

Approssimazione dei numeri reali con numeri razionali

Mathematica dà la possibilità di approssimare un numero reale attraverso un numero razionale. La funzione da utilizzare è `Rationalize[x, Δ]`, essendo x il numero e Δ l'errore (al più) commesso. Esempio:

```
Rationalize[ $\sqrt{3}$ , 10-3]
```

```
 $\frac{71}{41}$ 
```

Se richiediamo la rappresentazione decimale, dobbiamo applicare la funzione `N[]`; ricordando che ogni funzione richiedente un solo argomento può essere richiamata in notazione postfissa, si ha:

```
Rationalize[ $\sqrt{3}$ , 10-3] // N
```

```
1.73171
```

È interessante studiare l'approssimazione di $\sqrt{3}$ al variare di Δ . A tale scopo utilizziamo il comando `Evaluate[]`, che ci permetterà di definire una funzione della precisione richiesta. Scriviamo:

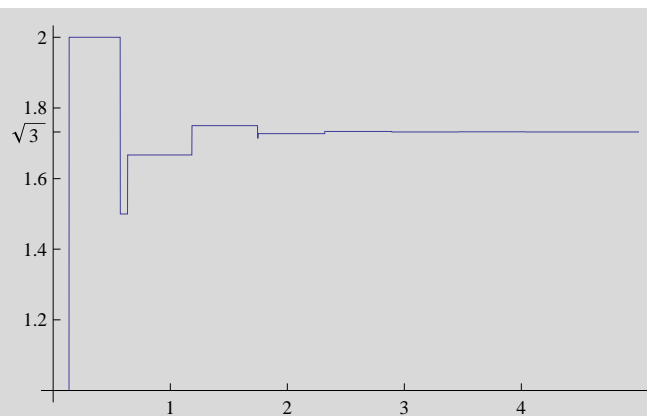
```
f[n_] := Rationalize[ $\sqrt{3}$ , 10-n]
```

Tracciamo un grafico di tale funzione (vedremo più avanti il significato delle varie opzioni utilizzate in `Plot`):

```

Plot[
  (*espressione della funzione*)
  f[n],
  (*intervallo in cui plottiamo*)
  {n, 0, 5},
  (*codominio*)
  PlotRange -> All,
  Ticks ->
  {
    (*punti sull'asse delle ascisse *)
    {1, 2, 3, 4},
    (*punti sull'asse delle ordinate*)
    {1.2, 1.4, 1.6, 1.8,  $\sqrt{3}$ , 2}
  }
]

```



La funzione `Chop[x, Δ]` restituisce 0 se x è un numero prossimo allo zero e se ne discosta di più di Δ . Ad esempio:

```
x = 5 - 4.999999999999
```

```
1.00009 × 10-12
```

```
Chop[x]
```

```
0
```

in quanto per default è $\Delta = 10^{-10}$.

```
Chop[x, 10-12]
```

```
1.00009 × 10-12
```

L'uso di `Chop` risulta molto utile quando è necessario valutare espressione molto complicate. In tal caso `Chop` elimina gli errori di arrotondamento.