Maxima: esperienze di programmazione

Milla Lacchini
CRIAD Centro di Ricerca dell'Informatica Applicata alla Didattica
Università di Bologna sede di Cesena
mlacchini@racine.ra.it

Maxima **è un sistema di calcolo algebrico comprendente un linguaggio di programmazione semplice dal punto di vista della sintassi ma efficace, mediante il quale è possibile descrivere algoritmi per risolvere problemi da quelli classici (di ricerca, ordinamento, etc.) a quelli di analisi numerica. Grazie alla sintassi semplificata del linguaggio, le istruzioni possono essere apprese in tempi ridotti e manipolate senza particolari difficoltà: in questo modo l'attenzione rimane focalizzata sulla risoluzione del problema oggetto di analisi senza disperdersi negli aspetti di carattere formale, ed è possibile conseguire risultati significativi con ragionevole impegno.**

La programmazione consente di perseguire importanti obiettivi formativi, fra i quali:

- favorire lo sviluppo di capacità metacognitive;

- acquisire la capacità di lavorare in gruppo attraverso la realizzazione di progetti complessi, e collaborare tra pari nella correzione degli errori dei programmi;

- stimolare negli studenti spirito critico e curiosità intellettuale, permettendo loro di costruire in modo autonomo funzionalità che i vari ambienti rendono disponibili già predefinite.

E' fondamentale, ai fini di promuovere un apprendimento significativo, che la programmazione non venga trascurata, invertendo la tendenza ad abbandonarla che molti docenti hanno assunto, essenzialmente a causa della sproporzione tra lo sforzo prodigato e l'esiguità dei risultati conseguiti.

Vengono di seguito presentate alcune esperienze di programmazione, realizzate nell'ambito del triennio ad indirizzo Brocca scientifico tecnologico presso il Liceo scientifico "G. Ricci Curbastro" di Lugo(RA), che utilizzano moduli sviluppati mediante tale linguaggio, nell'intento di proporre *Maxima* ai docenti di Matematica come strumento per fare programmazione nell'ambito della disciplina "Matematica e Informatica".

Algoritmo delle divisioni successive per il calcolo del massimo comun divisore tra due interi a e b non nulli (classe terza)

Maxima consente di introdurre i primi elementi di programmazione implementando semplici algoritmi , ad esempio per il calcolo dell'MCD.
L'algoritmo di Euclide delle divisioni successive considera il resto della divisione intera di a per b.
Sia q il quoziente intero della divisione di a per b e sia $r = a - b\,q$ cioè $r = a \,\text{Modulo}\, b$
- Se $r = 0$ allora il M.C.D. è b
- altrimenti si sostituisce a con b , b con r e si ripete il procedimento.
Utilizziamo la funzione DIVIDE(a,b) che effettua la divisione di a per b e restituisce una lista in cui il primo elemento è il quoziente intero, il secondo è il resto; per estrarre il resto dalla lista si utilizza la funzione PART.

```
(D2) 2
```

Maxima supporta la programmazione ricorsiva:

```
(D4) 2
```

Modifichiamo l'algoritmo in modo che il resto venga calcolato una volta sola:

Grafica a raster-scan: algoritmo di Bresenham per la generazione di linee (classe quarta)

L'esperienza condotta nella classe quarta utilizza un approccio alla grafica con il calcolatore dal punto di vista algoritmico, finalizzato a stimolare gli studenti a "guardare dentro il sistema" per riuscire a costruire in modo autonomo le funzioni che gli ambienti grafici mettono a disposizione. Vediamo, ad esempio, come risolvere il problema di disegnare un segmento sul monitor di un calcolatore.

Lo schermo si configura come uno spazio DISCRETO costituito da un insieme finito di elementi detti PIXEL (acronimo di *Picture Element*). Il Pixel è l'elemento minimo che può essere indirizzato e quindi acceso sul monitor; a differenza del punto geometrico, ha dimensioni non nulle ed è simile ad un quadratino. La discretizzazione del monitor implica che un segmento contenga un numero finito di pixel, a differenza del segmento geometrico che, in quanto insieme continuo, è costituito da una infinità non numerabile di punti. Ogni pixel è indirizzato tramite una coppia ordinata di interi che rappresentano le coordinate rispetto ad un sistema di assi ortogonali, avente l'origine nell'angolo in alto a sinistra del monitor e l'asse delle ordinate orientato verso il basso . Disegnare un pixel sullo schermo significa accenderlo nel colore opportuno, che risulta dalla combinazione dei tre colori di base: Red, Green, Blue.

La risoluzione di un monitor è il numero di PIXEL nell'unità di misura; si misura in DPI (Dots Per Inch, ovvero numero di punti per pollice, essendo 1 pollice=2.54 cm): maggiore è questo numero, minore è la dimensione dei singoli pixel e quindi migliore risulterà la definizione delle immagini. La risoluzione complessiva si indica come il prodotto fra il numero di pixel per riga e il numero totale di righe (es. 1024x768).

