Sesión 7: Derivación e integración numéricas.

1 Derivación numérica

1.1 Idea básica y primer ejemplo

Cuando una función derivable es díficil de manejar, o cuando sólo conocemos unos cuantos valores de la misma, se utiliza la derivación numérica para realizar estimaciones de la "verdadera" función derivada.

Las fórmulas más usales para la primera derivada son:
*) f'(a)~(f(a+h)-f(a))/h,
(conocida como "fórmula de dos puntos de diferencia progresiva");
*) f'(a)~(f(a)-f(a-h))/h,
(conocida como "fórmula de dos puntos de diferencia regresiva");
*) f'(a)~(f(a+h)-f(a-h))/(2*h),
(conocida como "fórmula de tres puntos para el punto medio").

```
-->  f(x):=sin(x) $
     a:0 $
     h:0.1 $
```

```
-->  (f(a+h)-f(a))/h;
```

```
-->  (f(a)-f(a-h))/h;
```

```
-->  (f(a+h)-f(a-h))/(2*h);
```

```
-->  define(d1f(x),diff(f(x),x,1));
```

```
-->  d1f(0);
```

1.2 Exactitud

Las fórmulas dadas se denominan de tipo interpolatorio. En los siguientes ejemplos intentamos justificar esta terminología.

```
--> kill(a,h) $
```

Ejemplo1

```
--> g1(x):=3*x+1 $
```

```
--> (g1(a+h)-g1(a))/h;
```

```
--> ratsimp((g1(a+h)-g1(a))/h);
```

```
--> define(d1g1(x),diff(g1(x),x,1));
```

```
--> d1g1(a);
```

```
--> (g1(a+h)-g1(a-h))/(2*h);
```

```
--> ratsimp((g1(a+h)-g1(a-h))/(2*h));
```

Ejemplo 2

```
--> g2(x):=2*x^2+3*x+1 $
```

```
--> (g2(a+h)-g2(a))/h;
```

```
--> ratsimp((g2(a+h)-g2(a))/h);
```

```
--> define(d1g2(x),diff(g2(x),x,1));
```

```
--> d1g2(a);
```

```
--> (g2(a+h)-g2(a-h))/(2*h);
```

```
--> ratsimp((g2(a+h)-g2(a-h))/(2*h));
```

Ejemplo 3

```
--> g3(x):=-5*x^3+2*x^2+3*x+1 $
```

```
--> (g3(a+h)-g3(a))/h;
```

```
--> ratsimp((g3(a+h)-g3(a))/h);
```

```
--> define(d1g3(x),diff(g3(x),x,1));
```

```
--> d1g3(a);
```

```
--> (g3(a+h)-g3(a-h))/(2*h);
```

```
--> ratsimp((g3(a+h)-g3(a-h))/(2*h));
```

EJERCICIO 1:
¿Existe algún polinomio de grado 3 tal que la fórmula de tres puntos para el punto medio proporcione el valor exacto?

1.3 Problemas de redondeo

En la Sesión 1 calculamos la derivada de la función seno en x=%pi a partir de la definición de derivada. Esta manera de proceder coincide con tomar la primera de las fórmulas dadas.

```
--> kill(all);
```

```
--> f(x):=sin(x) $
    a : %pi,numer $

    for i : 1 thru 20 step 1 do (
        h : 10^-i,
        print([i, (f(a+h)-f(a))/h])
    ) $
```

Algunas veces obtenemos el resultado correcto (casi) por casualidad.

```
--> f(x):=sin(x) $
    a : 0.0 $

    for i : 1 thru 20 step 1 do (
        h : 10^-i,
        print([i, (f(a+h)-f(a))/h])
    ) $
```

2 Integración numérica (Fórmulas de cuadratura)

2.1 Idea básica y primer ejemplo

Al igual que en el cálculo de derivadas, podemos proceder con el cálculo de integrales definidas. De hecho, la definición de integral definida se relaciona con el límite de una serie de aproximaciones (las sumas de Riemann) resultantes de la aplicación de ciertas fórmulas de cuadratura.

