

## Grafico della parabola

Il **grafico matematico** è un disegno che mostra l'andamento di una funzione  $f(x)$  al variare della variabile  $x$ . Per tracciare un grafico sul video del computer, si deve scegliere un intervallo di valori entro cui far variare la  $x$ . Per ogni valore di  $x$  si calcola il valore di  $f(x)$  e si disegna il punto corrispondente. Per evitare che il disegno sia composto da punti, invece di calcolare un solo punto, se ne calcolano due adiacenti e si collegano tracciando un piccolo segmento che li unisce. La funzione risultante è data dall'unione di questi segmenti.

La difficoltà principale nella rappresentazione dei grafici è la presenza di due diversi sistemi di riferimento:

- il sistema di riferimento cartesiano della funzione
- il sistema di riferimento dello schermo.

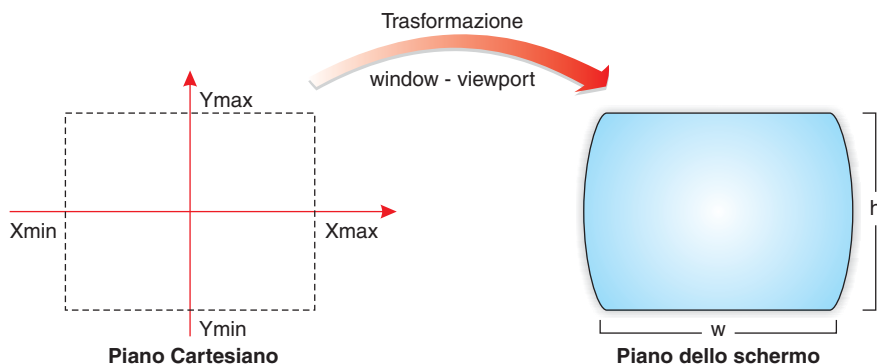
La funzione è solitamente pensata in un sistema di riferimento cartesiano, mentre deve essere disegnata sullo schermo usando il sistema di riferimento dello schermo.

Il sistema di riferimento cartesiano è illimitato e possiede una sua unità di misura. Il sistema di riferimento dello schermo è limitato nelle dimensioni e utilizza come unità di misura il pixel. Per visualizzare un grafico matematico sullo schermo, si deve fare una conversione tra i due sistemi di riferimento. Poiché non è possibile rappresentare l'intero piano cartesiano, bisogna definire una finestra all'interno del piano e mostrare sullo schermo solo il contenuto di questa finestra.

La finestra nel sistema di riferimento cartesiano, chiamata anche **window**, può essere identificata con quattro valori: XMAX, YMAX, XMIN, YMIN. Questi valori definiscono un rettangolo di tutti i punti  $x$  e  $y$  tali che  $XMIN < x < XMAX$  e  $YMIN < y < YMAX$ .

La finestra nel sistema di riferimento dello schermo, chiamata anche **viewport**, è caratterizzata da una larghezza e un'altezza misurate in pixel. Con la lettera  $w$  si indica la larghezza, mentre con la lettera  $h$  si indica l'altezza.

La trasformazione tra i due sistemi di riferimento viene chiamata **trasformazione di window-viewport**.



Il primo passo della trasformazione consiste nel calcolare a quanti pixel corrisponde l'unità di misura del piano cartesiano.

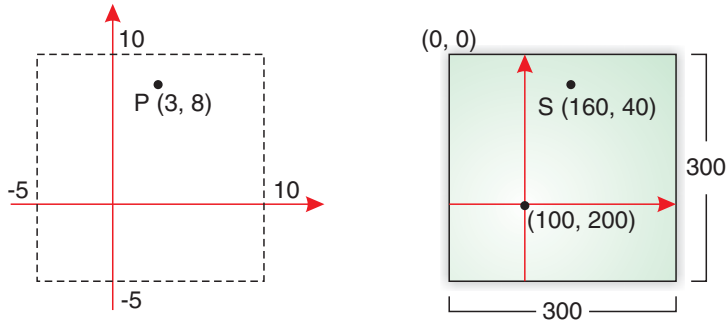
$$UNITA = w / (XMAX - XMIN)$$

In UNITA sono indicati i pixel corrispondenti ad un'unità nel sistema di riferimento cartesiano. Si può ipotizzare che anche per l'asse  $y$  venga usata la stessa unità di misura, in caso contrario occorre calcolare due diverse unità di misura.

Con un secondo passo si stabilisce come mappare il punto del piano cartesiano  $(px, py)$  nelle coordinate dello schermo. Il punto dello schermo  $(sx, sy)$  si ottiene con le seguenti formule:

$$sx = (px - XMIN) * UNITA;$$

$$sy = (YMAX - py) * UNITA;$$



Per esempio, supponiamo di aver definito  $XMAX=10$ ,  $YMAX=10$ ,  $XMIN=-5$ ,  $YMIN=-5$  e che le dimensioni dell'area sullo schermo siano  $w=300$  e  $h=300$ .

