RAPPRESENTAZIONE TRIDIMENSIONALE NEL PIANO

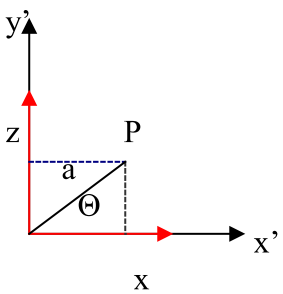
BY

“A Strange Site”

http://astrangesite.altervista.org

powered by
dott. Alessandro Strano

Per rappresentare il generico punto P(z, x, y) dello spazio a tre dimensioni in uno spazio a due dimensioni occorre “trasformare” le tre coordinate in una coppia di valori x’ e y’ ricorrendo alla “proiezione ortogonale”.



Immaginiamo di guardare il punto P (z, x, y) posti sull’asse y[1]; il grafico riportato sopra mostra la vista dalla suddetta posizione di osservazione. Ebbene, possiamo calcolare la misura delle “proiezioni ortogonali”[2] del segmento “a” ricorrendo alla trigonometria.

Si avrà:
$y' = a * sen(\Theta + \beta)$

---

[1] Siano x, y e z gli assi del sistema di riferimento cartesiano nello spazio tridimensionale e x’ y’ gli assi del sistema di riferimento nello spazio a due dimensioni.

[2] La “proiezione ortogonale” di un generico segmento “a” è il segmento “staccato” sull’asse cartesiano dalle perpendicolari condotte per gli estremi del primo.

$x' = a * \cos(\Theta + \beta)$

In cui Θ è l'angolo che il segmento "a" forma con l'asse x' e β è l'angolo di rotazione del sistema di riferimento cartesiano tridimensionale attorno all'asse y (nell'esempio β è zero).

Applicando le formule goniometriche di addizione[3] ricaviamo:
$y' = a * \text{sen}\,\Theta * \cos\beta + a * \cos\Theta * \text{sen}\,\beta = z * \cos\beta + x * \text{sen}\,\beta$
$x' = a * \cos\Theta * \cos\beta - a * \text{sen}\,\Theta * \text{sen}\,\beta = x * \cos\beta - z * \text{sen}\,\beta$

Se infine ruotiamo il sistema di riferimento cartesiano attorno all'asse "z" di un angolo α dobbiamo (applicando lo stesso procedimento) sostituire nelle formule la "x" con l'espressione $x * \cos\alpha - y * \text{sen}\,\alpha$.

In definitiva otteniamo le seguenti formule:
$k = x * \cos\alpha - y\,\text{sen}\,\alpha$
$y' = z * \cos\beta + k * \text{sen}\,\beta$
$x' = k * \cos\beta - z * \text{sen}\,\beta$

ove x' e y' sono appunto le coordinate cercate.

Il codice in Visual Basic[4] riportato sotto è una semplice applicazione per il disegno del grafico delle funzioni di due variabili basata sulle formule esposte sopra. Per provarlo, è sufficiente creare un "form" con un controllo "command" nel cui trigger "Click()" bisogna riportare il codice in oggetto. Per il grafico si è deciso di tracciare un segmento (parallelo all'asse y') tra i vari punti eccetto, ovviamente, dopo un punto di discontinuità; il "flag" TracciaLinea e la procedura di gestione di errori (GestErrore) servono proprio a questo scopo. Se si volessero tracciare i grafici di funzioni quadre[5] sarà sufficiente effettuare l'elaborazione sia per il punto P(z ,x , y) che per il punto P(-z, x, y), così come se si volessero tracciare dei

[3] $\text{sen}(\Theta + \beta) = \text{sen}\,\Theta * \cos\beta + \cos\Theta * \text{sen}\,\beta$; $\cos(\Theta + \beta) = \cos\Theta * \cos\beta - \text{sen}\,\Theta * \text{sen}\,\beta$.

Cfr. G. Zwirner – L. Scaglianti, Matematica per ragionieri periti commerciali e programmatori, Padova 1984, vol. II, p. 46.

[4] "Visual Basic" è marchio della Microsoft Corporation.

[5] Pensiamo ad esempio alla funzione z^2 = x^2 + y^2 che scriveremo come z = sqr(x^2 + y^2).

segmenti, paralleli all’asse x’, tra i punti sarà sufficiente aggiungere dopo il “next x” un altro ciclo simile al primo, ma in cui il “for y = ..” andrà scritto prima del “for x = ..” e la ”routine” di gestione errori dovrà essere modificata di consenguenza. Si badi che la soluzione proposta è sola una tra le possibili, forse la più intuitiva. Un altro approccio è ad esempio quello esposto in C. Cremonesi (Algoritmi di calcolo numerico, Milano 1990, pp. 6-12) che evita di tracciare le parti del grafico nascoste alla vista dell’osservatore, ma per l’esposizione di questo consiglio la lettura dell’opera citata.

```
On Error GoTo GestErrore
Dim x As Double, y As Double, z As Double
Dim x1 As Double, y1 As Double, k As Double, a As Double, b As Double
Dim l As Double, w As Double, r As Double, PiGreco As Double
Dim TracciaLinea As Boolean, zoom As Double
Cls
PiGreco = 4 * Atn(1)
w = Width / 2
l = Height / 2
'-------------------
' a= angolo di rotazione sull'asse z in gradi
' b= angolo di rotazione sull'asse y in gradi
' zoom = zoom
' Questi valori ed anche gli estremi dei
' due cicli FOR ed il valore di Step possono
' essere passati come parametri
'-------------------
a = 25
b = 0
zoom = w / 5
'-------------------
a = a * PiGreco / 180  'l’angolo è convertito in radianti
b = b * PiGreco / 180 'l’angolo è convertito in radianti
TracciaLinea = False
For x = -2.5 To 2.5 Step 0.1
   For y = -2.5 To 2.5 Step 0.1
      '-----------------------------
      'funzione
      '-----------------------------
      z = 3 * Sin(x) ^ 3 * Sin(y) ^ 3 'questa è la funzione a due variabili
      '-----------------------------
```

```
      k = x * Cos(a) - y * Sin(a)
      y1 = z * Cos(b) + k * Sin(b)
      x1 = k * Cos(b) - z * Sin(b)
     '------------------------------
     'x1 e y1 sono le coordinate nel piano
     'che di seguito vengono opportunamente
     'elaborate prima di tracciare il punto o
     ' la linea sul form (si fa in modo che l'origine
     ' degli assi coincida con il centro del form e
     'che i valori negativi di y ricadano nella parte
     'inferiore)
     '------------------------------
      x1 = (w + x1 * zoom)
      y1 = (l - y1 * zoom)
      If TracciaLinea Then
         Line -(x1, y1)
      Else
         PSet (x1, y1)
      End If
      TracciaLinea = True
ProssimoValore:
    Next y
  Next x
  Exit Sub
GestErrore:
  TracciaLinea = False
  Resume ProssimoValore
```