

# Laboratorio di Calcolo

## Prova d'esame 24 genao 2018 (mattina)

### 1 Introduzione

La prova ha come tema principale la simulazione mediante tecniche di 'Monte Carlo' (ovvero per *campionamento*) di un semplice modello che descrive l'intensità di corrente all'interno di un conduttore in funzione della differenza di potenziale applicata e di altri parametri. Esso verrà sviluppato in C, con il supporto di script R.

### 2 Modello fisico

Il modello più semplice possibile di come si forma una corrente elettrica all'interno di un conduttore ('resistore') ai cui capi è applicata una differenza di potenziale  $\Delta V$  (misurata in Volt) è il seguente: se il resistore è lungo  $L$  gli elettroni del mezzo risentono di una forza

$$F = e \cdot \frac{\Delta V}{L}, \quad (1)$$

con  $e = 1.6 \times 10^{-19}$  C (valore assoluto della carica dell'elettrone).

Se un elettrone, partendo da fermo, non subisce urti per un tempo  $\Delta t$ , esso percorrerà in questo tempo una distanza

$$\Delta x = \frac{1}{2} \frac{F}{m} (\Delta t)^2, \quad (2)$$

con  $m = 9.1 \times 10^{-31}$  kg (massa dell'elettrone).

Gli elettroni subiscono però urti con gli atomi del mezzo nei quali la loro velocità viene (praticamente) azzerata. Segue una nuova accelerazione, nuovo 'stop', etc. etc. Ma questi processi di accelerazione/frenata/accelerazione/etc. non sono mai identici in quanto dipendono da effetti microscopici niente affatto elementari. Ne segue che gli intervalli di tempo  $\Delta t$  fra urti consecutivi non sono costanti, bensì 'randomici', descritti dalla funzione di *densità di probabilità esponenziale*<sup>1</sup>

$$f(\Delta t) = \frac{1}{\tau} e^{-\Delta t/\tau} \quad (0 \leq \Delta t < \infty), \quad (3)$$

con  $\tau = 10^{-15}$  s. Per questo motivo, nonostante il moto fra un urto e l'altro sia uniformemente accelerato, gli elettroni si muovono all'interno del conduttore con velocità *mediamente* costante. Ne segue che anche il *flusso di elettroni* e quindi anche l'intensità di corrente è mediamente costante nei diversi tratti del conduttore (trattandosi di grandi numeri di elettroni e di tempi di misura macroscopici rispetto a quelli fra un urto e l'altro).

---

<sup>1</sup>Se un fenomeno può avvenire per diversi tempi  $t$  con *densità di probabilità*  $f(t)$  significa che la probabilità che esso avvenga fra  $t_1$  e  $t_2$ , entrambi maggiori di zero, è data da

$$P(t_1 \leq t \leq t_2) = \int_{t_1}^{t_2} f(t) dt.$$

La collisione avviene con probabilità 1 (ovvero con certezza) fra  $t_1 = 0$  e  $t_2 = \infty$ , in quanto

$$\int_0^\infty \frac{1}{\tau} e^{-t/\tau} dt = 1.$$

(Questa distribuzione è chiamata **esponenziale** a causa della sua forma matematica.)

### 3 Implementazione numerica mediante campionamento

L'idea è di partire dalla generazione 'casuale' dei valori di  $\Delta t$ , simulando la distribuzione di probabilità dei tempi fra una collisione e l'altra data dall'Eq. (3). Da questa si passa ai valori, sempre randomici, delle distanze fra una collisione e l'altra.

Vengono quindi eseguite 'tante' estrazioni finché la distanza totale non arriva al valore di riferimento  $L$  (vedi nel seguito). Ma il tempo che impiegherà un elettrone a percorrere la distanza  $L$  non è sempre lo stesso, a causa della relazione non lineare fra  $\Delta x$  e  $\Delta t$  data dall'Eq. (2). Possiamo però, dal tempo medio impiegato a percorrere  $L$ , valutare una corrente media, che sarà la nostra *stima* della corrente e che indicheremo con  $\hat{I}$ , mentre delle fluttuazioni dei valori di tempo avremo una valutazione dell'incertezza su  $I$ , che indicheremo come  $\sigma(I)$ , a cui diamo il significato di '*incertezza standard*' della stima. I dettagli del calcolo li vedremo nel seguito.

