



"Didattica della Matematica per le Scelte Economiche e Finanziarie"

S.S.I.S. Puglia – a.a. 2006/2007

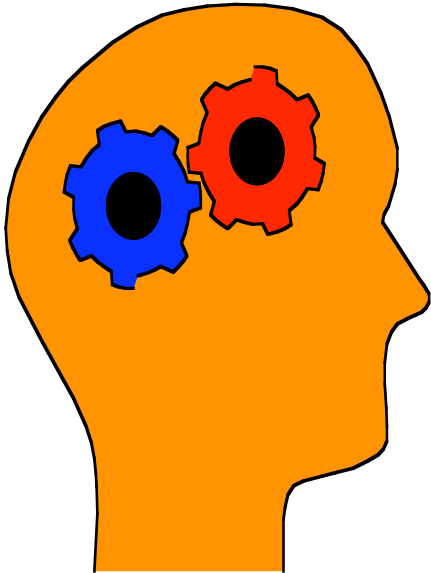
Prof. Crescenzo Gallo

Dipartimento di Scienze Economiche, Matematiche e Statistiche

Università di Foggia

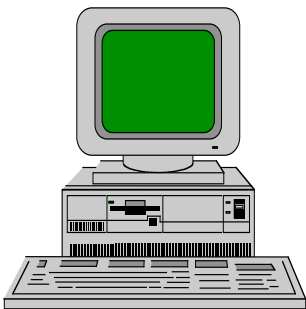
Marzo 2007

Programma di Matematica per il triennio degli Istituti Secondari Superiori



Obiettivi:

- avere consapevolezza del contributo della logica in ambito matematico
- sapere elaborare informazioni ed utilizzare consapevolmente metodi di calcolo e strumenti informatici
- saper tradurre e rappresentare in modo formalizzato problemi finanziari, economici e contabili attraverso il ricorso a modelli matematico-informatici



L'uso dell'elaboratore elettronico sarà via via potenziato utilizzando strumenti e metodi propri dell'informatica nei contesti matematici che vengono progressivamente sviluppati, come ad esempio:

- calcolo approssimato delle soluzioni di un'equazione algebrica o trascendente e di sistemi lineari;
- applicazioni al calcolo differenziale ed integrale;
- rappresentazione grafica di una funzione;
- applicazioni a fatti probabilistici e statistici;
- soluzione di un problema di programmazione lineare col metodo del simplesso;
- in genere risoluzione mediante programmazione su computer di problemi di ricerca operativa, matematica finanziaria ed attuariale e comunque del tipo semi-euristico.



L'elaboratore elettronico consente (oltre all'approfondimento delle conoscenze, dei linguaggi e dei metodi propri dell'informatica, mediante la visualizzazione di processi algoritmici non attuabile con elaborazione manuale) **la verifica sperimentale**

di nozioni teoriche già apprese e rafforza l'attitudine all'astrazione ed alla formalizzazione.

Obiettivo ultimo: dotare l'allievo - che sarà chiamato a svolgere un'attività di tipo decisionale per la quale non possono bastare intuito ed esperienza - di rigorosi metodi di analisi, di capacità relative alla modellizzazione di situazioni anche complesse, di abilità connesse al trattamento dei dati, che gli consentiranno di effettuare scelte consapevoli e razionali.

Esempio: Calcolo del massimo comune divisore

MCD (a , b) (a ≥ b; a , b ≥ 0)

Inizio

x ← a

y ← b

r ← x mod y

mentre (**r ≠ 0**) esegui

x ← y

y ← r

r ← x mod y

fineciclo

scrivi (**y**)

Fine.

Iterazione	a	b	x	y	r
--	22	6	22	6	4
1			6	4	2
2			4	2	0

↑
risultato finale

Esempio: Calcolo della radice quadrata $r = \sqrt{x}$

Inizio

$r \leftarrow 2.0$

$\epsilon \leftarrow 0.0001$

leggi (x)