Il valore di  $UNITA$  è uguale a  $300 / (10 - (-5)) = 20$ .

Il punto  $P$  sul piano cartesiano che ha coordinate  $(3, 8)$  viene trasformato nel punto  $S$  nel seguente modo:

$$sx = (3 - (-5)) * 20 = 160$$

$$sy = (10 - 8) * 20 = 40$$

L'origine nel sistema di riferimento cartesiano ha coordinate  $(0,0)$ : sullo schermo si trova in posizione  $(100, 200)$ .

Dopo aver definito la trasformazione, il disegno di un grafico procede con questi passi:

- calcolare un punto  $P$
- trasformare il punto  $P$  nel punto  $S$
- disegnare il punto  $S$ .

Il calcolo del punto  $P$  cambia a seconda della funzione considerata. La variabile  $x$  varia tra i valori  $XMIN$  e  $XMAX$ , incrementandosi ad ogni passo di un valore abbastanza piccolo; per ogni  $x$ , si calcola il valore di  $y = f(x)$ , ottenendo così il punto  $P(x,y)$  che deve essere rappresentato. Il procedimento può essere schematizzato con il seguente algoritmo:

```

inizio
  incremento = 0.01
  x = XMIN
  esegui mentre x < XMAX
    y = f(x)
    S = trasforma(x, y)
    disegna(S)
    x += incremento
  ripeti
fine

```

La funzione *trasforma* riceve come parametri i valori di  $x$  e di  $y$  e calcola il corrispondente punto  $S$  sullo schermo.

In Java per trattare i punti si usa la classe **Point** che fa parte del package *java.awt*. Essa contiene due attributi pubblici  $x$  e  $y$  che conservano valori interi: essendo pubblici, possono essere letti e modificati direttamente.

Per esempio le seguenti istruzioni creano un punto P di coordinate (2,2) con successiva modifica del valore di x.

```
Point P = new Point(2,2);
P.x = 4*P.y;
```

Se si usa il costruttore della classe *Point* senza parametri, il punto viene inizializzato al valore (0,0). Sulla base delle considerazioni precedenti, vediamo ora l'applet per tracciare il grafico di una parabola definita dalla funzione  $y = ax^2 + bx + c$ . I coefficienti *a*, *b*, *c* vengono inseriti dall'utente.

La parabola può essere descritta come una classe con il seguente diagramma:

Parabola
a
b
c
valuta

La classe *Parabola* memorizza il valore dei tre coefficienti e tramite il metodo *valuta* consente di calcolare il valore della funzione in un punto *x* passato come parametro.

### IMPLEMENTAZIONE DELLA CLASSE (*Parabola.java*)

```
class Parabola
{
    private double a, b, c;

    public Parabola(double a, double b, double c)
    {
        this.a = a;
        this.b = b;
        this.c = c;
    }

    public double valuta(double x)
    {
        double y;
        y = a*x*x + b*x + c;
        return y;
    }
}
```

La classe *Grafico* si occupa della gestione dell'interfaccia grafica e dell'inserimento dei coefficienti.

L'interfaccia grafica per questo problema deve consentire l'inserimento del valore dei tre coefficienti della parabola oltre alla visualizzazione del grafico matematico. Nella parte bassa dell'applet vengono posizionate le tre caselle di testo riservate all'inserimento dei coefficienti e un pulsante per confermare il disegno di una nuova parabola. La parte centrale viene interamente occupata dall'area di disegno.

## APPLET JAVA (*Grafico.java*)

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class Grafico extends JApplet implements ActionListener
{
    private JPanel p = new JPanel();
    private JPanel comandi = new JPanel();
    private JTextField a = new JTextField(5);
    private JTextField b = new JTextField(5);
    private JTextField c = new JTextField(5);
    private JButton disegna = new JButton("Disegna");

    private Piano g = new Piano();

    public void init()
    {
        p.setLayout(new BorderLayout());

        disegna.addActionListener(this);
        comandi.add(new Label("A= ", Label.RIGHT));
        comandi.add(a);
        comandi.add(new Label("B= ", Label.RIGHT));
        comandi.add(b);
        comandi.add(new Label("C= ", Label.RIGHT));
        comandi.add(c);
        comandi.add(disegna);
        comandi.setBackground(Color.white);

        p.setBackground(Color.lightGray);
        p.add(comandi, "South");
        p.add(g, "Center");
        setContentPane(p);
    }