Las fórmulas de cuadratura (para calcular la integral de una función f(x) continua en el intervalo [a,b]) más sencillas son:

```
    *) (b-a)*f(a)
    (Rectángulo a izquierdas)
    *) (b-a)*f(b)
    (Rectángulo a derechas)
    *) (b-a)*m, m=min{f(x)/a<=x<=b}
    (Rectángulo inferior)
    *) (b-a)*M, M=max{f(x)/a<=x<=b}
    (Rectángulo superior)
    *) (b-a)*f((a+b)/2)
    (Punto medio)
    *) (b-a)*(f(a)+f(b))/2
    (Trapecio)
    *) (b-a)*(f(a)+4*f((a+b)/2)+f(b))/6
    (Simpson)
```

```
--> kill(all) $
```

```
--> f(x):=sin(x) $
    a:0 $
    b:%pi $
```

```
--> (b-a)*f(a);
```

```
--> (b-a)*f(b);
```

```
--> (b-a)*f((a+b)/2);
```

```
--> (b-a)*f((a+b)/2), numer;
```

```
--> (b-a)*(f(a)+f(b))/2;
```

```
--> (b-a)*(f(a)+4*f((a+b)/2)+f(b))/6;
```

```
--> (b-a)*(f(a)+4*f((a+b)/2)+f(b))/6, numer;
```

```
--> integrate(f(x),x,0,%pi);
```

2.2 Exactitud

Las fórmulas dadas también son de tipo interpolatorio. En los siguientes ejemplos intentamos justificar nuevamente la terminología empleada.

```
--> kill(all)$
```

Ejemplo 1

```
--> p1(x) := 4*x-5 $
```

```
--> integrate(p1(x),x,a,b);
```

```
--> expand((b-a)*p1(a));
```

```
--> expand((b-a)*p1(b));
```

```
--> expand((b-a)*p1((a+b)/2));
```

```
--> expand((b-a)*(p1(a)+p1(b))/2);
```

```
--> expand((b-a)*(p1(a)+4*p1((a+b)/2)+p1(b))/6);
```

Ejemplo 2

```
--> kill(all)$
```

```
--> p2(x) := 3*x^2 + 4*x - 5 $
```

```
--> integrate(p2(x),x,a,b);
```

```
--> expand((b-a)*p2(a));
```

```
--> expand((b-a)*p2(b));
```

```
--> expand((b-a)*p2((a+b)/2));
```

```
--> expand((b-a)*(p2(a)+p2(b))/2);
```

```
--> expand((b-a)*(p2(a)+4*p2((a+b)/2)+p2(b))/6);
```

Ejemplo 3

```
--> kill(all)$
```

```
--> p3(x) := -4*x^3 + 3*x^2 + 4*x - 5 $
```

```
--> integrate(p3(x),x,a,b);
```

```
--> expand((b-a)*p3(a));
```

```
--> expand((b-a)*p3(b));
```

```
--> expand((b-a)*p3((a+b)/2));
```

```
--> expand((b-a)*(p3(a)+p3(b))/2);
```

```
--> expand((b-a)*(p3(a)+4*p3((a+b)/2)+p3(b))/6);
```

EJERCICIO 2:
Mediante el comando block define una función que ejecute una de las fórmulas de cuadratura vistas anteriormente.

2.3 Fórmulas compuestas

A la hora de la verdad, para calcular una integral definida por medio de fórmulas de cuadratura, se utilizan las fórmulas compuestas.

Se trata de subdividir el intervalo [a,b] en subintervalos de igual longitud, y aplicar una de las fórmulas anteriores en cada uno de estos "subintervalitos".

A mayor número de subintervalos considerados, más cercana al valor exacto de la integral será la aproximación obtenida.

EJERCICIO 3:
Determina las formulas de cuadratura compuestas correspondientes a las reglas del trapecio y de Simpson.

EJERCICIO 4:
Mediante el comando block, implementa una función para ejecutar las fórmulas compuestas del trapecio y de Simpson.
Como entrada de la función habrá que incluir el número de subintervalos.