ripeti

$y \leftarrow x / r$

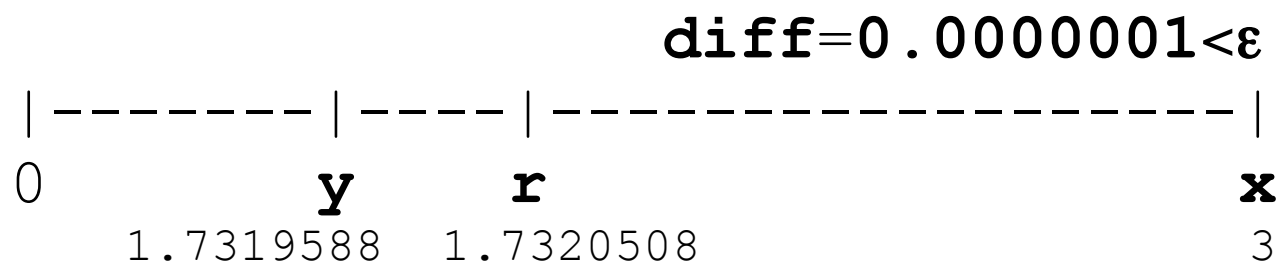
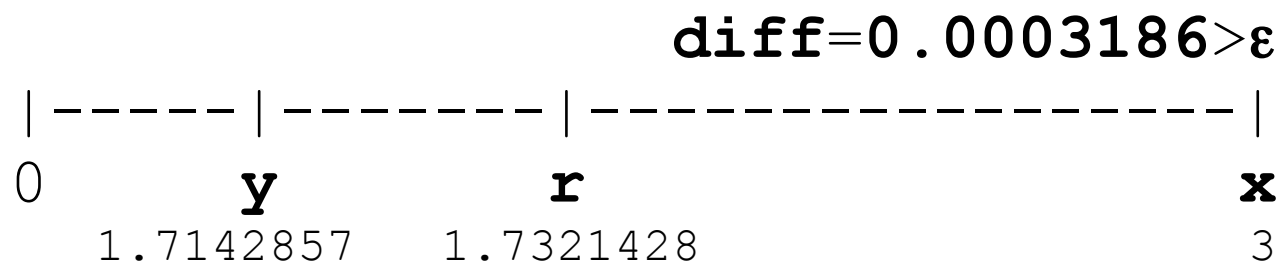
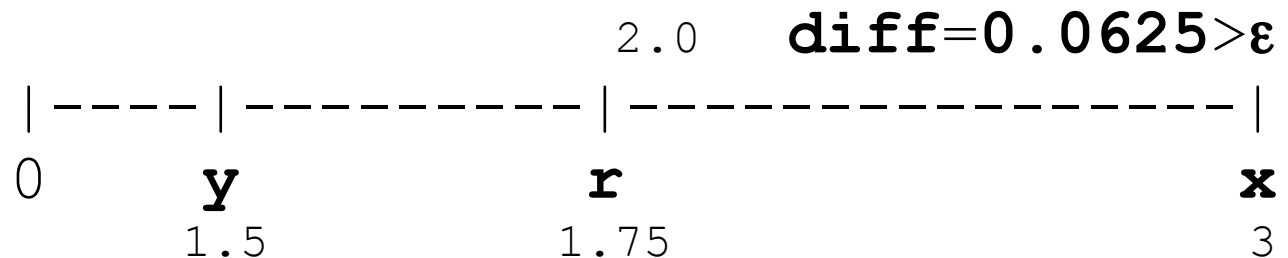
$r \leftarrow (y + r) / 2.0$

$\text{diff} \leftarrow |x - r^2|$

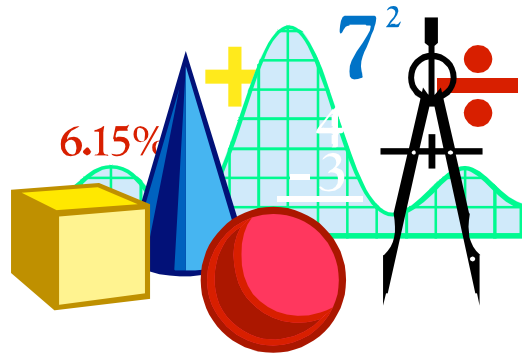
finché $\text{diff} \leq \epsilon$

scrivi (r)

Fine.



Il progresso della tecnica richiede una sempre più ampia utilizzazione dello *strumento matematico*, e tale esigenza si è potuta - almeno in parte - appagare grazie alla diffusione degli strumenti informatici.



Gli *elaboratori* hanno permesso di dominare con precisi processi numerici fenomeni che, fino a pochi anni fa, sembravano sottrarsi ad una sistematica trattazione quantitativa.

Ciò ha influito, d'altra parte, sulla *struttura* stessa del *pensiero matematico*, che è stato indotto a rivolgere la sua attività verso forme che erano state, in passato, meno curate.

Alcuni aspetti della ricerca scientifica hanno ricevuto da questo rinnovato spirito una decisiva spinta, lasciandoci intravedere *nuovi orizzonti* la cui completa scoperta è uno degli scopi della ricerca attuale.

Tale è ad esempio il caso delle scienze che sfruttano la statistica ed il calcolo delle probabilità su masse di dati particolarmente rilevanti come la medicina, la ricerca operativa, la "management science", la fisica delle particelle subatomiche, etc.

È oggi indispensabile la *conoscenza* dei *processi fondamentali* dell'*analisi numerica*, a causa dell'espansione nel campo provocata dallo sviluppo delle moderne tecnologie informatiche, ma anche tenendo conto del fatto che una notevole quantità di calcoli continuerà ad essere compiuta manualmente: la familiarità con il *saper calcolare* (calcolo manuale ed uso di una calcolatrice) è un auspicabile esercizio preliminare al calcolo sui computer.

