



**UNIVERSITÀ  
DEGLI STUDI  
DI BERGAMO**

# **Scomposizione Polinomiale con regola di Ruffini**

Progetto di Informatica 3A

in Scala

Documentazione

Rachid Zouaoui  
matricola:1034867

July 2021

# 1 Introduzione

Questo progetto è stato scritto in Scala e si propone come obbiettivo quello di creare un algoritmo che sia in grado di scomporre completamente un polinomio a partire dal valore dei suoi coefficienti sfruttando la Regola di Ruffini.

## 2 Regola di Ruffini

La regola di Ruffini descrive un algoritmo che permette di effettuare la divisione tra due polinomi, questa può essere utile quando si deve scomporre un polinomio per il quale risultino inefficaci i classici metodi di scomposizione (raccolgimento a fattor comune,...ecc). Per poter applicare la regola è necessario come prima cosa identificare le radici del polinomio che si intende scomporre per far ciò si prendono i divisori del quoziente tra il termine noto e il termine di grado massimo. Una volta trovati i divisori si sostituiscono al valore della  $x$  nel polinomio calcolando sostanzialmente  $P(x)$  e se il risultato di tale valore risulta nullo allora si è trovata la radice altrimenti si procede con il divisore successivo. Se nessun divisore è radice il polinomio non è scomponibile. Dopo aver identificato la radice del polinomio si costruisce una tabella come in figura e si seguono i passi descritti.

|   |                               |                 |                               |                 |   |
|---|-------------------------------|-----------------|-------------------------------|-----------------|---|
|   | coeff. del termine di grado 2 |                 | coeff. del termine di grado 1 |                 | coeff. del termine di grado 0, termine noto |
| coeff. del termine di grado massimo (3) | 1                             | 0               | 2                             | -3              |   |
| radice del polinomio                    | 1                             |                 |                               |                 |   |
|   |                               | $1 \cdot 1 = 1$ | $1 \cdot 1 = 1$               | $1 \cdot 3 = 3$ |   |
|   | 1                             | $0 + 1 = 1$     | $2 + 1 = 3$                   | $3 - 3 = 0$     |   |

il resto deve essere zero!

Una volta eseguiti i procedimenti si otterranno i coefficienti del polinomio quoziente della divisione tra:  $P(x)$  e  $(x - radice)$ . Si procede poi iterativamente

fino a quando non ci sono più radici disponibili e il polinomio sarà completamente scomposto.

### 3 Algortimo

In primo luogo per lo sviluppo del algoritmo è stato necessario importare la libreria Math. In particolare, questa contiene la definizione del metodo pow, che riceve in input due valori e restituisce l'elevamento a potenza del primo per il secondo.

Dopodichè sono state indentificate le seguenti classi:

- **Polinomio:** E' la classe principale che prende in input nel costruttore una lista di interi che rappresente i coefficienti del polinomio.
- **Parabola:** E' una classe figlia della classe Polinomio che prende in input i coefficienti delle rispettive incognite e termine noto.
- **Retta:** E' una classe figlia della classe Polinomio che prende in input i coefficienti delle rispettive incognite e termine noto.

La lista dei coefficienti passato ai costruttori ha come primo elemento il termine noto. Inoltre per mostrare le funzionalità delle traits, è stata creata una trait molto semplice chiamata Printable, dove viene definito un metodo vuoto print. L'override di questo metodo viene fatto in tutte le altre classi e stampa a video il nome della classe seguito dagli elementi che caratterizzano l'oggetto (coefficienti per i polinomi; a, b e c per le parabole; m e q per le rette).

#### 3.1 Meotodo RicercaDivisori

Questo metodo permette di ottenere tutte le radici del polinomio. Questo, riceve in input il polinomio stesso e se questo è almeno di grado due, calcola le possibili radici. Per calcolare le radici, viene calcolato il valore di riferimento ovvero  $a_0/a_n$  e ne si individuano tutti i divisori (sia da positivi che negativi). Dopodichè viene filtrata la lista dei divisori ottenuti per prendere soltanto valori diversi da zero e che rendono 0 il valore del polinomio. Quindi le radici del polinomio

#### 3.2 Meotodo ScomposizioneRuffini

Questo metodo contiene il vero e proprio algoritmo della scomposizione di Ruffini, ed è stato fatto in maniera ricorsiva esso riceve in ingresso un polinomio ed il numero delle radici trovate (ciò permette di capire se siamo alla prima iterazione o meno), ne calcola successivamente le possibili radici e poi in base al caso in cui si trova sceglie l'operazione che dovrebbero essere effettuate:

- **Prima iterazione:** Non ci sono possibili radici, stampa a video che il polinomio non è scomponibile.
- **Iterazioni successive:** Se non ci sono nuove radici la scomposizione è terminata, altrimenti viene scelta la prima radice disponibile e il suo valore viene stampato a video. Vengono poi calcolati i coefficienti del polinomio risultante dalla divisione e con gli stessi viene creato un nuovo polinomio. Viene infine richiamata la stessa funzione in maniera ricorsiva, fornendo come input il nuovo polinomio.

Per mostrare l'utilizzo delle High Order Functions sono state create delle nuove versioni delle funzioni RicercaDivisori e ScomposizioneRuffini, chiamate RicercaDivisoriHOF e ScomposizioneRuffiniHOF. La prima, riceve come input un polinomio e una funzione di valutazione, la seconda richiama la nuova versione della prima. Per poter scrivere questo metodo, sono state scritte due nuove funzioni di valutazione, che funzionano esattamente come i metodi di valutazione presenti nelle classi Polinomio e Parabola.

In fine nel main son stati fatti alcuni esempi con dei polinomi.