Dal punto di vista delle caratteristiche tecnologiche, si distinguono i seguenti tipi di monitor a matrice di punti:

- **CRT** (Catod Ray Tube ovvero \"Tubo a raggi catodici\"): tre fasci di elettroni, emessi da tre cannoni di elettroni raggruppati a delta, colpiscono i fosfori dei colori di base che ricoprono il monitor , convergendo (ad opera di un sistema di deflessione e messa fuoco del CRT) in un punto che viene illuminato. Un pixel è quindi formato da tre fosfori corrispondenti ai colori fondamentali; una lamina di metallo posta dietro lo strato di fosforo del CRT (shadow mask), con un foro in corrispondenza ad ogni triade RGB di punti, fa sì che ogni punto della triade venga colpito solo dal fascio di elettroni che proviene dal corrispondente emittore, e rimanga in ombra rispetto ai raggi che provengono dagli emittitori relativi agli altri due colori .

- **LCD** (Liquid Crystal Display ossia a cristalli liquidi). Determinate sostanze con struttura cristallina tipica di un solido, se opportunamente riscaldate, assumono una consistenza semiliquida, pur mantenendo una struttura cristallina al proprio interno. I cristalli liquidi si configurano come filamenti (per questo sono detti \"*nematici*\" dal greco \"*nemo*\" ossia \"filo\") all'interno di questa sostanza semifluida; possiedono pertanto una certa libertà di movimento, ed inoltre rifrangono i fasci di luce su di essi incidenti, secondo una modalità dipendente dal loro orientamento. Tale orientamento può essere modificato sottoponendo i filamenti ad un apposito campo elettrico.Un pixel corrisponde ad una cella di cristalli liquidi, alla quale viene applicato un potenziale elettrico; il voltaggio determina l'intensità e il colore del punto illuminato. Facendo passare il fascio di luce attraverso tre filtri che riproducono i colori fondamentali, e combinando questi tre colori sulla base del potenziale applicato, è possibile accendere il pixel opportuno nel colore desiderato. Questa tecnologia assicura una buona definizione, consumi ridotti, basse emissioni nocive e l'assenza di sfarfallio dell'immagine, presente invece nei monitor CRT e causato dal refresh per mantenere la fluorescenza del fosforo.

- a **plasma**: due pannelli di vetro, che racchiudono tra loro dei gas, sottoposti ad un campo elettrico generano raggi ultravioletti che stimolano i fosfori dei colori di base. Questa tecnologia è attualmente utilizzata per grandi pannelli.

Le funzioni base offerte da un qualunque dispositivo di grafica al calcolatore sono: disegnare un punto, tracciare un segmento. Proponiamo di seguito l'implementazione dell'algoritmo di Bresenham con numeri interi per la generazione di linee [7].

Per verificare quali Pixel vengono accessi a seconda dell'inclinazione del segmento, eseguire:

bresenham(1,1,13,1) , bresenham(1,1,9,19), bresenham(1,1,12,4), bresenham(1,1,12,2). Ogni chiamata del modulo *bresenham* deve essere seguita dalle istruzioni (C3) e (C4).

In [7] è descritto anche l'algoritmo per generare circonferenze, di facile implementazione.

Per la classe quinta vengono presentate delle esperienze inerenti alla implementazione di algoritmi di analisi numerica: zeri di una funzione, interpolazione statistica e matematica.

Zeri di una funzione (classe quinta)

Metodo delle approssimazioni successive

Discende dal teorema del punto fisso di seguito enunciato.

Sia [a,b] un intervallo limitato e chiuso e sia $\Phi(x)$ una funzione definita su [a,b] che verifica le seguenti condizioni:

1. $\Phi(x)$ è continua su [a,b];
2. $\Phi(x)$ muta l'intervallo [a,b] in se', cioè $\Phi(x) \in [a,b] \quad \forall x \in [a,b]$
3. $\Phi(x)$ è derivabile in [a,b] e per una opportuna costante $\lambda < 1$ risulta $\left|\Phi'(x)\right| < \lambda$

Allora per ogni scelta di $x_0 \in [a,b]$ la successione definita da $x_n = \Phi(x_{n-1})$ converge all'unico punto fisso ξ della funzione $\Phi(x)$, cioè all'unica soluzione dell'equazione $\xi = \Phi(\xi)$.

Consideriamo la funzione

$$\Phi(x) = e^{-x}$$

Su di un intervallo $[a,b]$ con $a > 0$ la funzione verifica le condizioni del teorema.
Il punto fisso della funzione è l'intersezione con la retta $y = x$; attivando il comando ZOOM e cliccando col mouse in prossimità del punto è possibile dedurne le coordinate.
Definiamo un sottoprogramma avente come parametri di ingresso: la funzione, la variabile x da considerare, l'iterato iniziale x_0 , la precisione eps a meno della quale determinare l'approssimazione della soluzione, il numero massimo di iterazioni da eseguire che assicura lam, terminazione del ciclo iterativo anche nel caso in cui il metodo non converga. Vengono utilizzate le variabili locali k, x_n , xprec; L'istruzione RETURN determina l'uscita dal sottoprogramma.