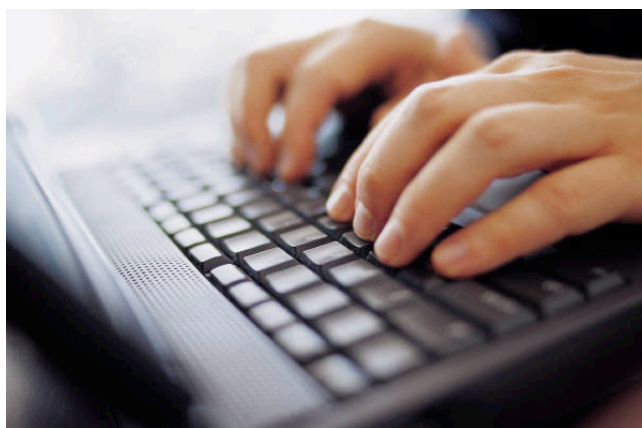
Mentre la disponibilità di calcolatori sempre più veloci ha reso possibile la *soluzione numerica* di molti *problemi* che precedentemente parevano di una insuperabile complessità, l'efficace uso di tali strumenti dipende in massima parte dal continuo progresso della ricerca nei campi connessi della matematica.

Inoltre, vi è una continua *necessità* di *personale* sufficientemente ben *preparato* sui principi e le tecniche esistenti del *calcolo numerico*, e che sia in grado non solo di adattare ad una data macchina i metodi disponibili, ma anche di comprendere le limitazioni proprie di ciascun metodo e di escogitare procedimenti di **controllo degli errori** e tecniche modificate che siano particolarmente appropriate a quella macchina.

L'urgenza della necessità di tale personale richiede un addestramento che conduca il più rapidamente possibile ad una buona *conoscenza operativa* delle tecniche associate ad alcuni degli aspetti più complessi del campo, non perdendo nel contempo di vista l'esame degli elementari processi fondamentali su cui queste tecniche sono basate.

Un tale *processo formativo* dovrebbe non solo esporre tecniche valide per ciascuno scopo, ma dovrebbe pure tentare di derivare le formule relative in modo tale che le ipotesi fondamentali siano messe in evidenza, i metodi di generalizzazione e modificazione siano ragionevolmente chiari, e che i problemi di **analisi degli errori, convergenza e stabilità** siano trattati tanto adeguatamente quanto lo permettono tempo e preparazione.

Il tutto dovrebbe essere accompagnato dall'*applicazione su computer* mediante esercizi pratici, nei quali vengono effettuati calcoli abbastanza reali per stabilire l'importanza pratica degli sviluppi teorici.



Esempio: Calcolo ricorsivo del fattoriale n!

$$\begin{cases} n! = n \cdot (n - 1)! \\ 0! = 1 \end{cases}$$

```
Function FATT (N: Integer) As Long
  If N=0 Then
    FATT := 1
  Else
    FATT := N*FATT (N-1)
  End If
End Function
```

$$3! = 3 \cdot 2!$$

$$2! = 2 \cdot 1!$$

$$1! = 1 \cdot 0!$$

$$0! = 1$$

FATT(3)=6

$$N=3 \Rightarrow \text{FATT} := 3 * \underline{\text{FATT}(2)}$$

$$N'=2 \Rightarrow \text{FATT}' := 2 * \underline{\text{FATT}(1)} = 2$$

$$N''=1 \Rightarrow \text{FATT}'' := 1 * \underline{\text{FATT}(0)} = 1$$

$$N'''=0 \Rightarrow \text{FATT}''' := 1$$

Esempio: Metodo di Newton-Raphson

Serve a determinare lo zero di una funzione derivabile $f(x)$, cioè a trovare \mathbf{x}^* tale che $f(\mathbf{x}^*) = 0$.

Scelto un punto iniziale x_0 , si determina la tangente a $f(x)$ in x_0 (pari alla derivata $f'(x_0)$) e la relativa intersezione con l'asse x (ponendo $y=0$), che sarà l'ascissa del punto successivo x_k , per $k=1, 2, \dots$; si approssima cioè la curva $f(x)$ con la tangente nel punto x_k per determinare il successivo punto $x=x_{k+1}$:

