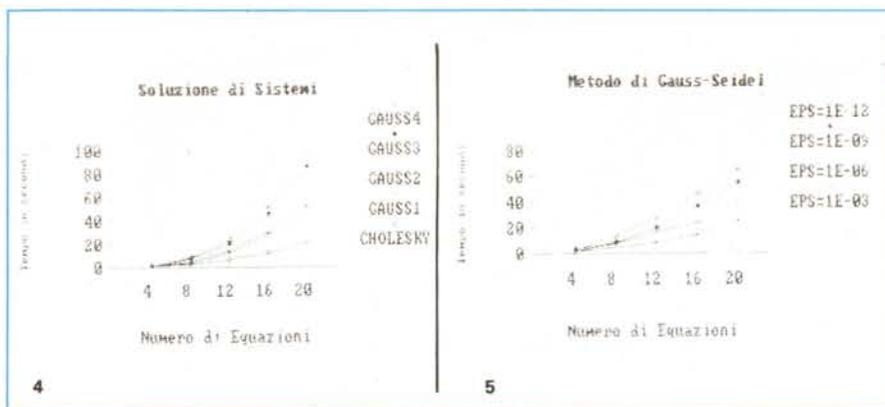


Figura 4 - Tempi di esecuzione al variare del numero di equazioni per il metodo di Cholesky e le diverse varianti di quello di Gauss.

Figura 5 - Tempi di esecuzione per il metodo di Gauss-Seidel, al variare del numero di equazioni e della precisione richiesta.

una matrice in se stessa: rispetto al metodo di Gauss abbiamo dimezzato lo spazio di lavoro necessario. Il listato INVMAT presenta appunto un sottoprogramma che inverte una matrice con il metodo di Gauss-Jordan, applicando la pivotizzazione totale.

Il metodo di Gauss-Seidel

Tutti i metodi visti fin qui sono metodi **diretti**, che arrivano alla soluzione con una sequenza finita di operazioni. Esistono però anche i metodi **iterativi**, in cui si stima in modo più o meno arbitrario una approssimazione iniziale della soluzione e poi si esegue ripetutamente un algoritmo che, ad ogni passo, porta ad una approssimazione migliore della precedente. Il procedimento termina quando si giudica che l'approssimazione sia abbastanza buona, oppure quando ci si rende conto che la successione delle soluzioni non converge.

Nel caso dei sistemi di equazioni lineari, possiamo assumere come «errore» la norma della differenza tra la soluzione precedente e quella appena ottenuta; quindi, ponendo

$$\text{err} = \max \{ \text{abs}(\text{old_x}(i) - \text{new_x}(i)) \}$$

$$\text{new} = \max \{ \text{abs}(\text{new_x}(i)) \}$$

il criterio di arresto per un qualunque metodo iterativo sarà del tipo

$$\text{err} \leq \text{eps} * \text{new}$$

Il metodo di Gauss-Seidel è il più noto tra i metodi iterativi per la soluzione dei sistemi lineari. Come si può vedere dalle linee 12070-12150 del listato, il metodo consiste nel ricavare il nuovo valore di X(I) risolvendo la I-esima equazione rispetto ad X(I): in questo modo il nuovo X(I) è calcolato basandosi sul vecchio valore di X(J) se J>I e sul nuovo valore di X(J) se J<I.

La convergenza di questo metodo è assicurata se la matrice dei coefficienti è a diagonale dominante. Per chiarire il perché di questa condizione, possiamo fare un esempio con due equazioni e

mostrare in un grafico cosa succede: la condizione matematica di dominanza della diagonale si traduce nella condizione geometrica che la prima retta formi con l'asse X un angolo maggiore di 45° e che la seconda formi un angolo minore di 45°. La figura 3 mostra le successive approssimazioni (P0, P1, P2) ottenute risolvendo il sistema

$$3x - y = 3$$

$$2x + 3y = 7.5$$

Se la condizione di dominanza della diagonale non è verificata, il metodo può convergere oppure no (provare per credere...).

Il programma usa come input la matrice A(NEQ,NEQ) ed il vettore B(NEQ), che contengono i coefficienti ed i termini noti rispettivamente, e lascia la soluzione cercata nel vettore X(NEQ). Inoltre richiede che la variabile EPS contenga la precisione (relativa) desiderata e la variabile MXN contenga il numero massimo di iterazioni da eseguire. Se la precisione EPS viene raggiunta in meno di MXN iterazioni, al ritorno la variabile OK viene posta uguale a -1, in caso contrario viene azzerata. Se le variabili EPS e MXN valgono 0, il sottoprogramma assegna opportuni valori standard.

Conclusioni

Concludendo questa panoramica sui vari metodi disponibili per risolvere i sistemi lineari, non si può dire che un solo metodo sia il migliore in tutti i casi: quello di Gauss è sicuramente applicabile sempre, ma è anche uno dei più lenti; quello di Gauss-Jordan è ancora più lento di quello di Gauss, ma è il migliore per invertire una matrice; quello di Cholesky è veloce, ma può essere applicato solo sotto condizioni restrittive; quello di Gauss-Seidel, infine, è applicabile sotto condizioni abbastanza restrittive ma facilmente verificabili, ma presenta tutti i vantaggi e gli svantaggi dei metodi iterativi.