(D3) $\%E^{-x}$

xn set to 0.0
xn set to 1.0
xn set to 0.36787944117144
xn set to 0.69220062755535
xn set to 0.50047350056364
xn set to 0.6062435350856
xn set to 0.54539578597503
xn set to 0.57961233550338
xn set to 0.56011546136109
xn set to 0.57114311508018
xn set to 0.56487934739105

(D5) 0.56487934739105

Il valore di xn ad ogni iterazione viene visualizzato due volte perchè compare in due assegnazioni

(D0) DONE

Modifichiamo il sottoprogramma affinchè stampi anche il numero di iterazioni eseguite:

(D2) $\%E^{-x}$

xn set to 0.0
xn set to 1.0
xn set to 0.36787944117144
xn set to 0.69220062755535
xn set to 0.50047350056364
xn set to 0.6062435350856
xn set to 0.54539578597503
xn set to 0.57961233550338
xn set to 0.56011546136109
xn set to 0.57114311508018
xn set to 0.56487934739105
xn= 0.56487934739105 in 14 iterazioni

(D4) fine

Metodo di bisezione

(D3) $x^2 - 78.8$

c = 8.876936408468858 in 101 iterazioni
(D4) fine

Interpolazione (classe quinta)

Dati $n+1$ punti (x_i, y_i) con ascisse tutte distinte, il problema dell'interpolazione matematica può presentarsi nelle seguenti situazioni:

1. I valori y_i sono ricavati sperimentalmente nell'ambito dello studio di un certo fenomeno naturale del quale si vuole costruire un modello matematico, cioè una relazione del tipo $y = f(x)$ tra le variabile indipendente x e la variabile dipendente y

2. Di una certa funzione F si conoscono soltanto i valori assunti in corrispondenza alle ascisse x_i ; pertanto si vuole costruire una funzione f che approssimi F, per poterla valutare in corrispondenza ad un qualunque valore $\bar{x} \neq x_i$, $\bar{x} \in [a, b]$ essendo $[a, b]$ un intervallo che contiene tutte le ascisse x_i

3. Di una certa funzione F si conosce l'espressione analitica che tuttavia risulta molto complessa; si rende pertanto necessario sostituire F con una funzione facilmente calcolabile, ad esempio per operazioni di integrazione e derivazione.

Interpolare gli $n+1$ punti significa determinare una funzione $y = f(x)$ passante per i punti assegnati, cioè tale che $f(x_i) = y_i \quad i = 0, 1, ..., n$

Si può assumere in molti casi che la funzione *y=f(x)* abbia l'espressione

$$y = a_0\phi_0(x) + a_1\phi_1(x) + ... + a_n\phi_n(x)$$

dove $\phi_i(x)$ sono funzioni elementari quali ad esempio i monomi 1, x, ..., x^n : in tal caso la funzione interpolatrice sarà un polinomio di grado al piu' n

$$p_n(x) = a_0 + a_1x + ... + a_nx^n$$

che può essere valutato, differenziato o integrato facilmente in un numero finito di intervalli, usando solo le operazioni aritmetiche fondamentali di addizione, sottrazione e moltiplicazione.

Nell'ipotesi in cui gli n+1 punti abbiano ascisse tutte distinte, esiste un solo polinomio che li interpola; tale polinomio può essere determinato:

- risolvendo il sistema lineare normale $p_n(x_i) = y_i \quad i = 0, 1, ..., n$ nelle n+1 incognite $a_0, a_1, ..., a_n$, ad esempio mediante il metodo di Gauss. La matrice dei coefficienti è una matrice di Vandermonde: essendo le ascisse x_i tutte distinte, è non singolare quindi il sistema ammette una sola soluzione, tuttavia questa matrice risulta mal condizionata, quindi il problema di risolvere tale sistema è in generale mal posto;

- mediante la formula di Lagrange;

- mediante la formula di Newton, nelle situazioni 2 e 3.

Aumentando il numero dei punti base, aumenta anche il grado del polinomio, ma in generale non diminuisce l'errore di interpolazione, come si può dedurre dagli esempi di seguito proposti.

