

Metodo Dicotomico

Enrico Bertolazzi

[-] Introduzione

Scopo: dati l'intervallo $[a, b]$ dove trovare il nuovo intervallo $[a_1, b_1]$ tale che $[a_1, b_1]$ abbia misura inferiore ad un valore assegnato e che contenga almeno uno zero della funzione $f(x)$.

[-] Inizializza Maple

```
> restart;
```

[-] Procedura Refine

Definisce la procedura refine che dato un intervallo costruisce un intervallo ampio la metà del precedente che contiene almeno uno zero della funzione.

```
> refine := proc (IN, fun)
    local a, b, c ;

    a := IN[1] ;
    b := IN[2] ;
    c := (a+b)/2 ;

    if f(a) * f(c) < 0 then
        return [a,c] ;
    end if ;
    if f(c) * f(b) < 0 then
        return [c,b] ;
    end if ;
    return [c,c] ;
end;
```

```
refine := proc(IN, fun)
local a, b, c;
a := IN[1];
b := IN[2];
c := 1/2*a + 1/2*b;
if f(a)*f(c) < 0 then return [a, c] end if;
if f(c)*f(b) < 0 then return [c, b] end if;
return [c, c];
end proc;
```

[-] Procedura Dicotomico

Richiama piu volte la funzione refine fino a quando l'ampiezza dell'intervallo e' inferiore ad una ampiezza assegnata.

```
> dicotomico := proc(IN, eps, fun)
  local II ;
  II := IN ;
  while II[2]-II[1] > eps do
    II := refine(II, fun) ;
  end do ;
  return II ;
end ;
```

```
dicotomico := proc(IN, eps, fun)
local II;
  II := IN;
  while eps < II[2] - II[1] do II := refine(II, fun) end do;
  return II;
end proc;
```

Procedura ottimizzata (elimina i ricalcoli)

```
> dicotomico_opt := proc(IN, eps, fun)
  local a, b, c, fa, fb, fc ;

  a := IN[1] ;
  b := IN[2] ;
  fa := fun(a) ;
  fb := fun(b) ;

  while b-a > eps do
    c := (a+b)/2 ;
    fc := f(c) ;

    if fa*fc <= 0 then
      b := c ;
    end if ;

    if fc*fb <= 0 then
      a := c ;
    end if ;

  end do ;
  return [a,b] ;
end ;
```

```
dicotomico_opt := proc(IN, eps, fun)
local a, b, c, fa, fb, fc;
  a := IN[1];
  b := IN[2];
  fa := fun(a);
  fb := fun(b);
  while eps < b - a do c := 1/2*a + 1/2*b;
```

```

fc := f(c);
if fa*fc <= 0 then b := c end if;
if fc*fb <= 0 then a := c end if;
end do;
return [a, b];
end proc;

```

Esempio d'uso

> Definisce la funzione da approssimare

```
> f := x -> x/(1+x*x) ;
```

$$f := x \rightarrow \frac{x}{1+x^2}$$

> Risolve il problema e stampa i risultati

```
> dicotomico([-2,1],1/100,f) ;
dicotomico_opt([-2,1],1/100,f) ;
```

$$\left[\frac{-1}{512}, \frac{1}{256} \right]$$

$$\left[\frac{-1}{512}, \frac{1}{256} \right]$$

```
>
```