**Jačina električnog polja u 2D prostoru**

Priloženi kod i simulacija za cilj imaju računanje jačine električnog polja u svakoj točki prostora, određenog dimenzijama 600px \* 600px, za određenu distribuciju naboja. Račun se sastoji od više pojedinačnih dijelova, stoga ću u nastavku pojedinačno objasniti kako se koji dio računa i koja je njegova potreba.

\*Važno je napomenuti kako je uzet omjer piksela i mjerne jedinice metra, 1m = 100px

Slika na kojoj se prikazuje tekst, snimka zaslona, Font

Opis je automatski generiran**1) Račun udaljenosti naboja od svih točaka u prostoru = d**

X i Y su 2D numpy liste, koje u sebi sadržavaju x odnosno y vrijednost za svaku točku u prostoru i prilikom primjenjivanja matematičkih operatora nad njima, oni djeluju na svaki element. Kako bi se dobila udaljenost naboja (**Q**) od svake točke, jedno po jedno se računa udaljenost **Q**(Qx, Qy) od točaka **T**(Tx,Ty) prema formuli za udaljenost dviju točaka

**2) Dobivanje vektora smjera**

Kako bismo došli do jediničnog vektora smjera čiji će smjer imati i vektor električnog polja, potrebno je prvo pronaći smjer vektora od **Q do T**, što će se napraviti prema formuli za određivanje vektora

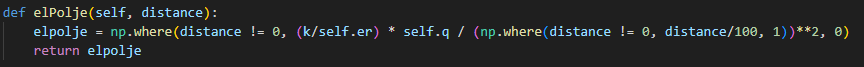
Gdje su: Tx-Qx = **d1**, Ty – Qy = **d2**

Ovo je korak gdje će se primijeniti i predznak ispred izraza vektora kako bi okrenuo orijentaciju ako je Q<0

**3)** **Dobivanje jediničnog vektora smjera =**

Sada kada imamo **udaljenost** Q od svih točaka i **vektore** od Q do svih točaka, možemo izračunati jedinične vektore smjera od Q prema svakoj točki, prema sljedećem izrazu

**4) Račun jakosti električnog polja naboja Q u svakoj točki prostora T |E|**



Koristeći se formulom za računanje jakosti električnog polja nekog naboja **Q** na točki udaljenoj za **d** u ovisnosti o relativnoj permitivnosti sredstva (), dobivamo skalarni iznos jakosti električnog polja za svaku točku, odnosno duljinu vektora električnog polja koju ćemo izračunati u sljedećem koraku. (k = 9∙109 N∙m2∙C-2)

Treba uočiti i da je ovo trenutak u kojemu udaljenost, koja je do tad bila izražena u broju piksela, pretvaramo u metre (**m**) tako da vrijedi **1m = 100px**; Naboj je izražen u **C** (kulonima).

**6)** **Dobivanje vektora električnog polja za svaki naboj i njihov zbroj**

U ovome koraku za svaku točku računamo vektor električnog polja prema izrazu i nadalje zbrajamo sve za svaki naboj u svakoj točki prema izrazu

**7) Račun ukupne jakosti električnog polja u svakoj točki prostora**

Preostaje nam još samo u listu *jakost\_el\_polja\_u\_prostoru* dodati ukupne jakosti električnog polja za svaku točku prostora koji se za svaku točku računa prema formuli duljine vektora

Ta lista, *jakost\_el\_polja\_u\_prostoru,* nadalje se prosljeđuje u *plt.imshow*

koji je odgovoran za crtanje tzv. ***heatmapa***, gdje će se lijepo nijansama boja prikazivati opadanje jakosti električnog polja s udaljenošću. Ukupni prostor biti će dimenzija 6\*6m, baš prema formuli pretvorbe piksela u metre od ranije.

Grafičko sučelje napravljeno u Pygameu uvelike pojednostavljuje korisnikovo upravljanje simulacijom i omogućuje mu mijenjanje ključnih varijabli kao i proizvoljnu distribuciju naboja u prostoru. Svaka čestica koju korisnik postavlja obilježena je nabojem (i predznakom) i svojom pozicijom u prostoru, dok je sam prostor obilježen dimenzijama i vrstom sredstva od kojeg je načinjen (tj. relativnom permitivnosti sredstva). Korisnik tako može mijenjati **iznos naboja**, pa i **predznak** naboja, **relativnu permitivnost sredstva**, a omogućena su mu i **4** načina distribucije naboja. **Prvo** može odabirom na prikazanom prostoru u Pygameu odrediti gdje će se i kakvi naboji nalaziti, **drugo** može odabrati distribuciju naboja poput kvadrata, **treće** distribuciju naboja u kružnici gdje sam određuje broja točaka u kružnici i posljednje, **četvrto** može unijeti koordinate naboja gdje bi želio da se nalazi. Kada se u prikazani prostor postavi barem jedan naboj, korisnik tipkom na **prihvati** započinje simulaciju i događa se prijelaz iz **main()** funkcije (odgovorne za grafičku prezentaciju koda, sučelje i distribuciju naboja) u drugu važnu funkciju **simulacija()** koja uz klase **Naboj()** [opisuje postavljenu česticu] **i Prostor()** [opisuje dimenzije i relativnu permitivnost prostora] ima ključnu ulogu u izvršavanju samog matematičkog modela i prezentacije unjetih podataka u obliku već spomenutog heatmap grafa.

Luka Uremović, 4.D