Risoluzione di un sistema lineare normale con matrice dei coefficienti non singolare applicando il metodo di eliminazione di Gauss

Ricordiamo che un sistema di equazioni si definisce *lineare* se costituito di equazioni lineari, cioè le incognite che compaiono in esse hanno grado al piu' uguale a 1, *normale* se il numero delle equazioni è uguale al numero delle incognite, *compatibile* o *consistente* se ammette almeno una soluzione.
Sia assegnato il sistema lineare normale di ordine n $A\mathbf{x} = \mathbf{b}$ dove $A = [a]$ è la matrice dei coefficienti, $\mathbf{b}=(b_1, ..., b_n)$ il vettore dei termini noti e $\mathbf{x}=(x_1, ..., x_n)$ il vettore soluzione. Se la matrice A dei coefficienti è non singolare (cioè ammette matrice inversa A^{-1}), il metodo di eliminazione di Gauss consente di risolvere il sistema con un costo computazionale dell'ordine di $\frac{n^3}{3}$. Applicando alla matrice A e al vettore $\mathbf{b}$ le trasformazioni elementari di Gauss, il sistema viene trasformato in un sistema triangolare superiore equivalente che può essere risolto mediante sostituzione all'indietro.
Consideriamo il sistema

$$\begin{cases} x_1 + 2x_2 + 3x_3 = 2 \\ -x_1 - x_2 - x_3 = -1 \\ x_1 + 4x_2 + 8x_3 = 5 \end{cases}$$

Poniamo

$$A^{(1)} = A = \begin{pmatrix} 1 & 2 & 3 \\ -1 & -1 & -1 \\ 1 & 4 & 8 \end{pmatrix} \qquad b^{(1)} = b = \begin{pmatrix} 2 \\ -1 \\ 5 \end{pmatrix}$$

Determiniamo ora nella prima colonna un elemento $a_{r1}^{(1)} \neq 0$, quindi scambiamo la riga 1 con la riga r sia nella matrice $A^{(1)}$ sia nel vettore $b^{(1)}$ (nell'esempio risulta $a_{11}^{(1)} \neq 0$ pertanto non effettuiamo alcuno scambio); se non esiste alcun indice r tale che $a_{r1}^{(1)} \neq 0$, la matrice $A^{(1)}$ è singolare e il metodo si arresta.
Si definiscono *elemento pivot* l'elemento $a_{11}^{(1)}$ risultante e *pivoting per colonna* il procedimento di ricerca applicato.
Consideriamo la trasformazione elementare di Gauss di indice 1:

$$L^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{a_{21}^{(1)}}{a_{(1)11}} & 1 & 0 \\ -\frac{a_{31}^{(1)}}{a_{11}^{(1)}} & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}$$

Applichiamo $L^{(1)}$ alla matrice dei coefficienti e al vettore dei termini noti (mediante prodotto righe per colonne) ottenendo i corrispondenti trasformati:

$$A^{(2)} = L^{(1)} \cdot A^{(1)} = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & 2 & 5 \end{pmatrix} \qquad b^{(2)} = L^{(1)} \cdot b^{(1)} = \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix}$$

La trasformazione $L^{(1)}$ elimina x_1 dalla seconda e dalla terza equazione: alla seconda equazione sostituisce la somma della seconda equazione con la prima moltiplicata per $-\frac{a_{21}^{(1)}}{a_{(1)11}}$, alla terza equazione sostituisce la somma della terza equazione con la prima moltiplicata per $-\frac{a_{31}^{(1)}}{a_{11}^{(1)}}$.

$$L^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{a_{32}^{(2)}}{a_{22}^{(2)}} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{pmatrix}$$

$$A^{(3)} = L^{(2)} \cdot A^{(2)} = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix} \qquad b^{(2)} = L^{(1)} \cdot b^{(1)} = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}$$

La trasformazione $L^{(2)}$, applicata alla matrice $A^{(2)}$, elimina x_2 dalla terza equazione, sostituendo ad essa la somma della terza equazione con la seconda moltiplicata per $\frac{a_{32}^{(2)}}{a_{22}^{(2)}}$.

Il sistema dato risulta così trasformato nel sistema triangolare superiore equivalente:

$$\begin{cases} x_1 + 2x_2 + 3x_3 = 2 \\ x_2 + 2x_3 = 1 \\ x_3 = 1 \end{cases}$$

che si risolve mediante sostituzione all'indietro a partire dall'ultima equazione:

$$\begin{cases} x_3 = 1/1 = 1 \\ x_2 = (1 - 2 \cdot 1)/1 = -1 \\ x_3 = (2 - 2 \cdot (-1) - 1 \cdot 1)/3 = 1 \end{cases}$$

Osserviamo che:

$L^{(2)} \cdot L^{(1)} \cdot A = A^{(2)}$

da cui discende la seguente fattorizzazione della matrice A:

$A = (L^{(2)} \cdot L^{(1)})^{-1} \cdot A^{(2)} = (L^{(1)})^{-1} \cdot (L^{(2)})^{-1} \cdot A^{(2)} = L \cdot R$

dove $L = (L^{(1)})^{-1} \cdot (L^{(2)})^{-1}$ è una matrice triangolare inferiore con elementi sulla diagonale principale tutti uguali a 1, e $R = A^{(2)}$ è una matrice triangolare superiore con elementi sulla diagonale principale tutti diversi da 0.

