Oppgave 1

Advarsel: vi hadde kun diskuteret definition av sandsynlighedsfelte med endelig antal af udfald, mens i denne opgave bruger vi et felt med uendelig mange udfald, $S = [0,1] \times [0,1]$. Vi ser bort fra denne detalje; intutionen er tilstrekkeligt i dette eksperiment.

Skriv et program, som givet et positiv heltal n:

1. Genererer n tilfeldige punkter (x_i, y_i), $i = 1, \ldots, n$, hvor x_i og y_i har uniform fordeling i $[0, 1)$. I kan enten bruge `Math.random()` eller en av metoder fra afsnitt 9.1 i [Sauer].

2. Beregn antallet m av disse punkter, som hører til cirkelen med radius 1 og center i origo.

3. Vi kan så tilnærme sandsynligheden av hændelse "en tilfeldig punkt fra enhedskvardrat $[0,1] \times [0,1]$ hører til cirkelen" med m/n.

4. På den anden side, vi kan beregne sandsynligheden av denne hændelse som

$$\frac{\frac{1}{4}\text{areal(cirkel)}}{\text{areal(kvadrat)}} = \frac{\pi}{4},$$

   se Fig. 1. Ud fra det og den forrige tilnærmelse kan vi beregne en tilnærmelse til π.

5. Hvis I har tid: man kan vise, at fejlen mellem π og dens tilnærmelse aftager som $n^{-1/2}$. Dvs for at reducere fejlen med en faktor 10 skal vi bruge 100 så mange tilfeldige punkter. Vink: π er givet ved `Math.PI` i JavaScript.

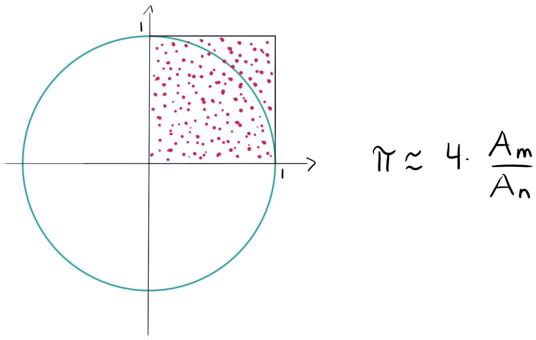

Figure 1: Illustration af vores Monte–Carlo eksperiment.

Oppgave 2

Nu vil vi gerne bestemme sandsyndligheten, at to tilfeldige korde på enhedscirkelen krydser hinanden, se Fig. 2.

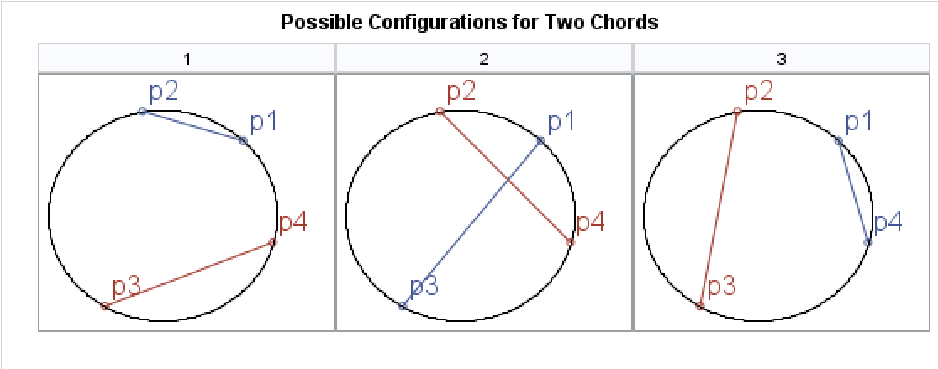

Figure 2: Mulige konfigurationer af to tilfeldige korde på enhedscirkelen.

Lad os first minde om, hvordan vi kan finde krydspunktet til to linjer i $\mathbb{R}^2$. Vi skal antage at enhver linje er givet i parametrisk form:

$$\vec{L}_1(\alpha) = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \alpha \begin{bmatrix} t_{x1} \\ t_{y1} \end{bmatrix}, \qquad \vec{L}_2(\beta) = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} + \beta \begin{bmatrix} t_{x2} \\ t_{y2} \end{bmatrix},$$

hvor $\alpha, \beta \in \mathbb{R}$ er to parametre. Linje $\vec{L}_1$ passer igennem punkt $(x_1, y_1) \in \mathbb{R}^2$ når $\alpha = 0$, or har retningsvektor $(t_{x1}, t_{y1}) \in \mathbb{R}^2$. Linje $\vec{L}_2$ passer igennem punkt $(x_2, y_2) \in \mathbb{R}^2$ når $\beta = 0$, or har retningsvektor $(t_{x2}, t_{y2}) \in \mathbb{R}^2$. For at bestemme om, og hvor, linjer $\vec{L}_1$ og $\vec{L}_2$ krydser, skal vi først løse en vektorligning $\vec{L}_1(\alpha) = \vec{L}_2(\beta)$ for parametrene (α, β). Krydsepunktet er da givet ved

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \vec{L}_1(\alpha) = \vec{L}_2(\beta).$$

Hvis punktet liger i enhedsdisken, dvs hvis $x_i^2 + y_i^2 \leq 1$, sa krydser de to korde.

Vi vil gerne bruge Monte–Carlo metoden til at tilnerme sandsynligheden numerisk, men vi skal først avgøre hvad "en tilfeldig korde" betyder.

Tilfeldig korde: model 1

Vi tager to tilfeldige vinkler $\theta_1, \theta_2 \in [0, 2\pi]$ som bestemmer endepunkter $p_1 = (\cos\theta_1, \sin\theta_1)$ og $p_2 = (\cos\theta_2, \sin\theta_2)$ på enhedscirkelen. Retningevektoren til linjen er givet ved $(t_x, t_y) = p_2 - p_1$.

Tilfeldig korde: model 2

Vi bestemmer en tilfeldig vinkel $\theta \in [0, 2\pi]$ og en tilfeldig avstand fra origo $d \in [0, 1]$. Kordets centrum kan da bestemmes som $(x_c, y_c) = (d\cos\theta, d\sin\theta)$. Kordets normallvektor er givet ved $\vec{n} = (\cos\theta, \sin\theta)$, og retningevektor er da $\vec{t} = (-\sin\theta, \cos\theta)$.

Tilfeldig korde: model 3

Vi bestemmer en tilfeldig punkt (x_c, y_c) i enhedsdisken som bestemmer kordets centrum. Det kan gøres ved at "kaste" to tilfeldige koordinater på enhedskvadrat $[-1, 1] \times [-1, 1]$. Hvis punktet ikke hører til enhedsdisken, så kaster vi igen. Kordets normallvektor er givet ved $\vec{n} = (x_c, y_c)$, og retningevektor er da $\vec{t} = (-y_c, x_c)$.

Oppgaven

Implementer et program som bruger Monte–Carlo metoden til at tilnerme krydsningssandsynligheden numerisk. Får I samme resultat når forskellige modeller af "tilfeldige korde" bruges? I kan læse om denne effekt på Wikipedia, `https://en.wikipedia.org/wiki/Bertrand_paradox_(probability)`