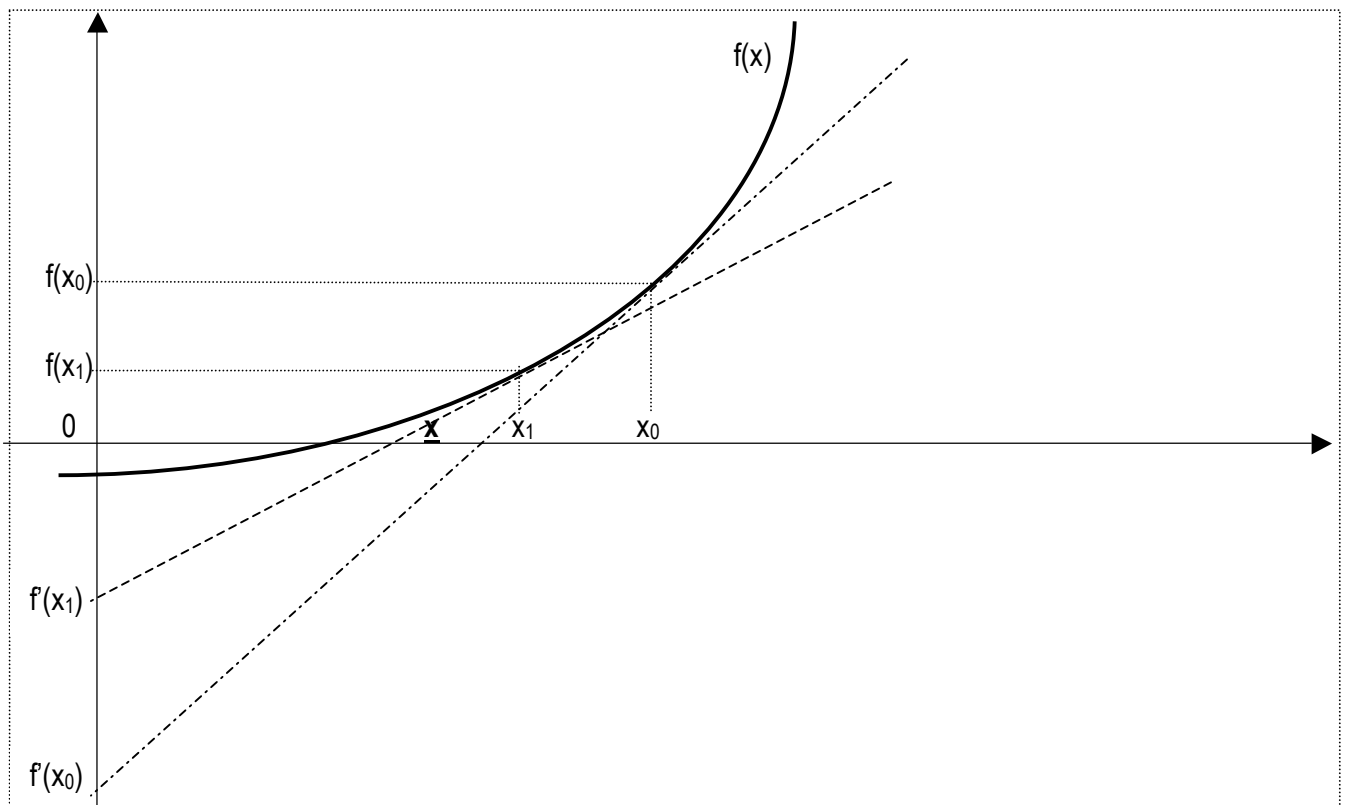
$$f'(x_k) = [y - f(x_k)] / [x - x_k]$$

(essendo la derivata il coefficiente angolare della retta tangente); intersecando quindi con l'asse x (ponendo cioè $y=0$) si ha:

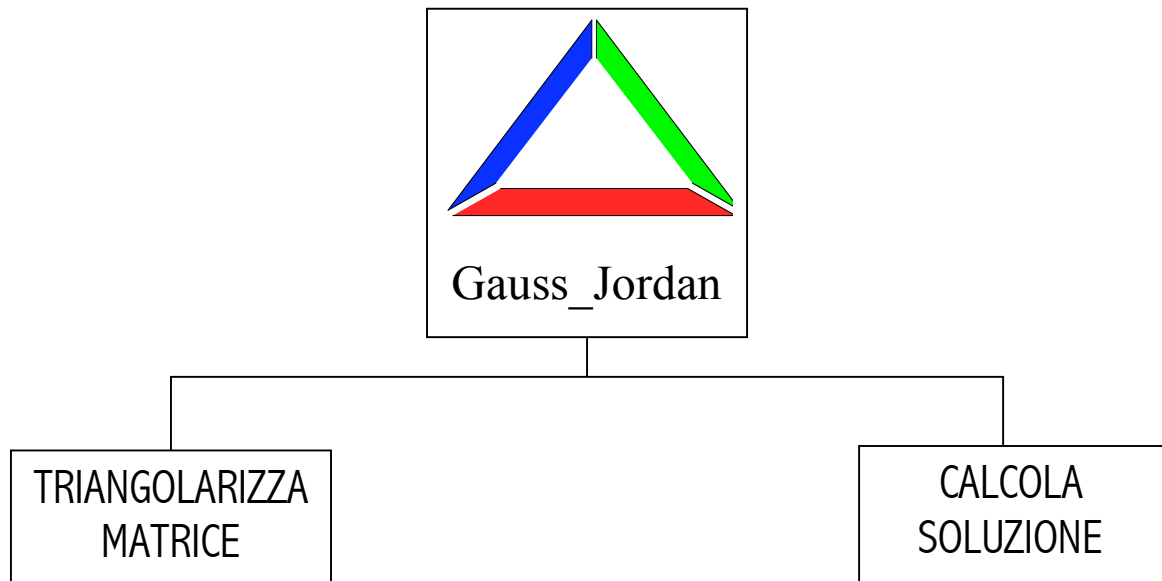
$$f'(x_k) = -f(x_k) / [x - x_k] \Rightarrow \mathbf{x} = \mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{f}(\mathbf{x}_k)/\mathbf{f}'(\mathbf{x}_k)$$