Per un sistema lineare normale di ordine n, al passo k $1 \leqslant k \leqslant n-1$ si applica la trasformazione elementare di Gauss

$$L^{(k)} = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & -\frac{a^{(k)}_{k+1,k}}{a^{(k)}_{k,k}} & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & & \dots & \dots \\ 0 & 0 & \dots & -\frac{a^{(k)}_{n,k}}{a^{(k)}_{k,k}} & 0 & \dots & 1 \end{pmatrix}$$

La matrice $L^{(k)}$ è triangolare inferiore con elementi sulla diagonale principale tutti diversi da 0, quindi è non singolare cioè ammette matrice inversa

$$(L^{(k)})^{-1} = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & \frac{a^{(k)}_{k+1,k}}{a^{(k)}_{k,k}} & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & & \dots & \dots \\ 0 & 0 & \dots & \frac{a^{(k)}_{n,k}}{a^{(k)}_{k,k}} & 0 & \dots & 1 \end{pmatrix}$$

Per ridurre gli errori di troncamento, in generale si sceglie come elemento pivot il massimo, in valore assoluto, fra gli elementi della colonna presi in esame ossia $a^{(k)}_{r,k} = \max_{i \geqslant k} |a^{(k)}_{i,k}|$; questo criterio di scelta tuttavia garantisce la stabilità numerica del metodo di Gauss solo in particolari condizioni [7].

Applichiamo ora il metodo al sistema iniziale utilizzando *Maxima*. Per comodità consideriamo la matrice completa del sistema, ottenuta giustapponendo alla matrice dei coefficienti il vettore dei termini noti:

$$\mathrm{Ac} = \left(\begin{array}{ccc|c} 1 & 2 & 3 & 2 \\ -1 & -1 & -1 & -1 \\ 1 & 4 & 8 & 5 \end{array}\right)$$

Il caricamento della matrice può avvenire:

- a partire dalle equazioni del sistema mediante la funzione AUGCOEFMATRIX, che restituisce la matrice completa del sistema, coi termini noti trasportati al primo membro e quindi nell'ultima colonna con segno cambiato:
- mediante il comando ENTERMATRIX che chiede in input i singoli elementi.

Definiamo la lista con le equazioni del sistema:

(D1) $[3\,\mathrm{x3} + 2\,\mathrm{x2} + \mathrm{x1} = 2, -\mathrm{x3} - \mathrm{x2} - \mathrm{x1} = -1, 8\,\mathrm{x3} + 4\,\mathrm{x2} + \mathrm{x1} = 5]$

Generiamo la matrice completa del sistema:

(D2) $$\begin{pmatrix} 1 & 2 & 3 & -2 \\ -1 & -1 & -1 & 1 \\ 1 & 4 & 8 & -5 \end{pmatrix}$$

Mediante la funzione TRIANGULARIZE applichiamo le trasformazioni elementari di Gauss alla matrice dei coefficienti e al vettore dei termini noti, ottenendo la matrice $A^{(2)}$ e il vettore $-b^{(2)}$ già descritti in precedenza:

(D3) $$\begin{pmatrix} 1 & 2 & 3 & -2 \\ 0 & 1 & 2 & -1 \\ 0 & 0 & 1 & -1 \end{pmatrix}$$

Implementiamo il sottoprogramma che risolve il sistema triangolare superiore:

$x1 = 1$
$x2 = -1$
$x3 = 1$
(D7) stampa soluzione effettuata

(D8) $[2\,\mathrm{x3} + \mathrm{x2} + \mathrm{x1} = 2, \mathrm{x3} - \mathrm{x2} + 2\,\mathrm{x1} = 1, 3\,\mathrm{x3} + 2\,\mathrm{x2} + \mathrm{x1} = 2]$

(D9) $$\begin{pmatrix} 1 & 1 & 2 & -2 \\ 2 & -1 & 1 & -1 \\ 1 & 2 & 3 & -2 \end{pmatrix}$$

(D10) $$\begin{pmatrix} 1 & 1 & 2 & -2 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 3 \end{pmatrix}$$

Si osservi che Maxima sceglie il pivot $a_{r2}^{(2)}$ nella terza riga, cioè privilegia la riga contenente il maggior numero di elementi nulli per ridurre le moltiplicazioni.
In questo esempio la matrice dei coefficienti è singolare e il sistema non compatibile

coefficiente di $\mathrm{x3} = 0$
(D11) matrice dei coefficienti singolare

(D12) $[\mathrm{x3} + \mathrm{x2} + \mathrm{x1} = 0, -\mathrm{x3} + \mathrm{x2} + 2\,\mathrm{x1} = -3, 2\,\mathrm{x1} - 4\,\mathrm{x3} = -6]$

(D13) $\begin{pmatrix} 1 & 1 & 1 & 0 \\ 2 & 1 & -1 & 3 \\ 2 & 0 & -4 & 6 \end{pmatrix}$

(D14) $\begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & -1 & -3 & 3 \\ 0 & 0 & 0 & 0 \end{pmatrix}$

In questo caso la matrice R è singolare, come l'algoritmo segnala, ma il sistema è compatibile

coefficiente di x3 = 0
(D15) matrice dei coefficienti singolare

Verifichiamo ora le proprietà delle trasformazioni $L^{(1)}$ ed $L^{(2)}$, in quanto matrici triangolari inferiori.

```
Is the matrix  1. Diagonal  2. Symmetric  3. Antisymmetric  4. General
```

```
Matrix entered.
```

