1. zárthelyi feladatsor

Programcsomagok a numerikus módszerekben

2011. március 29.

Oldd meg az alábbi feladatokat MATLAB segítségével! Munkádhoz minden **saját** kézzel írott jegyzetet használhatsz. Netet, szomszédot nem, ennek megsértése 0 pontos eredményt von maga után. 45 perc áll a rendelkezésedre. Törekedj a szép, átlátható kódolásra.

Az elkészített fájlokat egyetlen `zip` fájlba tömörítve töltsd fel az *eLearning* rendszerbe!

1. Készítsd el a `inverzinterpol` nevű függvényt, amely egy vektorban adott kezdeti pontokból (x) kiindulva, az inverz interpolációs eljárással végrehajt n lépést.

A függvény adja vissza az n-dik lépés során kapott új pontot (xmo), valamint az itt felvett függvényértéket (fxmo), továbbá készítsen két ábrát, az egyiken a lépésenként kapott új pontok sorozatát kirajzolva, míg a másik ábrán magát a függvényt megjelenítve a $[min(x_i), max(x_i)]$ intervallumon.

A mintafüggvényt - amire végrehajtjuk a módszert - „inline” módon add meg az alábbi formában: `f = @(x)(cos(x)+4*x-2);`

Beküldendő a `inverzinterpol.m` fájl, valamint az `[xmo, fxmo] = inverzinterpol([-0.5,0.5,1],4);` hívás eredményeként kapott két ábra (képként).

Egy kis segítség: `function [eredm, fvertek] = inverzinterpol(x,n)`; `figure` parancs.

(5 pont)

2. Készíts egy `csebisevpolinomok` nevű függvényt, amely a paraméterként kapott vektorban szereplő sorszámú Csebisev polinomokat rajzolja ki egy ábrára, különböző színekkel. ($[-1, 1]$ intervallumon, 0.001 finomságú felosztás mellett). (pl. `csebisevpolinomok([0,3,5]);` hívás hatására kirajzolja T_0, T_3, valamint T_5 polinomokat)

Beküldendő a `csebisevpolinomok.m` fájl, valamint egy kép, mégpedig a példaként említett hívás eredményeként kapott ábra.

Segítség: `function csebisevpolinomok(v)`; `hold, acos` parancs használata.

A különböző színekhez például használhatod a következő rgb kódot: `[i/n, 1-i/n, i^2/n^2]`, ha n különböző polinomot kell kirajzolni, és ciklusban rajzoljuk ki, ahol i a ciklusváltozó.
(pl: `plot(x,y,'color',[i/n, 1-i/n, i^2/n^2]);`)

(4 pont)

3. Készítsd el a `fejerlepcso` függvényt, amely elkészíti adott alappontokhoz, és függvényértékekhez tartozó Fejér-féle lépcsőparabolás interpolációs polinomot, az órán szereplő képlet alapján.
Emlékeztető: $A_i(x) = [1 - 2\,(x - x_i)\, l_i'(x_i)]\, l_i^2(x)$

A paraméterek az egyszerűség kedvéért: `function fejerlepcso(x,f)`, sorrendben az alappontok vektora (`x`), valamint a hozzá tartozó függvényértékek az `f` vektorban.

Például próbáld ki az $f(x) = x \sin x$ függvény interpolálására a $\{0, \frac{\pi}{4}, \frac{\pi}{2}\}$ pontokon. Ábrázold a közelítés hibáját (abszolút értékét) a $[0, \frac{\pi}{2}]$ intervallumon. A visszatérési érték pedig legyen az interpoláció maximális hibája az előbb említett intervallumon. (0.001 finomságú felosztás mellett)

Beküldendő `fejerlepcso.m` fájl. (itt kép nem kell)

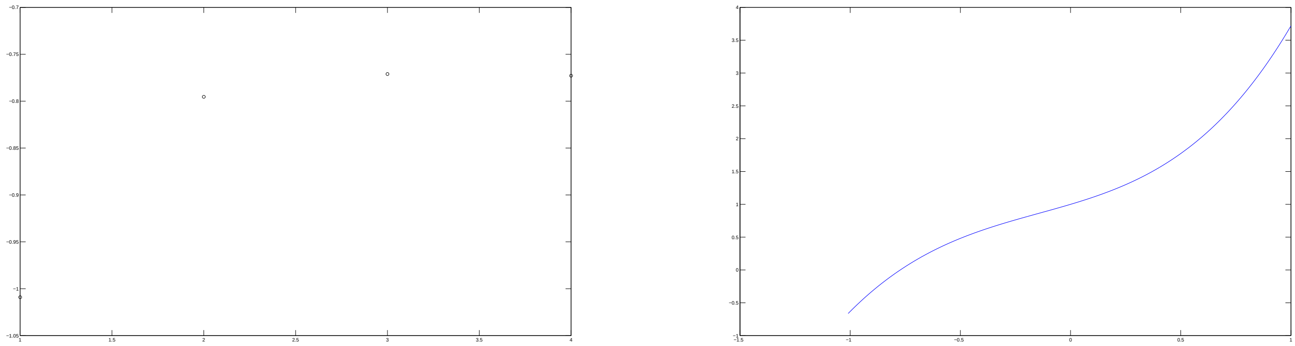
Segítség: a kérdéses hívás: `maxhiba = fejerlepcso([0,pi/4,pi/2],[0,sqrt(2)*pi/8,pi/2]);` a hiba számítása: `hibaxx = abs(fxx-hxx)`, feltéve, hogy `fxx` változó a megfelelő felosztás melletti függvényértékek vektora (ehhez a függvényt inline módon „beégetve” használhatod), `hxx` pedig az interpolációs polinom értékeinek vektora ugyanazon felosztás mellett; ezek után `maxhiba = max(hibaxx);` utasítással megkapható a maximális hiba.

(6 pont)

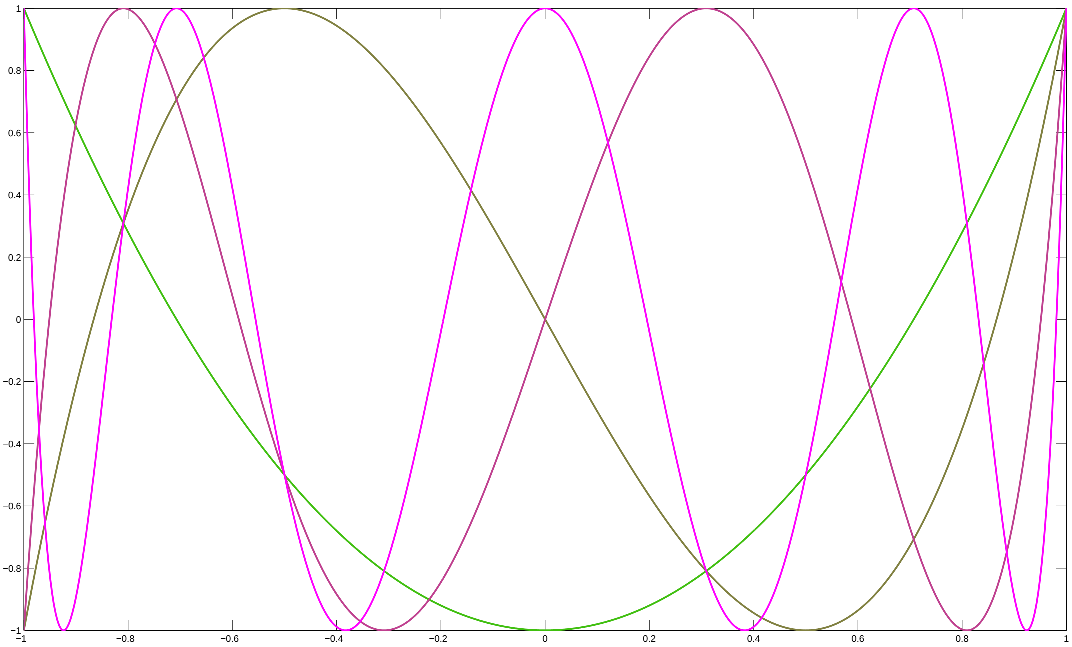
Ha így nem menne, akkor megoldható a Newton-féle táblázattal is, de ekkor a maximális pontszám:

(4 pont)

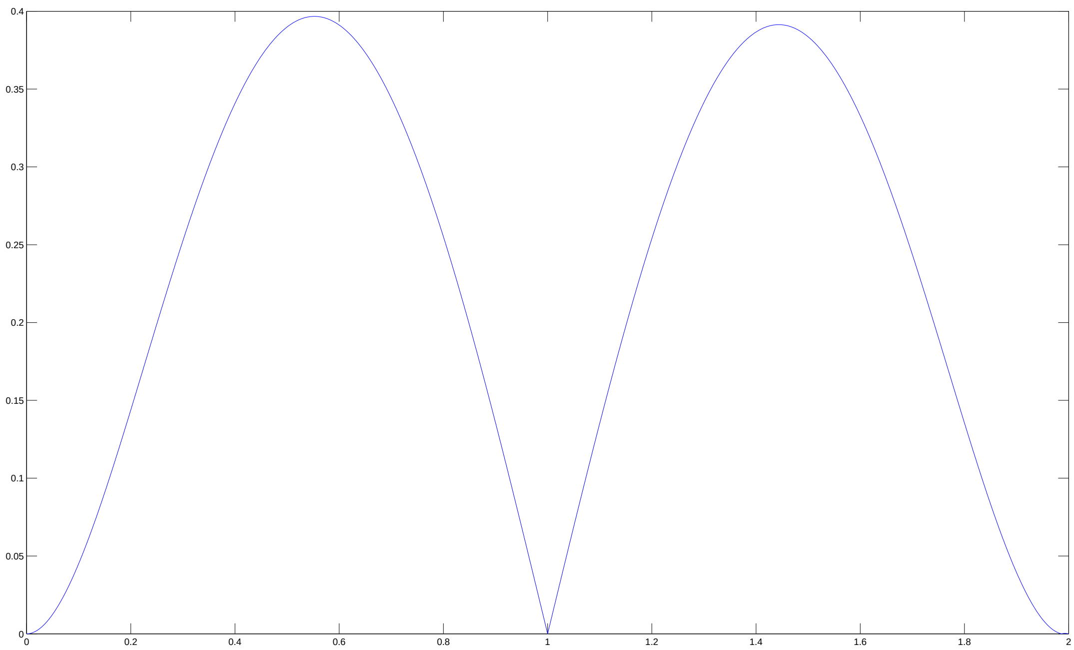
Jó munkát!



(a) 1. feladat, $e^x + x^3$ fv, ugyanazon paraméterekkel történő hívása esetén



(b) 2. feladat, $2, 3, 5, 8$ paraméter vektorral



(c) 3. feladat, példafüggvény, de $0, 1, 2$ pontokon interpolálva