Eötvös Loránd Tudományegyetem

Természettudományi Kar

Kerényi Péter

Gyökkeresés iterációval

BSc. szakdolgozat

Témavezető:

Sigray István

Analízis tanszék

Budapest, 2011

Tartalomjegyzék

Köszönetnyilvánítás

Szeretnék köszönetet mondani témavezetőmnek, Sigray Istvánnak, aki érdekes felvetéseivel és kérdéseimre adott körültekintő válaszaival segítette munkámat. Köszönettel tartozom csoporttársaimnak, kiváltképp Englert Ákosnak, akinek MatLab könyvét immáron másfél éve használom dolgozatom elkészítéséhez.

1. fejezet

Bevezetés

A dolgozatban különböző harmadfokú polinomok gyökeit keressük, mind valós, mind pedig komplex együtthatós polinomok esetében. A gyökök megtalálásához két iterációs eljárást használunk. Először a talán legismertebb eljárást a Newton-Raphson módszert (továbbiakban Newton-módszer) vizsgáljuk (2. fejezet, 3. fejezet), példákon keresztül bemutatjuk a módszer működését valamint a működés során fellépő esetleges hibákat, a hibák okait és lehetséges megoldásokat. Ezek után áttérünk egy Curt McMullen [McMullen 1987] által módosított algoritmusra (4. fejezet), példák segítségével kielemezzük annak működését és működésének sajátosságait. A két módszer elvégzése során kapott adatokat összehasonlítjuk és összefoglaljuk hasonlóságaikat és különbségeiket (5. fejezet). A példák megválasztásánál William Gilbertet [Gilbert 2001] követjük és négy tizedesjegy pontossággal dolgozunk.

A vizsgálatokhoz az általunk a `MatLab` programcsomagban készített `NewMeth` programot használjuk. A program leírását és fő paramétereit tartalmazza az A. függelék. A NewMeth program letölthető az ELTE TTK Matematika intézetének honlapjáról (`http://www.cs.elte.hu/~keppabt/documents/NewMeth.zip`). A program kicsomagolás (és a megfelelő `MatLab` program mellett) azonnal indítható, így az olvasó maga is meg tudja ismételni a dolgozatban tárgyalt példákat vagy akár új feladatok megoldására is használhatja azt.

2. fejezet

Newton-módszer a valós esetben

2.1. Newton-módszer és konvergencia-sebessége

A Newton-módszer egy approximációs eljárás, melynek alapvető ötlete, hogy a függvény egyik x^* gyökéhez "közeli" x_0 kezdőpontot ismerve próbálja meghatározni az $f(x)$ függvény egyik x^* gyökét.

Jelöljük ε_0 -val a nulladik tag hibáját, azaz $\varepsilon_0 = x^* - x_0$. Írjuk fel az $f(x)$ függvény x_0 körüli Taylor-sorát.

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \dots$$

$x = x^*$-ra az alábbi összefüggést kapjuk:

$$f(x^*) \approx f(x_0) + f'(x_0)(x^* - x_0)$$

$$0 \approx f(x_0) + f'(x_0)(\varepsilon_0 + x_0 - x_0)$$

$$\varepsilon_0 \approx -\frac{f(x_0)}{f'(x_0)}$$

$$x^* \approx x_0 - \frac{f(x_0)}{f'(x_0)}$$

Iterálva az eljárást a következő pont kiszámításának képlete:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

