

Lezione 6 (parte prima)

Enrico Bertolazzi

```
> restart:
with(plots):
Warning, the name changecoords has been redefined

> # procedura dicotomico
# [a,b] = intervallo di partenza
# f      = funzione di riferimento
# epsi   = tolleranza ammessa per la radice

dicotomico := proc (a,b,f,epsi)
  local aa, bb, cc, N ;

  # in base ad epsi la tolleranza ammessa calcola
  # il numero di cifre necessarie nei conti.
  # ceil(x) = piu piccolo intero maggiore di x.
  N := 1+ceil(evalf(- log10(epsi))) ;

  # usa almeno 4 cifre decimali!.
  if N < 4 then N := 4 end if ;

  # trasforma il floating point l'intervallo [a,b]
  aa := evalf(a,N) ;
  bb := evalf(b,N) ;

  # controllo che in [a,b] ci possa essere una radice
  if f(aa)*f(bb) > 0 then
    # i dati iniziali non sono compatibili con
    # il metodo di bisezione
    return [a,b] ;
  end if ;

  if f(aa) = 0 then
    # trovata radice in aa
    return [aa,aa] ;
  end if ;

  if f(bb) = 0 then
    # trovata radice in bb
    return [bb,bb] ;
  end if ;

  # procedura di bisezione
  while bb-aa > epsi do
```

```

cc := evalf((aa+bb)/2,N) ;
if f(cc) = 0 then
  # caso fortunato cc e` una radice di f(x)
  return [cc,cc] ;
end if ;
if f(aa)*f(cc) < 0 then
  bb := cc ;
else
  aa := cc ;
end if ;
end ;
return [aa,bb] ;
end proc ;

```

```

> # esempio d'uso
fun := x -> (x-1)*(x-2)*(x-3) ;
      fun := x -> (x-1) (x-2) (x-3)

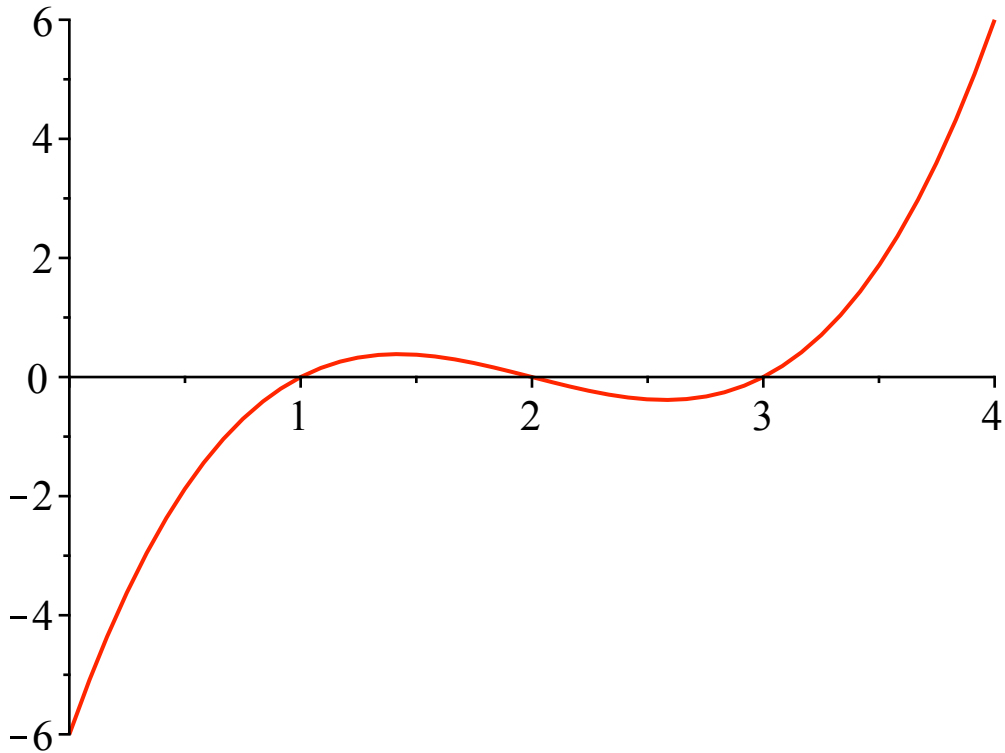
```

(1)

```

> plot(fun,0..4) ;

```



```

> dicotomico(0,1.2,fun,10^(-6)) ;
      [0.99999957, 1.0000001]

```

(2)