(D1) $\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}$

Verifichiamo ora che $(L^{(1)})^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ \frac{a_{21}^{(1)}}{a_{(1)_{11}}} & 1 & 0 \\ \frac{a_{31}^{(1)}}{a_{11}^{(1)}} & 0 & 1 \end{pmatrix}$

(D2) $\begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$

```
Is the matrix  1. Diagonal  2. Symmetric  3. Antisymmetric  4. General
```

```
Matrix entered.
```
(D3) $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{pmatrix}$

(D4) $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{pmatrix}$

Verifichiamo infine che $(L^{(2)} \cdot L^{(1)})^{-1} = (L^{(1)})^{-1} \cdot (L^{(2)})^{-1} = L$, essendo L ancora una matrice triangolare inferiore con elementi sulla diagonale principale tutti uguali a 1:

(D5) $\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -1 & -2 & 1 \end{pmatrix}$

(D6) $\begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 2 & 1 \end{pmatrix}$

(D7) $\begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 2 & 1 \end{pmatrix}$

Alcuni esempi di sistemi la cui risoluzione costituisce un problema mal posto
Risolviamo il sistema:
$\begin{cases} x + 1.001\, y = 2.000 \\ x + 1.001\, y = 2.001 \end{cases}$

(D1) $[[x = 2, y = 0]]$

Perturbando lievemente il termine noto della prima equazione sulla terza cifra decimale e risolvendo il nuovo sistema, si ottiene una soluzione completamente diversa dalla precedente:

(D2) $[[x=1, y=1]]$

Questa lieve variazione del termine noto ha prodotto una rilevante modifica nella soluzione.
Il problema di risolvere un sistema tale che una piccola variazione dei coefficienti e/o dei termini noti determini una rilevente modifica della soluzione, si dice *mal posto* e la matrice dei coefficienti *mal condizionata*.
Se un problema è mal posto, l'errore *inerente*, dovuto alla rappresentazione dei dati iniziali del problema, prevale sull'errore *algoritmico*, cioè sugli errori di arrotondamento determinati dalle operazioni di calcolo eseguite: la bontà del risultato è pregiudicata a prescindere dal procedimento risolutivo utilizzato.
La sensibilità della soluzione del sistema rispetto a variazioni della matrice A dei coefficienti e del vettore b dei termini noti può essere misurata dall'*indice di condizionamento* di A, espresso in funzione della norma della matrice e della sua inversa:
$\mathrm{cond}(A) = \|A\| \cdot \|A^{-1}\|$
Se questo numero è piccolo allora il problema di risolvere il sistema $A\mathbf{x} = \mathbf{b}$ risulta ben posto e la matrice A ben condizionata, se è grande invece il problema può essere mal posto.
Classici esempi di matrici mal condizionate sono rappresentati dalla matrice di Vandermonde e dalla matrice di Hilbert.
Nel problema dell'interpolazione polinomiale, la matrice dei coefficienti è una matrice di Vandermonde. Vediamo un esempio con n=3, cioè supponendo di voler interpolare 4 punti: imponendo il passaggio del generico polinomio di grado 3 per i 4 punti, otteniamo un sistema di 4 equazioni nelle 4 incognite a0, a1, a2, a3 che rappresentano i coefficienti del polinomio.

(D1) $\mathrm{A3\,x0^3 + A2\,x0^2 + A1\,x0 + a0 = y0}$

(D2) $\mathrm{A3\,x1^3 + A2\,x1^2 + A1\,x1 + a0 = Y1}$

(D3) $\mathrm{A3\,x2^3 + A2\,x2^2 + A1\,x2 + a0 = Y2}$

(D4) $\mathrm{A3\,x3^3 + A2\,x3^2 + A1\,x3 + a0 = Y3}$

Determiniamo la matrice dei coefficienti, verificando che è effettivamente una matrice di Vandermonde:

(D5) $\begin{pmatrix} 1 & \mathrm{x0} & \mathrm{x0^2} & \mathrm{x0^3} \\ 1 & \mathrm{x1} & \mathrm{x1^2} & \mathrm{x1^3} \\ 1 & \mathrm{x2} & \mathrm{x2^2} & \mathrm{x2^3} \\ 1 & \mathrm{x3} & \mathrm{x3^2} & \mathrm{x3^3} \end{pmatrix}$

Valutiamo gli elementi della matrice attribuendo dei valori alle ascisse dei punti base:

(D1) v4

Determiniamo la matrice inversa:

(D7)
$$\begin{pmatrix} 14.44444444444449 & -19.84126984126992 & 9.285714285714331 & -2.888888888888899 \\ -24.11111111111121 & 37.69841269841285 & -20.14285714285723 & 6.55555555555558 \\ 12.88888888888894 & -21.82539682539691 & 13.71428571428577 & -4.777777777777798 \\ -2.22222222222223 & 3.968253968253984 & -2.857142857142867 & 1.111111111111117 \end{pmatrix}$$

Implementiamo l'algoritmo per il calcolo dell'indice di condizionamento della matrice di Vandermonde dell'esempio

Calcoliamo l'indice di condizionamento:

(D9) 2245.888888888898

All'aumentare dell'ordine della matrice, l'indice di condizionamento cresce rapidamente a causa della crescita, in valore assuluto, degli elementi della matrice inversa.