POSTAL COMPUTER

PC XT IBM COMPATIBILE L. 750.000

SCHEDA MADRE 6/10 MHZ, 1 DRIVE 360K, SCHEDA CGA O HERCULES, 256K ESPANDIBILE A 640K SU PIASTRA, TASTIERA AVANZATA 101 TASTI

PC XT IBM COMPATIBILE L. 1.200.000

SCHEDA MADRE 6/10 MHZ, 1 DRIVE 360K, SCHEDA GRAFICA HERCULUS O CGA, 1 HARD DISK 20 MEGA, 256 ESPANDIBILE A 640K SU PIASTRA, TASTIERA AVANZATA 101 TASTI.

PC PHILIPS 9111

768K 1 DRIVE 5 1/4" e 1 DRIVE 3 1/2"
L. 1.200.000

MANNESMANN MT 81
L. 290.000

PC AT IBM COMPATIBILE L. 1.850.000

SCHEDA MADRE 80286, 12 MHZ, O WAIT, 512K ESPANDIBILE A 1024K, 1 DRIVE 5,25" DA 1.2 MB 1 HARD DISK DA 20 MB SCHEDA HERCULES O CGA TASTIERA AVANZATA 101 TASTI.

TELEFAX MURATA M-1 L. 1.300.000

- COMPATIBILITÀ: G2 G3
- VELOCITÀ DI TRASMISSIONE 15 SECONDI
- APPARECCHIO TELEFONICO A TASTIERA INCORPORATO
- FOTOCOPIATORE
- RICEZIONE AUTOMATICA
- ROTOLO CARTA TERMICA 216 mm x 30 metri.
- OROLOGIO/CALENDARIO DIGITALE

HARD DISK SEAGATE 20 MB	L. 350.000
HARD DISK SEAGATE 40 MB	L. 660.000
HARD DISK CONTROLLER PER XT	L. 100.000
HARD DISK CONTROLLER PER AT	L. 220.000
SCHEDA GRAFICA E.G.A.	L. 300.000
SCHEDA VGA	L. 430.000
SCHEDA SERIALE	L. 40.000
SCHEDA PARALLELA	L. 35.000
SCHEDA PORTA JOYSTICK	L. 28.000
SCHEDA MADRE XT	L. 140.000
SCHEDA MADRE AT (16 MHZ O WAIT)	L. 450.000
TASTIERA AVANZATA 101 TASTI	L. 110.000
DRIVE 5,25 360KB	L. 110.000
DRIVE 5,25 1,2MB	L. 170.000
DRIVE 3,50 720KB	L. 150.000
DRIVE CONTROLLER	L. 49.000
CAVO PARALLELO	L. 15.000
DATA SWITCH A 2 PORTE	L. 60.000
MOUSE ANKO	L. 59.000
JOYSTICK I.B.M. ANKO	L. 45.000

STAMPANTI CITIZEN GRAFICA - NLQ

CITIZEN 120 D L. 335.000 120 CPS, SET. EPSON IBM 80 COL. TRATO IN TRAZIONE, FRI- ZIONE INTER. OPZIONALE IBM/COMMODORE	CITIZEN MSP 50 L. 950.000 250/300 CAR/SEC., 80 COL.
CITIZEN LSP 100 L. 550.000 -160 cps, 80 COL.	CITIZEN MSP 55 L. 1.040.000 250/300 CAR/SEC., 136 COL.
CITIZEN MSP 10E L. 650.000 -160 CAR/SEC., 80 COL.	CITIZEN HQP 40 L. 920.000 - 24 AGHI, 200 CPS ALTISSIMA QUALITÀ
CITIZEN MSP 15E L. 539.000 160 CAR/SEC., 136 COL.	CITIZEN HQP 45 L. 1.350.000 - 24 AGHI, 200 CPS ALTISSIMA QUALITÀ
CITIZEN MSP 40 L. 610.000 - 200/240 CAR/SEC., 136 COL.	CITIZEN 180E COMPLETA DI INTERFACCIA IBM O COMMODORE - L.340.000
CITIZEN MSP 45 L. 750.000 - 200/240 CAR/SEC., 136 COL.	CITIZEN OVERTURE 110 L.3.600.000 - STAMPANTE LASER

**TUTTI I PRODOTTI CITIZEN SONO COPERTI
DA CERTIFICATO DI GARANZIA DELLA VALIDITÀ DI DUE ANNI**

OFFERTA MONITOR

PHILIPS		Segue PHILIPS	
MONITOR 8875 14" MULTISINK	L. 935.000	MONITOR 7749 14" TTL	L. 210.000
MONITOR 8833 14" CGA	L. 450.000	compatibile IBM sist. 2	L. 136.000
MONITOR 8802 14" COLORI	L. 360.000	MONITOR 7513 12" TTL	L. 183.000
MONITOR 9043 14" EGA	L. 535.000	MONITOR 7713 14" TTL	L. 183.000
MONITOR 9053 14" EGA	L. 595.000	ANTAREX	
MONITOR 9073 14" EGA	L. 680.000	BOXER 14" P39 JAN DUAL	L. 190.000
MONITOR 7723 14" TTL	L. 192.000	BIM 12" PC DM 216B	L. 135.000
MONITOR 7743 14" TTL	L. 205.000	CT 9000 SHR EGA JAN	L. 670.000
MONITOR 9082 14" VGA	L. 700.000	CT 9000/L MR14 DIM 414	L. 430.000

**PREZZI
SU RICHIESTA**

GARANZIA 12 MESI

**PREZZI IVA ESCLUSA
SPESE DI SPEDIZIONE ESCLUSE**

TEL. 06/3651688

TELEFONATECI