sino ad arrivare al punto \mathbf{x}^* desiderato.

La convergenza dell'algoritmo può essere stabilita sul numero di cifre approssimate del risultato, ad es.: $|x_{k+1} - x_k| < \epsilon$, oppure su un limite massimo di iterazioni oltre il quale si accetta comunque il risultato ottenuto.



Algoritmo di **Gauss-Jordan** per la risoluzione di un sistema lineare: $A \cdot X = C$



‘Ipotizziamo che i dati (N, A, C) e i risultati (X) siano gestiti a parte (ad es. in celle di un foglio di lavoro).

Option Explicit

Const **MAX_N** = 20 ‘n.ro massimo di incognite

‘**Variabili di input:**

Dim **N** as integer ‘dimensione effettiva del sistema

Dim **A**(1 to **MAX_N**, 1 to **MAX_N**+1) as single ‘matrice del sistema

Dim **C**(1 to **MAX_N**) as single ‘colonna dei termini noti

‘**Variabili di output:**

Dim **X**(1 to **MAX_N**) as single ‘vettore soluzione

‘**Variabili (globali) di lavoro:**

Dim **PASSO** as integer, **RIGA** as integer

```
Sub Gauss_Jordan()  
    TRIANGOLARIZZA_MATRICE  
    CALCOLA_SOLUZIONE  
End Sub
```

```
Sub TRIANGOLARIZZA_MATRICE()  
    'Triangolarizzo la matrice A del sistema  $AX=C$ :  
    For PASSO=1 To N  
        DIVIDI_PER_IL_PIVOT  
        For RIGA=PASSO+1 To N  
            AGGIORNA_IL_RESTO  
        Next RIGA  
    Next PASSO  
End Sub
```

```
Sub CALCOLA_SOLUZIONE()  
    'Calcolo il vettore soluzione X:  
    For RIGA=N To 1 Step -1  
        X(RIGA)=C(RIGA)  
        For PASSO=N To RIGA+1 Step -1  
            X(RIGA)=X(RIGA)-A(RIGA,PASSO)*X(PASSO)  
        Next PASSO  
    Next RIGA  
End Sub
```

```
Sub DIVIDI_PER_IL_PIVOT()  
Dim J as Integer  
    'Divido per il Pivot la riga...  
    For J=PASSO+1 To N  
         $A(\text{PASSO},J)=A(\text{PASSO},J)/A(\text{PASSO},\text{PASSO})$   
    Next J  
    '... e il termine noto corrente  
     $C(\text{PASSO})=C(\text{PASSO})/A(\text{PASSO},\text{PASSO})$   
    'Pongo il Pivot = 1  
     $A(\text{PASSO},\text{PASSO})=1$   
End Sub
```

```
Sub AGGIORNA_IL_RESTO()  
    'Ricalcolo la riga ...  
    For J=PASSO+1 To N  
         $A(\text{RIGA},J)=A(\text{PASSO},J)*A(\text{RIGA},\text{PASSO})-A(\text{RIGA},J)$   
    Next J  
    '... ed il termine noto corrente  
     $C[\text{RIGA}]:=C[\text{PASSO}]*A[\text{RIGA},\text{PASSO}]-C[\text{RIGA}];$   
    'Azzero gli elementi restanti sotto al Pivot  
     $A(\text{RIGA},\text{PASSO})=0;$   
End Sub
```

$$\begin{cases} 2x - 3y = -4 \\ x + y = 3 \end{cases}$$

$$\left[\begin{array}{cc|c} \underline{2} & -3 & -4 \\ 1 & 1 & 3 \end{array} \right] \longrightarrow \left[\begin{array}{cc|c} 1 & -1.5 & -2 \\ 0 & \underline{-2.5} & -5 \end{array} \right]$$

$$\left[\begin{array}{cc|c} 1 & -1.5 & -2 \\ 0 & 1 & 2 \end{array} \right] \begin{array}{l} x_1 \\ x_2 \end{array}$$

$$\mathbf{x_2 = 2}$$

$$\mathbf{x_1 = -2 + 1.5 \cdot 2 = -2 + 3 = 1}$$

La programmazione lineare



Problemi di programmazione si devono affrontare ogni qualvolta si debbono svolgere un certo numero di attività e si disponga di una quantità limitata di risorse o si debbano *rispettare dei limiti nell'assegnazione delle risorse disponibili alle singole attività*.

