

Grafica a raster-scan: algoritmo di Bresenham per la generazione di linee

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 \(classe terza\)](#)



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

[Zeri di una funzione \(classe quinta\)](#)

[Interpolazione matematica \(classe quinta\)](#)

[Interpolazione statistica \(classe quinta\)](#)

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].

`bresenham(x1,y1,x2,y2):=block([x,y,dy,s1,s2,app,scambio,errore,i], x:x1, y:y1,`



```
(C1) dx:abs(x2-x1), dy:abs(y2-y1), s1:segno(x2-x1), s2:segno(y2-y1), if dy<dx then
(app:dx, dx:dy, dy:app, scambio:1) else scambio:0, errore:2*dy-dx, for i:1 step 1 thru
dx do ( vx[i]:x, vy[i]:y, do (if errore<0 then return(done) else (if scambio=1 then
x:x+s1 else y:y+s2, errore:errore-2*dx ) ), if scambio=1 then y:y+s2 else x:x+s1,
errore:errore+2*dy ), return(done) )$
```

```
(C2) bresenham(1,1,1,13);
```

```
(C3) p`retta:append(["NoLines: true", "LargePixels: true"], create`list([vx[i],vy[i]],i,1,dx));
```

```
(C4) xgraph`curves([p`retta]);
```

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.

Bibliografia

- 1) G.Casadei, M.Lacchini, *Il Problem Solving algoritmico*, Atti didamatica 2004, Omnicom, 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, *Einführung in die Numerische Mathematik*, Springer Verlag, Berlin-Heidelberg-New York,1972, ed. italiana Zanichelli, Bologna, 1974