### 4 Programmi/script preliminari

#### 4.1 Estrazione di numeri casuali distribuiti secondo una distribuzione esponenziale

In questa prima parte vogliamo scrivere la funzione che genera numeri casuali esponenziali a partire dal 'classico' generatore di numeri casuali uniformi e controllarla con il generatore di numeri casuali `rexp()` del linguaggio R. Si può infatti dimostrare che se  $x$  è una variabile casuale uniforme e 'continua' fra 0 e 1, allora la variabile

$$t = -\tau \cdot \log(x)$$

segue una distribuzione di probabilità esponenziale con parametro `tau`.

`test_esponenziale.R` :

- a) generare un vettore di  $N$  numeri casuali 'uniformi' (`'x'`);
- b) applicare la trasformazione descritta sopra per ricavarsi il vettore degli  $N$  tempi, usando `tau = 1`;
- c) calcolarsi mediante l'appropriata funzione ad 'alto livello' di R la media dei tempi;
- d) fare l'istogramma dei tempi;
- e) generare altri  $N$  tempi facendo però uso della funzione di sistema `rexp()` con parametro `rate = 1/tau` (attenzione!) e anche per questi fare l'istogramma e calcolare il valor medio;
- f) ripetere i punti (a)-(e) con `tau=2` e `tau=1/2`.

( $N$  può essere grande 'a piacere', senza esagerare – si giudichi da quanto regolare appare l'istogramma).

I risultati devono essere ragionevolmente consistenti ma ovviamente non saranno 'mai' coincidenti, trattandosi di campionamento random.

`test_esponenziale.c` A questo punto non rimane che scrivere il generatore esponenziale in C e il `main` per testarlo.

- a) scrivere la funzione `rexp()` che abbia `tau` come argomento e 'ritorni' un numero (pseudo-)casuale esponenziale ottenuto da a partire da un numero casuale uniforme fra 0 e 1, come abbiamo visto nel punto precedente;

- b) scrivere il `main` per testarla con lo stesso `N` e gli tre valori di `tau` usati precedentemente nello script di R; il `main` calcola anche le medie e le stampa, in modo che, confrontate con i valori ottenuti con R, ci diano confidenza che la funzione... funzioni bene.

Note:

- si usino ovunque variabili `double`;
- non c'è bisogno di definire un vettore in cui immagazzinare i numeri casuali 'ritornati' da `rexp()`;
- nel confronto fra i diversi metodi non si otterranno mai gli stessi valori medi, essendo le variabili 'aleatorie';
- per la *seed* del generatore di numeri casuali si usi il proprio numero di matricola (e si ricorda che il settaggio va fatto una sola volta nel `main`).

`test_statistica.c` Un altro lavoro preliminare consiste nella scrittura della funzione che fa un *riassunto statistico* (minimalista) degli elementi di un vettore, ovvero che calcoli la *media* e la *deviazione standard*<sup>2</sup> dei suoi valori, ovvero quello che in R fanno le function `mean()` e `sd()`, e più precisamente, dati  $N$  valori  $x_i$ ,

$$\bar{x} = \sum_{i=1}^N \frac{x_i}{N} \quad (4)$$

$$\sigma(x) = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N}}. \quad (5)$$

- a) scrivere la funzione `statistica()` alla quale si possa passare nel modo consueto un generico vettore di variabili `double` e rimandi indietro nel modo opportuno i valori `double` di `media` e `sigma`;
- b) scrivere un semplice `main` che testi la funzione usando il semplice vettore di `double` ordinati 1., 2., ..., 100.;
- c) si confronti il risultato con quanto calcolato in R mediante le funzioni `mean()` e `sd()`:
- `test_statistica.R`

Note:

- anche in questo nel programma C caso si faccia uso ovunque di variabili `double` (ovviamante a parte le variabili intere di loro natura);
- i valori della deviazione standard non coincideranno esattamente in quanto `sd()` di R la calcola con 'N-1' al denominatore della formula (5).