In situazioni di questo genere il problema consiste nel *distribuire le risorse disponibili tra le varie attività in modo tale da ottimizzare l'efficienza complessiva*.

I possibili modi di *assegnare le risorse* disponibili possono essere *in numero finito*, ma possono anche essere infiniti. Nei casi in cui le scelte possibili siano finite si potrebbe, in teoria, enumerarle tutte.

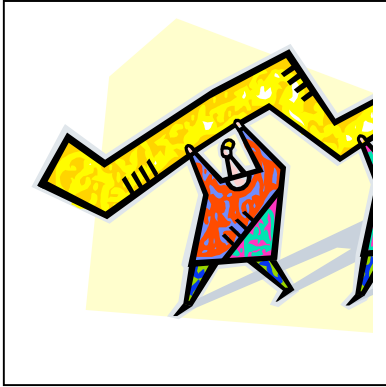
In pratica, purtroppo, l'*enumerazione* richiede quasi sempre di gran lunga troppo tempo: si pensi, ad esempio, che vi sono ben $10!$ modi diversi di distribuire dieci contratti, ove se ne voglia dare uno a ciascuno di dieci offerenti.

Anche nel caso di scelte continue - come per la divisione di un capitale tra più possibili investimenti - le condizioni cui devono soddisfare le variabili del problema sono spesso di natura tale da non ammettere una trattazione matematica in termini di calcolo infinitesimale.

È solo da pochi anni che i matematici hanno compreso che, ai fini pratici, dovevano essere elaborati *concreti metodi di calcolo* anche per quei problemi che in linea teorica erano già risolti mediante l'enumerazione.

Ovviamente, i primi *problemi* ad essere *affrontati* sono stati quelli (relativamente semplici) in cui l'efficienza e le condizioni limitative potevano essere espresse come *funzioni lineari delle risorse da assegnare*. Lo studio di situazioni di questo genere va sotto il nome di "*Programmazione Lineare*" ed i metodi pertinenti possono essere divisi in tre principali gruppi.

Essi sono:



Problemi di assegnazione. Questo gruppo di problemi è caratterizzato dall'esistenza di una *matrice di efficienza*, rappresentante in modo sintetico ciò che accade quando si associa un certo numero di "*origini*" con un egual numero di "*destinazioni*". Ogni origine deve essere associata ad una ed una sola destinazione, ed il problema consiste nell'*effettuare le associazioni in modo da rendere massima (o minima) l'efficienza complessiva*. Il citato esempio dei dieci contratti da distribuire fra dieci offerenti è un problema di assegnazione.



Problemi di trasporto. Questo gruppo di problemi può essere riguardato come una generalizzazione del precedente. La matrice di efficienza non è più necessariamente quadrata: *un certo numero di "origini" può essere qui associato ad un diverso numero di "destinazioni"*. Si conosce il movimento complessivo in partenza da ogni origine, ed il movimento totale in arrivo ad ogni destinazione: si vuole determinare il modo migliore di associare origini e destinazioni nell'ambito imposto da queste limitazioni.

Problemi risolvibili con il metodo del simplesso.

Questo gruppo comprende sia i problemi di assegnazione che quelli di trasporto, ma il metodo usato per determinare la soluzione è piuttosto faticoso, per cui i metodi caratteristici dei precedenti due gruppi dovrebbero essere usati in tutti i casi in cui ciò è possibile. Il tipo di problema che richiede il metodo del simplesso si illustra meglio con un esempio, sia pure senza entrare nei dettagli matematici.

Supponiamo che una macchina possa fare 400 pezzi del tipo (1) e 300 pezzi del tipo (2) in una settimana; se, nel corso della settimana, la produzione cambia da (1) a (2) o viceversa la macchina può produrre in tutto 600 pezzi. Il guadagno delle vendite di (1) è di € 20,00 al pezzo, di (2) è di € 50,00 al pezzo. Si vogliono determinare le quantità x_1 ed x_2 di pezzi che devono essere prodotte ogni settimana in modo da rendere massimo il profitto.

In termini matematici, il problema consiste nel trovare il massimo dell'espressione: $z=2x_1+5x_2$ sotto le condizioni: $0 \leq x_1 \leq 400$, $0 \leq x_2 \leq 300$, $x_1+x_2 \leq 600$.

In generale, si tratta di determinare il massimo (o il minimo) di una funzione lineare di un certo numero di variabili non negative, sotto condizione che le variabili stesse soddisfino una serie di disequazioni lineari.

