# Mètodes Numèrics I Probabilístics

---

# Pràctica 3: Generació de variables aleatòries a $\mathbb{S}^n$

Grau: Matemàtica Computacional i Analítica de Dades

Assignatura: Mètodes Numèrics i Probabilístics

NIUs dels estudiants: 1600959, 1544112

Noms dels estudiants: Pau Blasco Roca, Alberto Real Quereda

# Índex

**1 Implementació del treball** **1**

**2 Funcions d' "aleatori.c"** **2**

2.1 Funció generaUniforme . . . . . 2

2.2 Funció generaNormal . . . . . 2

2.3 Funció muller . . . . . 3

**3 Manual d'utilització** **4**

**4 Anàlisi dels resultats** **4**

# 1 Implementació del treball

El programa principal anomenat “arrels” utilitza les següents llibreries:

- <stdio.h>
- <stdlib.h>
- “aleatori.h”

El header “aleatori.h” conté les funcions: “generaUniforme”, “generaNormal” i “muller”.

El programa principal repeteix 7 milions de vegades l’aloritme muller, i compta cuants polinomis hi ha de cada tipus, (0 arrels reals, 2 arrels reals o 4 arrels reals).

# 2 Funcions d' "aleatori.c"

Aquest arxiu conté les funcions:

- generaUniforme
- generaNormal
- muller

## 2.1 Funció generaUniforme

- generaUniforme(float **a**, float **b**, float* **r**):
  - **a**: nombre real que defineix l'inici de l'interval.
  - **b**: nombre real que defineix el final de l'interval.
  - ***r**: apuntador a una llista de dimensió 2 on guardarem els nombres aleatoris.

PROCÉS:

La funció guardarà a ***r** dos nombres de l'interval $[a, b]$ seleccionats pseudoaleatoriament.

## 2.2 Funció generaNormal

- generaNormal(float **mu**, float **sigma**):
  - **mu**: nombre real que defineix la mitjana a una distribució normal.
  - **sigma**: nombre real que defineix la desviació estàndar a una distribució normal.

PROCÉS:

La funció crida a "generaUniforme(-1,1,**r**)" per generar dos nombres aleatoris dins de l'interval [-1,1] i repeteix aquest procés mentre que la suma d'aquests nombres al quadrat sigui més gran que 1. Una vegada aconseguim dos valors vàlids calculem **n1** i **n2** de la següent manera:

$$n_1 = u\sqrt{\frac{-2\log s}{s}} \qquad n_2 = v\sqrt{\frac{-2\log s}{s}}$$

I fem:

$$n_1 = n_1 \cdot \sigma + \mu \qquad n_2 = n_2 \cdot \sigma + \mu$$

per tal d'ajustar aquests valors a la distribució normal amb la qual volem treballar.

### 2.3 Funció muller

- muller(double ***v**):
  - **v**: apuntador a una variable on guardarem el vector x normalitzat.

PROCÉS:

La funció crea el vector $x$, de dimensió 5 on guardem 5 nombres aleatoris que segueixen una distribució normal, amb paramentres ($\mu = 0$, $\sigma = 1$). Normalitzem el vector $x$ i el guardem a l'apuntador $v$.

## 3 Manual d'utilització

Per fer servir aquest programa primer l'hem de compilar amb:

gcc -c aleatori.c -Wall -o aleatori.o

gcc -c arrels.c -Wall -o arrels.o

gcc aleatori.o arrels.o -lm -o arrels -static

Una vegada hem compilat només hem de cridar al programa de la següent manera:

./arrels

Fent aquesta crida es mostra per pantalla la proporció de polinomis de cada tipus.

## 4 Anàlisi dels resultats

En el nostre cas les proporcions que trobem son les següents:

$$c0 : 0.085675 \quad c2 : 0.670093 \quad c4 : 0.244231$$

on:

$c_0$ són els polinomis amb 0 arrels reals

$c_2$ són els polinomis amb 2 arrels reals

$c_4$ són els polinomis amb 4 arrels reals

(La resta de casos es donen quan $a = 0$, que virtualment no pasarà mai).