Numerikus módszerek 2. GY
Beadható programozási feladatok 1.

Az 1. beadható programozási házi feladatok közül 1 db szabadon választott feladat megoldásával kiváltható a beadandó házi feladatok 1. feladatsorának 5 db kötelezően megoldandó feladata, a hibátlanul megoldott programozási feladat tehát 15 pontot ér. A választott feladat megvalósításának nyelve szabadon választható, viszont az órán tanult *numerikus módszerekhez* külső programkönyvtár nem használható, a *numerikus probléma megoldása* a szerző saját implementációja kell, hogy legyen. Az egyéb funkciók (pl. rajzolás) ellátásához bármilyen könyvtár használható.

1. **Interpolációs polinom ábrázolása.** Írjunk programot, amely adott pontpárokhoz előállítja a Lagrange-interpolációs polinomot. A bemenet legyen $(x, y, \tilde{x})$, ahol $x, y \in \mathbb{R}^{n+1}$ vektorok $\tilde{x} \in \mathbb{R}^{N+1}$ vektor valamely $n, N \in \mathbb{N}$, $n, N \geq 1$ értékek mellett. A feladat azon p polinom tetszőleges x helyen vett helyettesítési értékeinek kiszámítása, amely az (x_i, y_i) pontokon áthalad, azaz amelyre

$$p(x_i) = y_i \quad (i = 0, \ldots, n)$$

A program kimenete legyen egy ábra, amelyen láthatók az $(x_i, y_i) \quad (i = 0, \ldots, n)$ pontpárok (pl. nagyobb piros körök), továbbá az ezekre illeszkedő $p \in \mathcal{P}_n$ interpolációs polinom értékei az $\tilde{x}_i$ pontokban, azaz az $(\tilde{x}_j, p(\tilde{x}_j)) \quad (j = 0, \ldots, N)$ pontpárok (pl. kisebb kék körök).

2. **Kép nagyítása.** Írjunk programot, ami beolvas egy tetszőleges formátumú képet, melyet a kép arányainak megtartásával az eredeti méretének tetszőleges egészszám-szorosára képes nagyítani. Tekintsünk első közelítésben egy szürkeárnyalatos *(grayscale)* képet. Egy $n \times m$ méretű szürkeárnyalatos kép egy $T \in \mathbb{R}^{m \times n}$ mátrixként ábrázolható. Legyen $k \in \mathbb{N}^+$, a feladat azon $\tilde{T} \in \mathbb{R}^{k(m-1)+1 \times k(n-1)+1}$ mátrix meghatározása, amelyben

$$\tilde{t}_{k(i-1)+1,k(j-1)+1} = t_{ij} \quad (i = 1, \ldots, n, \quad j = 1, \ldots, m)$$

és minden más $\tilde{t}_{pq}$ elem értékét az azt körülvevő 4 elemből számítunk ki bilineáris interpoláció segítségével. Pontosan, ha

$$k(i-1) + 1 \leq p < ki + 1 \quad k(j-1) + 1 \leq q < kj + 1$$

akkor $\tilde{t}_{pq}$ meghatározásához a $\{\tilde{t}_{k(i-1)+1,k(j-1)+1}, \tilde{t}_{ki+1,k(j-1)+1}, \tilde{t}_{ki+1,kj+1}, \tilde{t}_{k(i-1)+1,kj+1}\}$ elemeket használjuk (vegyük észre, hogy a fenti elemek a $\tilde{T}$ mátrix definíciója miatt rendre a

$$\{t_{ij}, t_{i+1,j}, t_{i+1,j+1}, t_{i,j+1}\}$$

elemekkel egyenlők). Ha színes képet használunk bemenetként, minden képpont három szín intenzitásából tevődik össze, a vörösből (R), zöldből (G) és kékből (B). Az egyes színcsatornák egyenként

viszont az előzőekben tárgyalt szürkeárnyalatos képeknek felelnek meg, a színes kép felnagyítása tehát annyiban különbözik a fenti feladattól, hogy az R, G és B intenzitásokat tartalmazó képeket külön-külön kell felnagyítanunk, majd a felnagyított $\tilde{R}, \tilde{G}, \tilde{B}$ színintenzitás-képekből előállítható a felnagyított színes kép. A program bemenete legyen (T, k), ahol T egy tetszőleges (szürkeárnyalatos, vagy színes) kép, $k \in \mathbb{N}^+$ pedig a nagyítás mértéke. A kimenet legyen a k-szorosára nagyított kép (megjelenítve, vagy a merevlemezre írva).