Quella che segue non intende essere una dimostrazione dettagliata del metodo di risoluzione del semplice, ma semplicemente una tecnica di calcolo che aiuti praticamente nella soluzione di problemi di questo tipo. È bene inoltre ricordare che, nei calcoli di questo tipo, i problemi che si incontrano in pratica interessano un numero così grande di variabili da non poter essere affrontati con il solo ausilio delle consuete calcolatrici da tavolo. Probabilmente non è un caso se lo sviluppo dei metodi di programmazione lineare ha coinciso storicamente con la disponibilità dei (grandi) calcolatori.



SIMPLEX

*Soluzione di un problema di programmazione lineare
col metodo del simplesso*

Autore: Prof. Crescenzo Gallo

Linguaggio: VB

Data di revisione: marzo 2007

Questo programma risolve un problema di programmazione lineare con il metodo del simplesso. Il problema è del tipo:

$$\max Z = \sum_{i=1}^n c_i \cdot x_i$$

soggetta a: $\mathbf{AX} \leq \mathbf{B}$

Se il problema non è risolvibile, viene emesso un opportuno messaggio, altrimenti vengono di volta in volta visualizzate le soluzioni parziali, ed alla fine della convergenza dell'algoritmo la soluzione ottima.

'Il programma è scritto in VisualBasic for Applications

Option Explicit

Const MAX_M=10 ' n.ro massimo di vincoli (righe) del modello (= n.ro di variabili in base)

Const MAX_N=6 'n.ro massimo di variabili (colonne)

'Variabili di Input:

Dim C(1 to MAX_N) as single 'vettore dei coefficienti

Dim A(1 to MAX_M,1 to MAX_N) as single 'matrice del modello

Dim N as integer 'n.ro effettivo di variabili del modello da risolvere

Dim M as integer 'n.ro effettivo di vincoli del modello

'Variabili di Output:

Dim Z as single 'valore ottimale della funzione obiettivo

Dim X(1 to MAX_M) as single 'vettore della soluzione; inizialmente ospita il vettore B dei termini noti del modello

'Variabili di lavoro:

Dim PIVOT as single 'elemento centrale per l'applicazione delle formule di trasformazione di Gauss ad ogni iterazione

Dim IN_BASE(1 to MAX_M) as integer 'indici delle variabili in base

Dim FUORI_BASE(1 to MAX_N) as integer 'indici delle variabili fuori base

Dim SOLUZIONE_OTTIMA as boolean 'flag indicante la convergenza dell'algoritmo

Dim ERRORE as boolean 'vero se l'algoritmo non converge o vi è errore

Dim E as integer 'indice della variabile entrante

Dim U as integer 'indice della variabile uscente

Dim MESSAGGIO as string

```
sub SIMPLEX()  
  LEGGI_DATI_MODELLO  
  INIZIALIZZA_TABLEAU  
  
  do  
    DET_VAR_ENTRANTE  
    DET_VAR_USCENTE  
    RICALCOLA_Z_X_C  
    if not ERRORE then  
      MESSAGGIO="** SOLUZIONE PARZIALE **"  
      VISUALIZZA_SOLUZIONE  
      TEST_DI_OTTIMALITA  
      if not SOLUZIONE_OTTIMA then  
        RICALCOLA_MATRICE_A  
      end if  
    end if  
  loop until SOLUZIONE_OTTIMA or ERRORE  
  
  if not ERRORE then  
    MESSAGGIO="** SOLUZIONE OTTIMA **"  
    VISUALIZZA_SOLUZIONE  
  end if  
  
end sub
```

```
sub ERROR_ROUTINE()  
    MsgBox MESSAGGIO, vbExclamation  
end sub
```

```
sub LEGGI_DATI_MODELLO()  
dim i as integer, j as integer  
Cells(1, 1) = "--- METODO DEL SIMPLESSO: max Z=CX soggetta a: AX<=B ---"  
Cells(3, 1)="N.ro di vincoli (max " & MAX_M & "): "  
M=Val(Cells(3, 3))  
Cells(4, 1)= "N.ro di variabili (max " & MAX_N "): "  
N=Val(Cells(4, 3))  
Cells(6, 1)= "Vettore dei coefficienti della funzione obiettivo: "  
for i=1 to N  
    C(i)=Cells(7, i)  
next i  
Cells(9, 1)="Matrice dei coefficienti del modello: "  
for i=1 to M  
    for j=1 to N  
        A(i, j)=Cells(9+i, j)  
    next j  
next i  
Cells(21, 1)= "Vettore dei termini noti: "  
for i=1 to M  
    X(i)=Cells(22, i)  
next i  
end sub
```

```
sub INIZIALIZZA_TABLEAU()  
dim j as integer, i as integer  
for j=1 to N  
    C(j)=-C(j)  
    FUORI_BASE(j)=j  
next j  
Z=0  
for i=1 to M  
    IN_BASE(i)=N+i  
next i  
SOLUZIONE_OTTIMA=false  
ERRORE=false  
end sub
```

```

sub DET_VAR_ENTRANTE()
dim j as integer
  E=1
  for j=2 to N
    if C(j)<C(E) then E=j
  next j
end sub