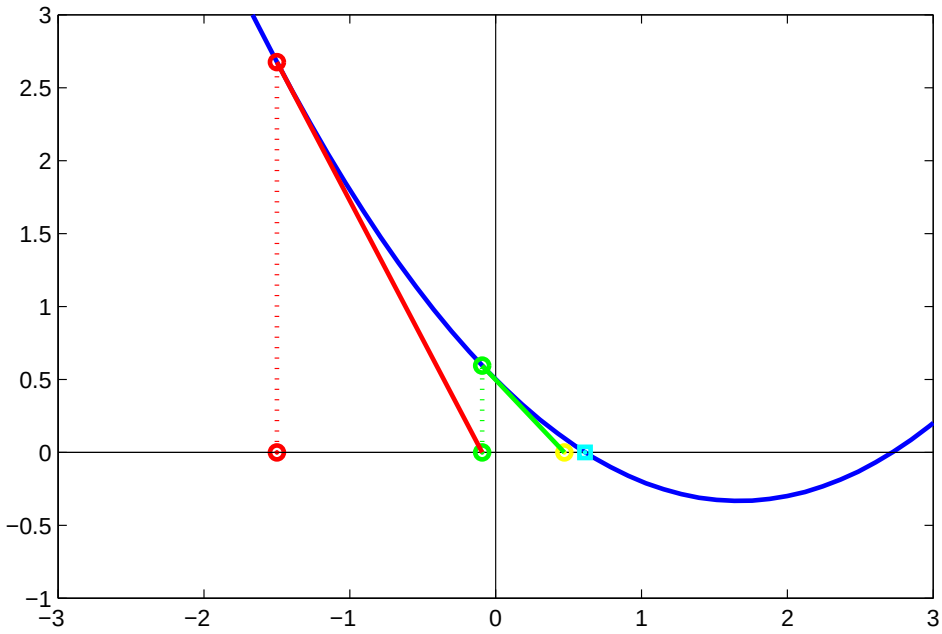
A leképezés tehát a következőképpen adható meg:

$$R(x) = x - \frac{f(x)}{f'(x)}$$

Geometriailag ez azt jelenti, hogy egy függvény x^* gyökét úgy találja meg a módszer, hogy a gyökhöz megfelelő távolságon belül található x_0 ponthoz érintőt húz, majd ezen érintő és az x tengely metszéspontja lesz az x_1 jobb közelítése az x^* gyöknek. Ezek után az eljárás iterálva folytatódik.

2.1. Példa. (2.1. ábra) Tekintsük az $f(x) = 0,3x^2 - x + 0,5$ függvényt. Indítsuk az iterációt az $x_0 = -1,5$ pontból.

`NewMeth(0,0.3,-1,0.5,-1.5,40,'newton')`



2.1. ábra. Az $f(x) = 0,3x^2 - x + 0,5$ függvény megoldása Newton-módszerrel. Első iterációs lépés pirossal, második iterációs lépés zölddel, gyök ciánnal jelölve.

Első iterációs lépés (2.1. ábrán pirossal jelölve):

$f(-1,5) = 2,675$

Az érintő meredeksége: $f'(-1,5) = -1,9$

Következő pont: $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = -1,5 - \frac{f(-1,5)}{f'(-1,5)} = -1,5 - \frac{2,675}{-1,9} = -0,0921$

Második iterációs lépés (2.1. ábrán zölddel jelölve):

$f(x_1) = 0,5946$

$f'(x_1) = -1,0552$

$x_2 = 0,4714$

Harmadik iterációs lépés:

$f(x_2) = 0,0952$

$f'(x_2) = -0,7171$

$x_3 = 0,6042$

Negyedik iterációs lépés:

$f(x_3) = 0,0053$

$f'(x_3) = -0,6374$

$x_4 = 0,6125$

Ötödik iterációs lépés:

$f(x_4) = 0,00004$

$f'(x_4) = -0,6325$

$x_5 = 0,6126$

$f(x_5) \approx 0$ (minimum 4 tizedesjegy pontossággal)

A megtalált gyök $0,6126$ (2.1. ábrán cián négyzettel jelölve).

A 2.1. Példa esetén a Newton-módszer az ötödik iterációs lépésben találja meg az egyik gyököt (minimum 4 tizedesjegy pontossággal). Általánosságban a konvergencia sebességéről a következő tételt fogalmazhatjuk meg.