Studio del polinomio interpolatore della funzione di Runge con utilizzo della formula di Lagrange

Esprimiamo il polinomio interpolatore mediante la formula di Lagrange:

$p_n(x) = \sum_{i=0}^{n} y_i L_i(x)$

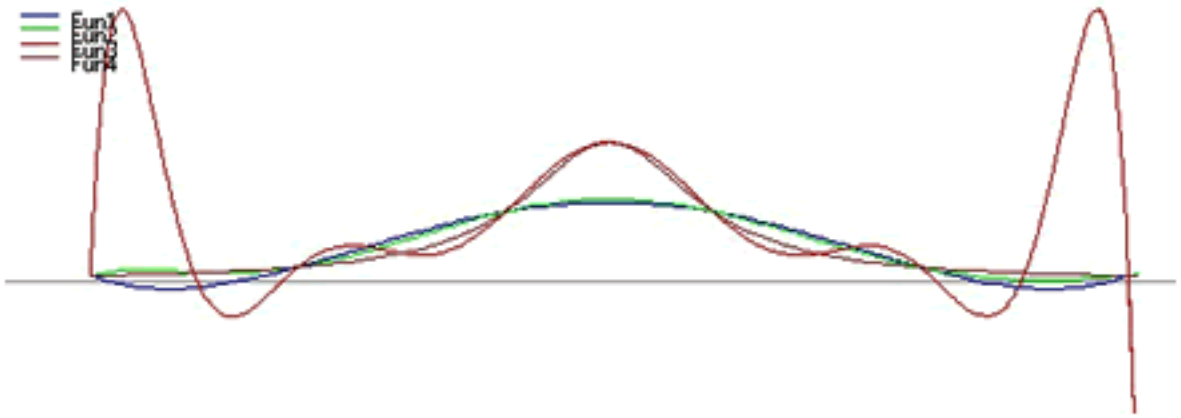
I polinomi $L_i(x)$ soddisfano le condizioni $L_i(x_j) = \begin{cases} 1 & \text{se } i = j \\ 0 & \text{se } i \neq j \end{cases}$

pertanto sono espressi dalla formula $L_i(x) = \frac{(x-x_0)(x-x_1)...(x-x_{i-1})(x-x_{i+1})...(x-x_n)}{(x_i-x_0)(x_i-x_1)...(x_i-x_{i-1})(x_i-x_{i+1})...(x_i-x_n)}$ $i = 0, 1, ..., n$

Studiamo ora come varia l'errore di interpolazione

$R(x) = p_n(x) - F(x)$

per la funzione di Runge $F(x) = \frac{1}{1+25x^2}$, al variare del numero dei punti base e quindi del grado del polinomio interpolatore, considerando ascisse equidistanti sull'intervallo [-1,1]



Studio del polinomio interpolatore della funzione di Runge con utilizzo della formula di Newton delle differenze divise

Esprimiamo il polinomio interpolatore mediante la formula di Newton delle differenze divise:

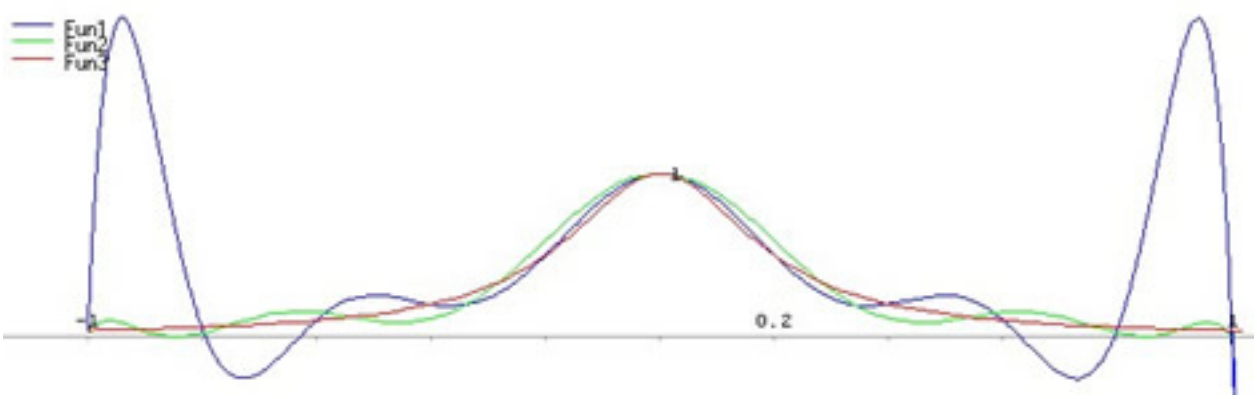
$p_n(x) = F(x_0) + (x - x_0)F[x_0, x_1] + (x - x_0)(x - x_1)F[x_0, x_1, x_2] + ... + (x - x_0)...(x - x_{n-1})F[x_0,..., x_n]$