```

```

sub DET_VAR_USCENTE()
Dim PRIMO_POSITIVO as integer 'indice del primo rapporto positivo  $b_i/a_{i,e}$ 
Dim RMIN as single 'valore del rapporto minimo  $b_i/a_{i,e}$ 
Dim R as single, ROW as integer
  PRIMO_POSITIVO=1
  do while (A(PRIMO_POSITIVO,E)<=0) and (PRIMO_POSITIVO<M)
    PRIMO_POSITIVO=PRIMO_POSITIVO+1
  loop
  RMIN=X(PRIMO_POSITIVO)/A(PRIMO_POSITIVO,E)
  U=PRIMO_POSITIVO
  for ROW=PRIMO_POSITIVO+1 to M
    if A(ROW,E)>0 then
      R=X(ROW)/A(ROW,E)
      if R<RMIN then
        U=ROW
        RMIN=R
      end if
    end if
  next ROW
  ROW=IN_BASE(U)
  IN_BASE(U)=FUORI_BASE(E)
  FUORI_BASE(E)=ROW
end sub

```



```

sub RICALCOLA_Z_X_C()
dim i as integer, j as integer, z1 as single
  PIVOT=A(U,E)
  for j=1 to N
    A(U,j)=A(U,j)/PIVOT
  next j
  X(U)=X(U)/PIVOT
  z1=Z-X(U)*C(E)
  if z1 <=Z then
    MESSAGGIO="** L'ALGORITMO NON CONVERGE **"
    ERROR_ROUTINE
    ERRORE=true
    exit sub
  else
    Z=Z1
  end if
  for i=1 to M
    if i<>U then X(i)=X(i)-A(i,E)*X(U)
  next i
  for j=1 to N
    if j<>E then C(j)=C(j)-A(U,j)*C(E)
  next j
  C(E)=-C(E)/PIVOT
end sub

```

```

sub TEST_DI_OTTIMALITA()
dim j as integer
  SOLUZIONE_OTTIMA=true
  j=1
  do
    if C(j)<0 then
      SOLUZIONE_OTTIMA=false
    else
      j=j+1
    end if
  loop until (j>N) or (not SOLUZIONE_OTTIMA)
end sub

```

```

sub RICALCOLA_MATRICE_A()
dim i as integer, j as integer
  for i=1 to M
    for j=1 to N
      if (i<>U) and (j<>E) then
        A(i,j)=A(i,j)-A(i,E)*A(U,j)
      end if
    next j
  next i
  for i=1 to M
    if i<>U then
      A(i,E)=-A(i,E)/PIVOT
    end if
  next i
  A(U,E)=1.0/PIVOT
end sub

```

```

sub VISUALIZZA_SOLUZIONE()
dim i as integer
Cells(25, 1) = MESSAGGIO
Cells(27, 1) = "Valore della funzione obiettivo Z="
Cells(27, 3) = Z
Cells(29, 1) = "Vettore soluzione: "
  for i=1 to M
    Cells(29+i, 1) = "X(" & IN_BASE(i) & ")="
    Cells(29+i, 2) = X(i)
  next i
end sub

```

$$\max z = 2x_1 + 5x_2$$

$$\begin{cases} x_1 \leq 400 \\ x_2 \leq 300 \\ x_1 + x_2 \leq 600 \end{cases}$$

Tableau iniziale:

	x_1	x_2	
x_3	1	0	400
x_4	0	1	300
x_5	1	1	600
	-2	-5	$z=0$

Soluzione parziale:

	x_1	x_2	
x_3	1	0	400
x_4	0	1	300
x_5	1	1	300
	-2	5	$z=1500$

Soluzione ottima:

	x_1	x_2	
x_3	1	0	100
x_4	0	1	300
x_5	1	-1	600
	2	3	$z=2100$

$$\begin{cases} x_1 + x_2 \leq 5 \\ 2x_2 - x_3 \leq 3 \\ x_i \geq 0 \end{cases}$$

$$\max z = x_1 + 2x_2 - 3x_3$$

	x_1	x_2	x_3	
x_4	1	1	0	5
x_5	0	②	-1	3
c_j	-1	-2	3	0

Soluzione parziale:

ricalcola z , x , c

	x_1	x_2	x_3	
x_4	1	1	0	3.5
x_5	0	1	-0.5	1.5
c_j	-1	-1	2	3

Test ottimalità negativo:

ricalcola matrice A

	x_1	x_2	x_3	
x_4	①	-0.5	0.5	3.5
x_5	0	0.5	-0.5	1.5
c_j	-1	1	2	3

Soluzione parziale:

ricalcola z , x , c

	x_1	x_2	x_3	
x_4	1	-0.5	0.5	3.5
x_5	0	0.5	-0.5	1.5
c_j	1	0.5	2.5	6.5

Soluzione ottima.