2.1. Tétel. Ha az x_0 kezdőpont "elég jó" közelítése az x^* gyöknek, $f'(x_0) \neq 0$ és f kétszer folytonosan differenciálható x^* környezetében, akkor a Newton-módszer

kvadratikusan konvergens:

$$|x_{n+1} - x^*| \leq c|x_n - x^*|^2$$

Bizonyítás.

$$x_{n+1} - x^* = x_n - \frac{f(x_n)}{f'(x_n)} - x^* = x_n - x^* - \frac{f(x_n) - f(x^*)}{f'(x_n)} =$$

$$= \frac{f'(x_n)(x_n - x^*) - (f(x_n) - f(x^*))}{f'(x_n)}$$

A Lagrange középérték tétel szerint

$$\exists \xi_n \in [x_n, x^*] : f(x_n) - f(x^*) = f'(\xi_n)(x_n - x^*)$$

Tehát az előző kifejezés tovább egyenlő:

$$\frac{(f'(x_n) - f'(\xi_n))(x_n - x^*)}{f'(x_n)}$$

Ismét a Lagrange középérték tételt alkalmazva $\exists \tilde{\xi_n} \in [x_n, x^*] : f'(x_n) - f'(x^*) = f''(\tilde{\xi_n})(x_n - x^*)$, tehát az egyenlőséget tovább folytatva kapjuk, hogy:

$$\frac{f''(\tilde{\xi_n})(x_n - \xi_n)(x_n - x^*)}{f'(x_n)}$$

Ebből következik, hogy:

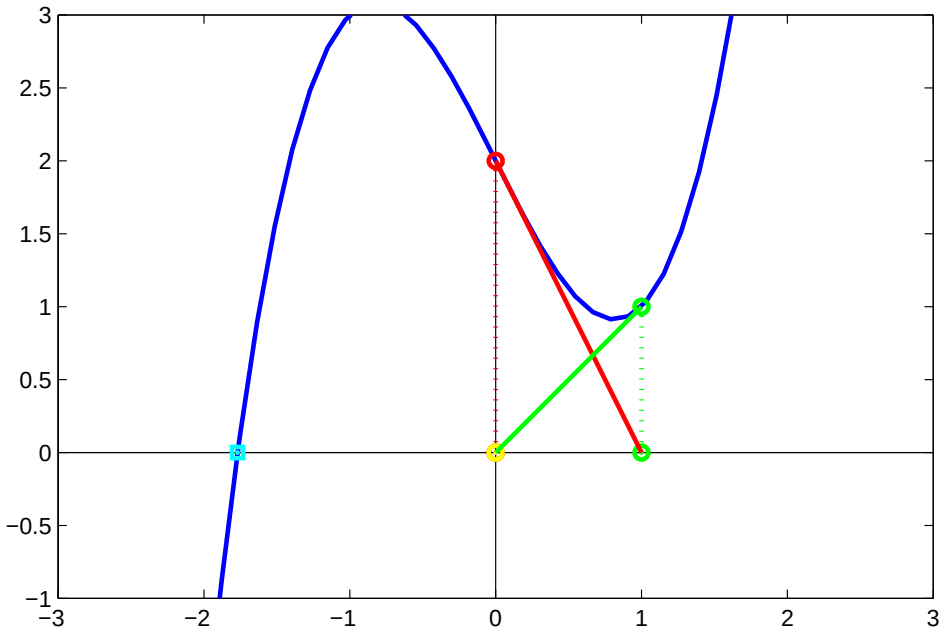
$$|x_{n+1} - x^*| \leq \frac{|f''(\tilde{\xi_n})|}{|f'(x_n)|}|x_n - x^*|^2$$

2.2. Ellenpéldák

Mi történik viszont akkor, ha nem "elég jó" kezdőpontot választunk vagy függvényünk nem differenciálható a megfelelő módon. Ilyenkor előfordul, hogy az iteráció lassabban vagy egyáltalán nem konvergál a gyökhöz. A következőkben erre látunk majd néhány példát.