essendo le differenze divise definite dalle formule ricorrenti:

$$\begin{cases} F[x_0] = F(x_0) \\ F[x_0, x_1] = \frac{F(x_0) - F(x_1)}{x_0 - x_1} \\ F[x_0,..., x_n] = \frac{F[x_0,..., x_{n-1}] - F[x_1,..., x_n]}{x_0 - x_n} \end{cases}$$

Mettiamo ora a confronto il polinomio di interpolazione di grado $n = 10$ con ascisse

$x_i = -1 + i\,h \;\; i = 1, 2, ..., n$ equidistanti di passo $h = 2/n$, con il polinomio di interpolazione di grado $n = 10$ avente come ascisse gli $n + 1$ zeri del polinomio di Chebyshev di grado $n + 1 = 11$



Dal grafico si può concludere che l'errore di interpolazione risulta minore se i punti base hanno come ascisse gli zeri del polinomio di Chebyshev.

Approssimazione: metodo dei minimi quadrati (classe quinta)

Data una distribuzione di punti (x_i, y_i) $\quad i = 1, 2, ..., n$ con andamento pressochè lineare, si vuole determinare la retta $\bar{y} = mx + q$ che approssima nel miglior modo possibile tali punti secondo il metodo dei minimi quadrati,, cioè la retta per la quale risulta minima la somma dei quadrati degli scarti

$\sum_i \varepsilon_i^2 \qquad \varepsilon_i = y_i - \bar{y_i} \quad i = 1, 2, ..., n$

Si può verificare che tale retta ha equazione $y = y_b + \frac{\sum_i (x_i - x_b)(y_i - y_b)}{\sum_i (x_i - x_b)^2}(x - x_b)$

dove $x_b = \frac{\sum_i x_i}{n}$ e $y_b = \frac{\sum_i y_i}{n}$ sono le coordinate del *baricentro* della distribuzione.

Per il coefficiente angolare m, attraverso opportuni passaggi si perviene alla relazione

$$m = \frac{\sum_i (x_i - x_b)(y_i - y_b)}{\sum_i (x_i - x_b)^2} = \frac{\sum_i x_i y_i - \frac{1}{n}(\sum_i x_i)(\sum_i y_i)}{\sum_i x_i^2 - \frac{1}{n}(\sum_i x_i)^2}$$

Costruiamo ora l'algoritmo che, data in input la distribuzione di punti di coordinate (1,-5); (2,-12.4); (3,-15.7); (4,-15.1); (4,-10.5); (5,-1.9); (6,10.7); (7,10.7); (8,27.4) , disegna in un unico grafico i punti e la retta dei minimi quadrati che li approssima.

Ciclo che inizializza il vettore delle ascisse e calcola le varie somme

Calcolo coordinate del baricentro della distribuzione:

Calcolo del coefficiente angolare della retta:

Per disegnare il grafico, costruiamo due liste: una con le coordinate dei punti della distribuzione, l'atra con le coordinate di un insieme di punti della retta

Le due liste di punti vengono disegnate su grafico mediante il comando XGRAPH_CURVES, che utilizza il modulo GRAPH. Questo programma di grafica consente di produrre anche animazioni; l'eseguibile può essere scaricato da http://jean-luc.ncsa.uiuc.edu/Codes/xgraph ; non è disponibile per Windows; in ambiente Linux basta copiare l'eseguibile, contenuto nel file *xgraph*, in /usr/X11R6/bin.

Bibliografia

1) G.Casadei, M.Lacchini, *Il Problem Solving algoritmico*, Atti didamatica 2004, Omniacom, Ferrara, 2004

2) D.Bini, M.Capovani, O.Menchi, *Metodi numerici per l'algebra lineare* , Zanichelli, Bologna, 1988

3) S. Harrington, *Computer Graphics*, McGraw Hill, 1983

4) F.Fontanella, A.Pasquali, *Calcolo numerico* , Pitagora, Bologna, 1984

5) I.Galligani, *Elementi di analisi numerica*, Zanichelli, Bologna, 1987

6) A.Paoluzzi, *Informatica grafica*, La Nuova Italia Scientifica, 1988

7) D.F. Rogers, *Procedural Elements for Computer Graphics*, McGraw Hill, USA, 1985, ed.italiana Tecniche Nuove, Milano, 1988

8) J. Stoer, *Einfuhrung in die Numerische Mathematik*, Springer Verlag, Berlin-Heidelberg-New York,1972, ed. italiana Zanichelli, Bologna, 1974