S9-maxima.pdf

Sesión 9 Maxima resuelta

1º Cálculo

Grado en Ingeniería Informática

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada

<u>EJERCICIOS SESION 9 MÁXIMA RESUELTOS:</u>

<u>1.</u>

Sea la función f(x)=1/1+x^2.

a) **Considera 21 puntos igualmente distribuidos sobre la gráfica de la función f en el intervalo [−1,1]. Halla el polinomio interpolador de Lagrange que pasa por estos puntos y dibújalo junto a la gráfica de f. Interpreta el resultado y observa si la interpolación es buena.**

21 puntos -> 20 trozos

```
kill (all);

load(interpol)$

p:makelist(-1+k/10,k,0,20)$;

f(x):=1/(1+x^2);

puntos:makelist([p[i],f(p[i])],i,1,20);

define(g(x),expand(lagrange(puntos)))$;

wxdraw2d(
        point_type=7,
```

```
        color=red,
        points(puntos),
        color=dark-green,
        explicit(g(x),x,-2,2),
        color = blue,
        explicit(f(x),x,-2,2),
        yrange=[-2,2]
    );
```

Vemos que es una buena interpolación ya que ni se nota la diferencia de valor en los puntos dados con la gráfica. Pero más allá de esos puntos, se empieza a distorsionar, algo típico del polinomio Lagrange.

b) Elige ahora los 21 puntos sobre la gráfica de la función f en el intervalo [-5,5] y calcula el polinomio de Lagrange que pasa por dichos puntos. Interpreta de nuevo el resultado y observa si la interpolación es buena.

```
a:makelist(-5+k/2,k,0,20);

puntos2:makelist([a[i],f(a[i])],i,1,20);

define(p(x),expand(lagrange(puntos2)));
```

```
wxdraw2d(
        point_type=7,
        color=red,
        points(puntos2),
        color=dark-green,
        explicit(p(x),x,-6,6),
        color = blue,
        explicit(f(x),x,-6,6),
        yrange=[-2,2]
    );
```

Donde vemos una interpolación buena en la parte central, pero en los extremos se empieza a alejar a f(x), por lo que no es del todo buena la interpolación.

c) Calcula el polinomio interpolador por el método de los splines cúbicos en la situación del apartado anterior y compara el resultado con el polinomio de Lagrange obtenido.

```
define(q(x),cspline(puntos2))$
```

```
wxdraw2d(
        point_type=7,
        color=red,
        points(puntos2),
        color=dark-green,
        explicit(p(x),x,-6,6),
```

```
        color = blue,
        explicit(f(x),x,-6,6),
        color = pink,
        explicit(q(x),x,-6,6),
        yrange=[-2,2]
    );
```

Vemos que la línea rosa se aleja mucho más a la realidad de la función, por lo que la interpolación es bastante mejor que con lagrange

2.

Se considera la función f(x)=log(x). Queremos aproximar el valor de log(pi) utilizando:
¿Cuál de los dos polinomios aproxima mejor a log(pi)?

a)El polinomio de Lagrange que pasa por los cinco primeros naturales.

```
kill (all);

f(x):=log(x);

load(interpol)$

naturales:[[1,f(1)],[2,f(2)],[3,f(3)],[4,f(4)],[5,f(5)]];

define(t(x),expand(lagrange(naturales)));
```

b) El spline cúbico que pasa por dichos puntos.

Dibuja las tres gráficas protagonistas del problema.

```
define(q(x),cspline(naturales))$
```

(Para ver quien se aproxima mejor haríamos el error entre la función dada y con t(x) o q(x) o también se puede hacer a ojo quien se aproxima más a la realidad.)

```
float(q(%pi));

float(t(%pi));

float(f(%pi));   /*El valor real */
```

Se aproxima mejor a log de pi lagrange, pero una decena más exacto que spline. A partir del extremo de la derecha(no existe log negativo) la función de lagrange se aleja más del

log que el spline cúbico

```
float(t(%pi));

wxdraw2d(
        point_type=7,
        color=red,
        points(naturales),
        color=dark-green,
        explicit(t(x),x,0,7),
        color = blue,
        explicit(f(x),x,0,7),
        color = pink,
```

```
    explicit(q(x),x,0,7),
    yrange=[-1,3]
);
```