interpol_Lagrange

March 24, 2025

[1]:
```
reset()
```

[2]:
```
%display typeset
```

1 Polinómios de Lagrange

[3]:
```
def poli_Lagrange(xi):
    '''



    '''
    n=len(xi)
    x=var('x')
    L=[]
    for indice in range(n):
        num = 1
        den = 1
        res = 1
        for i in range(n):
            if i != indice:
                num = num * (x-xi[i])
                den = den * (xi[indice] - xi[i])
                res = res * (x-xi[i])/(xi[indice]-xi[i])
        print(num, den)
        res2 = res.expand()
        L.append(res2)
    return L
```

[4]:
```
xi=vector([1,-1,2]); xi
```

[4]: $(1, -1, 2)$

[5]:
```
poli_Lagrange(xi)
```

```
(x + 1)*(x - 2) -2
(x - 1)*(x - 2) 6
(x + 1)*(x - 1) 3
```

[5]:

$$\left[-\frac{1}{2}\,x^2+\frac{1}{2}\,x+1,\frac{1}{6}\,x^2-\frac{1}{2}\,x+\frac{1}{3},\frac{1}{3}\,x^2-\frac{1}{3}\right]$$

[6]:
```
def poli_interp_Lagrange(xk, fxk):
    '''



    '''
    n = len(xk)
    pol = 0
    L = poli_Lagrange(xk)
    for i in range(n):
        pol+=L[i]*fxk[i]
    return pol
```

[7]:
```
fxk=vector([6,0,12])
```

[8]:
```
poli_interp_Lagrange(xi, fxk)
```

```
(x + 1)*(x - 2) -2
(x - 1)*(x - 2) 6
(x + 1)*(x - 1) 3
```

[8]: $x^2+3\,x+2$

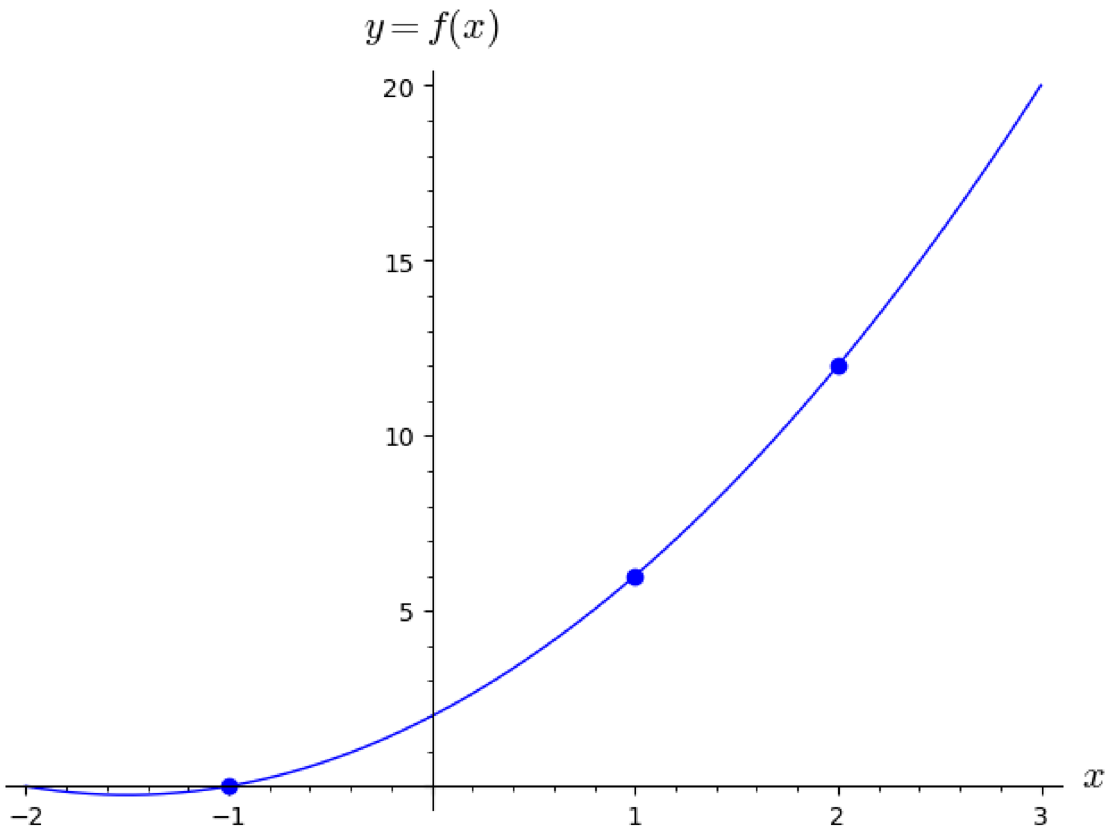
[9]:
```
def plot_poli_interp_Lagrange(xi,fxi,a,b):
    '''



    '''
    nodes=[]
    x = var('x')
    for i in range(len(xi)):
        nodes.append((xi[i],fxi[i]))
    f(x)=poli_interp_Lagrange(xi,fxi)
    P = plot(f(x),a,b)
    Q = line(nodes, marker='o', linestyle="", axes_labels=['$x$','$y=f(x)$'])
    (P+Q).show()
    return f
```

[10]:
```
plot_poli_interp_Lagrange(xi,fxk,-2,3)
```

```
(x + 1)*(x - 2) -2
(x - 1)*(x - 2) 6
(x + 1)*(x - 1) 3
```



[10]: $x \mapsto x^2 + 3\,x + 2$

[ ]: