Interpolazione di Lagrange

Enrico Bertolazzi

Introduzione

Scopo: dati i punti $p_0, p_1, ..., p_N$ dove

$$p_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

trovare il polinomio $P(x) = \sum_{i=0}^{N} a_i x^i$ che interpola i punti dati, cioe'

$$P(x_i) = y_i \quad \text{per} \quad i = 0 .. N$$

Soluzione: Si costruiscono i polinomi

$$L_k(x) = \frac{F_k(x)}{F_k(x_k)}, \qquad F_k(x) = \left(\prod_{i=0}^{k-1} (x - x_i)\right) \left(\prod_{i=k+1}^{N} (x - x_i)\right)$$

e dal fatto che $L_i(x_j) = \delta_{i,j}$ otteniamo

$$P(x) = \sum_{i=0}^{N} y_i L_i(x)$$

Carica le librerie

```
> initialize ;
  with(plots):
```

initialize

```
Warning, the name changecoords has been redefined
```

Definisce la procedura Lagrange

```
> lagrange := proc(xy)
    local i, j, t, N, X, Y, L, poly ;

    N := nops(xy) - 1 ;
```

```
    X := array(0..N, [seq(xy[i][1],i=1..N+1)]) ;
    Y := array(0..N, [seq(xy[i][2],i=1..N+1)]) ;
    L := array(0..N) ;

    # costruisce i polinomi L
    for i from 0 to N do
      poly := t -> product( t-X[j], j=0..i-1 ) *
                   product( t-X[j], j=i+1..N ) ;
      poly := expand( poly(x) / poly(X[i])) ;
      L[i] := unapply(poly,x) ;
    end do;

    poly := unapply(sum(Y[j]*L[j](x),j=0..N),x);
    return L, poly ;
  end proc :
```

Procedura di Stampa

```
> lagrange_print := proc(L, poly, xy)
    local i, N ;
    N := nops(xy) - 1 ;
    print("polinomio = ", poly) ;
    for i from 0 to N do
      print(i,L[i]) ;
    end do ;
  end proc :
```

Procedura di disegno

```
> lagrange_plot := proc(L, poly, xy)
    local i, N, dx, xmin, xmax, XY, GA, GB, GC, GD ;

    N := nops(xy) - 1 ;

    xmin := min(seq(xy[i][1],i=1..N+1)) ;
    xmax := max(seq(xy[i][1],i=1..N+1)) ;
    dx   := (xmax-xmin)/20 ;
    XY   := [seq([xy[i][1],0],i=1..N+1)] ;
    GA   := plot(poly,xmin-dx..xmax+dx,thickness=2,color=blue):
    GB   := plot(L,xmin-dx..xmax+dx,color=green):
    GC   := plot(XY,xmin-dx..xmax+dx,style=point,
                 symbol=DIAMOND,symbolsize=20,color=red):
    GD   := plot(xy,xmin-dx..xmax+dx,style=point,
                 symbol=CIRCLE,symbolsize=20,color=blue):
    display({GA,GB,GC,GD},axes=normal,title='Lagrange');
  end proc :
```

Esempio d'uso

```
> # Definisce la funzione da approssimare
  f := x -> 1+x+x^3/4 ;
```

$$f := x \rightarrow 1 + x + \frac{1}{4} x^3$$

```
> # Definisce i punti da interpolare
  pnts := [seq([k,f(k)], k=-2..1)] ;
```

$$pnts := \left[[-2, -3], \left[-1, \frac{-1}{4} \right], [0, 1], \left[1, \frac{9}{4} \right] \right]$$

```
> # risolve il problema e stampa i risultati
  L, P := lagrange(pnts) ;
  lagrange_print(L, P, pnts) ;
  lagrange_plot(L, P, pnts) ;
```

$$L, P := L, poly$$

$$\text{"polinomio = "}, x \rightarrow 1 + x + \frac{1}{4} x^3$$

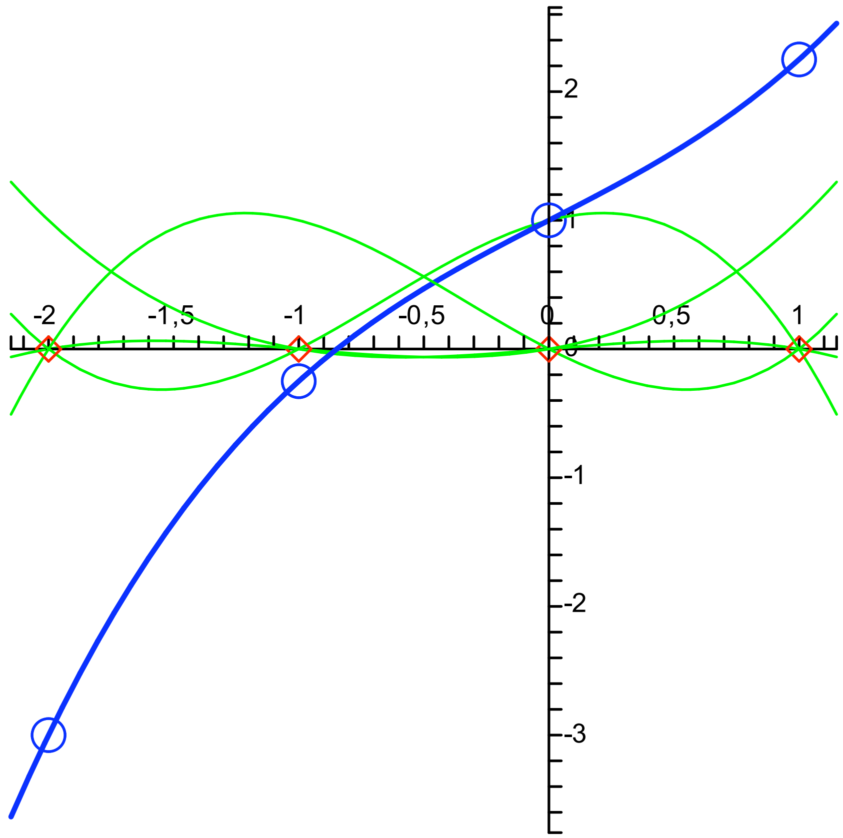
$$0, x \rightarrow -\frac{1}{6} x^3 + \frac{1}{6} x$$

$$1, x \rightarrow \frac{1}{2} x^3 + \frac{1}{2} x^2 - x$$

$$2, x \rightarrow -\frac{1}{2} x^3 - x^2 + \frac{1}{2} x + 1$$

$$3, x \rightarrow \frac{1}{6} x^3 + \frac{1}{2} x^2 + \frac{1}{3} x$$

Lagrange



Soluzione con le primitive Maple

```
> interp( [seq(pnts[k][1], k=1..nops(pnts))],
          [seq(pnts[k][2], k=1..nops(pnts))], x ) ;
```

$$\frac{1}{4}x^3 + x + 1$$

```
>
```