## 5 Valutazione numerica della corrente in funzione della differenza di potenziale

Siamo finalmente giunti al cuore della prova, ovvero al programma in C che calcola la corrente nel conduttore in funzione di  $\Delta V$  e degli altri parametri di interesse.

1. Il programma comincia col chiedere un valore di  $\Delta V$  (in Volt) che sia positivo, iterando la richiesta ad oltranza nel caso che esso sia negativo o nullo (si usino valori ragionevoli pensando ad esempio alle batterie)

---

<sup>2</sup>La deviazione standard è una misura della dispersione dei valori  $x_i$  intorno alla loro media

2. Quindi calcola il tempo impiegato da un singolo elettrone a percorrere la distanza di riferimento  $L = 10^{-6}$  m, mediante i seguenti passi (dopo aver settato il seed una tantum all’inizio con il solito numero di matricola):

- (a) definire le variabili  $\mathbf{t}$  e  $\mathbf{x}$ , che rappresentano tempo e posizione dell’elettrone e azzerarle.
- (b) estrarre un valore random esponenziale di tempo  $\mathbf{Dt}$  (tempo intercorso fra la partenza e l’urto), mediante la funzione scritta precedentemente e usando il valore di  $\tau$  riportato nella descrizione del modello; il valore ottenuto va aggiunto alla variabile  $\mathbf{t}$ ;
- (c) calcolare la distanza  $\mathbf{Dx}$  percorsa nel tempo  $\mathbf{Dt}$  facendo uso della Eq. (2), con  $F$  calcolata facendo uso della Eq. (1), e aggiungerla a  $\mathbf{x}$ .
- (d) Si procede, ripetendo i punti (b) e (c), fintanto che  $\mathbf{x}$  è inferiore di  $L$ .
- (e) Quando  $\mathbf{x}$  diventa maggiore o uguale di  $L$  il ciclo si interrompe e viene calcolato il tempo impiegato come  $t_{tot} = t - t_{corr}$ , dove

$$t_{corr} = \Delta t - \sqrt{(\Delta t)^2 - \frac{2m}{F}(x - L)}, \quad (6)$$

$x$  e  $\Delta t$  si riferiscono agli ultimi valori calcolato nel ciclo, mentre  $F$  va come sempre calcolato dall’Eq. (1).

- 3. Ripetere questi step per  $N = 100$  elettroni, salvando i valori di  $t_{tot}$  di ogni step nell’array `TT[100]`, che va anche salvato sul file `output.txt` (formato testo, un valore per riga).
- 4. Calcolare la *media* e la *deviazione standard* degli elementi del vettore `TT[100]` usando la funzione `statistica()`.
- 5. Da  $\overline{TT}$  e  $\sigma(TT)$  calcolare finalmente la *stima del valore* della corrente, ottenuto come

$$\hat{I} = \frac{e}{\overline{TT}},$$

e l’*incertezza standard* su tale valore, valutata come

$$\sigma(I) = \frac{\sigma(TT)}{\sqrt{N}} \cdot \frac{e}{\overline{TT}^2}.$$

Stampare quindi tali valori.

→ `calcolo_corrente.c`

## 6 Calcolo della corrente in R

Si rilegga da uno script R il file `output.txt` e, facendo uso delle funzioni ad alto livello del linguaggio, si ricalcolino il valor medio e la deviazione standard dei tempi totali, e da essi  $\hat{I}$  e  $\sigma(I)$ , seguendo la traccia del punto precedente.

→ `calcolo_corrente.R`