3. **Kulcskockás animáció.** Írjunk programot, ami egy téglalap kulcskockákkal vezérelt animációját valósítja meg. Legyen $t \in \mathbb{N}^{n+1}$ úgy, hogy $0 = t_0 < t_1 < t_2 < \cdots < t_n = T$, továbbá $x, y, \alpha, s \in \mathbb{R}^{n+1}$ vektorok. A t_i diszkrét időpillanatban *(frame)* a téglalap állapotát az $(x_i, y_i, \alpha_i, s_i)$ paraméternégyes írja le, ahol a paraméterek rendre: vízszintes pozíció, függőleges pozíció, elforgatás, skálázás. Számítsuk ki a paraméterek értékét a hiányzó képkockákban lineáris interpoláció segítségével. Keressük tehát azokat az $\tilde{x}, \tilde{y}, \tilde{\alpha}, \tilde{s} \in \mathbb{N}^{T+1}$ vektorokat, amelyekre

$$\tilde{x}_{t_i} = x_i \quad \tilde{y}_{t_i} = y_i \quad \tilde{\alpha}_{t_i} = \alpha_i \quad \tilde{s}_{t_i} = s_i$$

továbbá bármely két kulcskocka közötti paraméterérték a kulcskockákon felvett paraméterértékek közötti lineáris interpolációként áll elő, azaz például ha $t_i \leq j < t_{i+1}$, akkor $\tilde{x}_j$-t az x_i és x_{i+1} értékek közötti lineáris interpoláció határozza meg. A program bemenete legyen (t, x, y, α, s), kimenete pedig az animáció, azaz a $0, 1, \ldots, T$ időpillanatokban a transzformált téglalap kirajzolva, vagy képkockánként a merevlemezre írva.

4. **Vegyipari reakció.** Vegyész professzor barátunk segítséget kér tőlünk egy kísérlet elvégzéséhez, melyben az A és B komponens reakcióját szeretné megvizsgálni. A probléma a következő: a reakció végrehajtásához az A és B komponensből ugyanannyi mennyiség kell, hogy rendelkezésre álljon. Az A komponens radioaktív, aránylag rövid felezési idővel, ezért csak speciális helyről rendelhető $C > 0$ mennyiségben. Az A anyag mennyiségét időben a $Ce^{-\lambda t}$ függvény írja le az idő függvényében (az időt most órában mérjük), ahol $\lambda > 0$ a bomlási állandó. A B komponenst a professzor diákjai állítják elő (ösztöndíjért cserébe) nagyjából állandó sebességgel. A B anyagból egy óra alatt egy diák egységnyi mennyiséget keletkeztet, ezért a B anyag mennyiségét az időben a Dt összefüggés írja le, ahol $D \in \mathbb{N}^+$ a munkában részt vevő diákok száma. A reakciót abban a t időpillanatban kell elindítani, amikor az A anyagból és a B anyagból pontosan ugyanannyi áll rendelkezésre, azaz ha a

$$Ce^{-\lambda t} = Dt$$

egyenlet teljesül. Ezt az egyenletet viszont nehéz megoldani, ezért professzor barátunk már egy numerikus közelítéssel is megelégedne. Ehhez legyen $f(t) := Ce^{-\lambda t} - Dt$. A fenti feladat tehát ekvivalens az $f(t^*) = 0$ nemlineáris egyenlet megoldásával. Vegyük észre, hogy

$$f'(t) = -\lambda e^{-\lambda t} - D < 0 \qquad (t \in [0, +\infty))$$

azaz az f függvény szigorúan monoton csökkenő, így invertálható. Ezért a következő ötletünk támad: közelítsük a megoldást a

$$t_i = \frac{C}{D}\frac{i}{n} \qquad (i = 0, \ldots, n)$$

pontokra felírt inverz interpoláció segítségével (könnyű belátni, hogy a $[0, C/D]$ intervallum bármely λ esetén tartalmazza a keresett megoldást). A programunk bemenete legyen tehát (C, D, λ, n), a kimenet pedig legyen az inverz interpolációval (annak egy lépésével) meghatározott $\tilde{t}^*$, amely az $f(t^*) = 0$ egyenlet közelítő megoldása ($t^* = f^{-1}(0) \approx \tilde{L}_n(0) = \tilde{t}^*$).