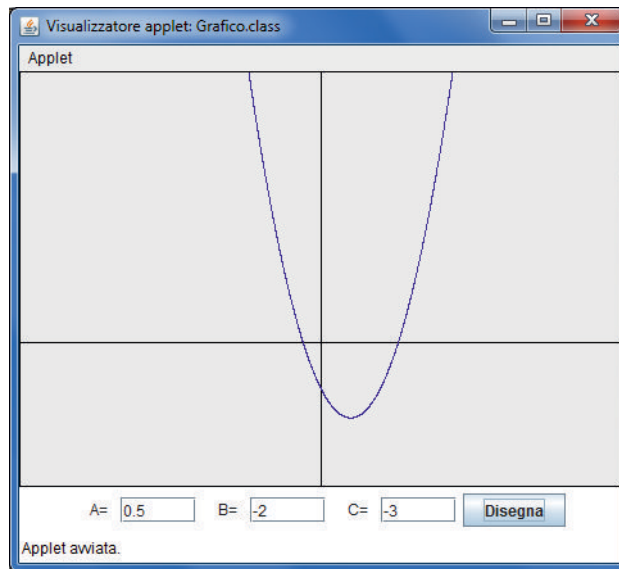
    // gestore di eventi per il bottone
    public void actionPerformed(ActionEvent e)
    {
        String bottone = e.getActionCommand();
        String numero;
        double valA, valB, valC;
        Parabola parab;

        if (bottone.equals("Disegna"))
        {
            try
            {
                numero = a.getText();
                valA = Double.valueOf(numero).doubleValue();
            }
        }
    }
}
```

```

        numero = b.getText();
        valB = Double.valueOf(numero).doubleValue();
        numero = c.getText();
        valC = Double.valueOf(numero).doubleValue();
        parab = new Parabola(valA, valB, valC);
        g.setParabola(parab);
    }
    catch(Exception exc)
    {
        parab = null;
        g.setParabola(parab);
        return;
    }
}
}
}
}

```



La classe *Piano* gestisce l'area di disegno, cioè il piano cartesiano, conosce la parabola e ne traccia il grafico quando viene richiesto.

Si osservi, nel metodo *paintComponent*, la chiamata per ereditare le operazioni grafiche standard:

```
super.paintComponent(g);
```

Il metodo *setParabola* forza il ridisegno del piano cartesiano con il metodo *repaint()*.

### IMPLEMENTAZIONE DELLA CLASSE (*Piano.java*)

```

import java.awt.*;
import javax.swing.*;

```

```

class Piano extends JPanel
{
    private Parabola parab = null;
    private final double XMAX = 20.0;
    private final double XMIN = -20.0;
    private final double YMAX = 18.0;
    private double YMIN;
    private double UNITA;
    private int w, h;

    public void paintComponent(Graphics g)
    {
        super.paintComponent(g);

        // calcola le dimensioni dell'area di disegno
        w = getSize().width;
        h = getSize().height;

        // calcola YMIN e l'unita' di misura
        UNITA = (double) w / (XMAX - XMIN);
        YMIN = YMAX - ((double) h / UNITA);

        // disegna gli assi
        disegnaAssi(g);

        // disegna la parabola
        if (parab != null)
        {
            disegnaParabola(g);
        }
    }

    // riceve come parametro le coordinate di un "punto reale" e
    // lo trasforma in un "punto schermo"
    public Point trasforma(double x, double y)
    {
        Point schermo = new Point();
        schermo.x = (int) Math.round((x - XMIN)*UNITA);
        schermo.y = (int) Math.round((YMAX - y)*UNITA);
        return schermo;
    }

    // traccia gli assi cartesiani
    public void disegnaAssi(Graphics g)
    {
        Point p1,p2;
        g.setColor(Color.black);
        g.drawRect(0,0,w-1,h-1);
        p1 = trasforma(XMIN, 0);
        p2 = trasforma(XMAX, 0);
        g.drawLine(p1.x, p1.y, p2.x, p2.y);
        p1 = trasforma(0, YMIN);
        p2 = trasforma(0, YMAX);
        g.drawLine(p1.x, p1.y, p2.x, p2.y);
    }
}

```

```

// traccia la parabola
public void disegnaParabola(Graphics g)
{
    double incremento=0.01;
    double x;
    Point p1, p2;

    g.setColor(Color.blue);
    p1 = trasforma(XMIN, parab.valuta(XMIN));
    for(x=XMIN+incremento; x<=XMAX; x+=incremento)
    {
        p2 = trasforma(x, parab.valuta(x));
        // disegna solo se i punti sono compresi nell'area grafica
        if ((p1.y>0) && (p1.y<h) && (p2.y>0) && (p2.y<h))
        {
            g.drawLine(p1.x, p1.y, p2.x, p2.y);
        }
        p1.x = p2.x;
        p1.y = p2.y;
    }
}

// imposta la nuova parabola e ridisegna il grafico
public void setParabola(Parabola parab)
{
    this.parab = parab;
    repaint();
}
}

```

L'applet *Grafico.class* viene richiamata da una pagina Web e posizionata al centro della finestra del browser.

#### **PAGINA WEB** (*Parabola.htm*)

```

<HTML>
<HEAD>
<TITLE>Parabola</TITLE>
</HEAD>
<BODY>
<APPLET CODE="Grafico.class" WIDTH="600" HEIGHT="500">
Applet non eseguibile.
</APPLET>
</BODY>
</HTML>

```

Utilizzando il precedente esempio come traccia, si possono disegnare i grafici di altre funzioni matematiche.