2.2. Példa. (2.2. ábra) Tekintsük az $f(x) = x^3 - 2x + 2$ függvényt. Az iterációt az $x_0 = 0$ pontból indítva a következő eredményeket kapjuk:

```
NewMeth(1,0,-2,2,0,40,'newton')
```



2.2. ábra. Az $f(x) = x^3 - 2x + 2$ függvény megoldása Newton-módszerrel. Első iterációs lépés pirossal, második iterációs lépés zölddel, gyök ciánnal jelölve.

Első iterációs lépés (2.2. ábrán pirossal jelölve):

$f(x_0) = 2$

$f'(x_0) = -2$

$x_1 = 1$

Második iterációs lépés (2.2. ábrán zölddel jelölve):

$f(x_1) = 1$

$f'(x_1) = 1$

$x_2 = 0$

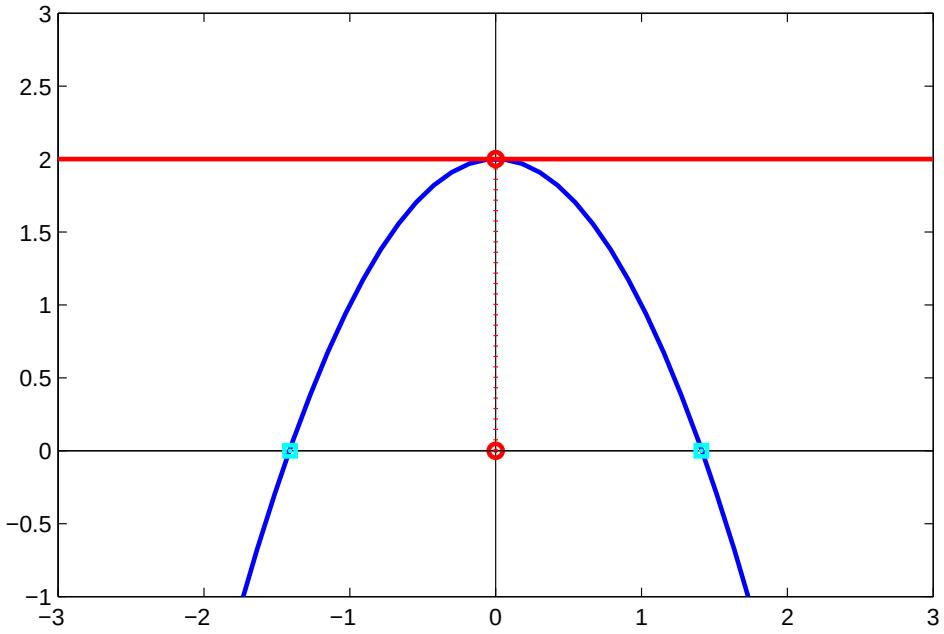
Innentől kezdve az iteráció ezen két pont között oszcillál vagyis egy 2 hosszú ciklusba

jutunk.

Ha a kezdőpontban a függvény deriváltja nulla, akkor nem tudjuk kiszámítani a következő pontot, mert ilyenkor nullával kellene osztani. Geometriailag ez azt jelenti, hogy a függvény egy szélsőértékében húzunk a függvényhez egy érintőt, ami pedig párhuzamos lesz az x tengellyel.

2.3. Példa. (2.3. ábra) Tekintsük az $f(x) = 2 - x^2$ függvényt. Az iterációt az $x_0 = 0$ pontból indítva a következő eredményeket kapjuk:

```
NewMeth(0,-1,0,2,0,40,'newton')
```



2.3. ábra. Az $f(x) = 2 - x^2$ függvény megoldása Newton-módszerrel. Első iterációs lépés pirossal, gyök ciánnal jelölve.

Első iterációs lépés (2.3. ábrán pirossal jelölve):

$f(x_0) = 2$

$f'(x_0) = 0$

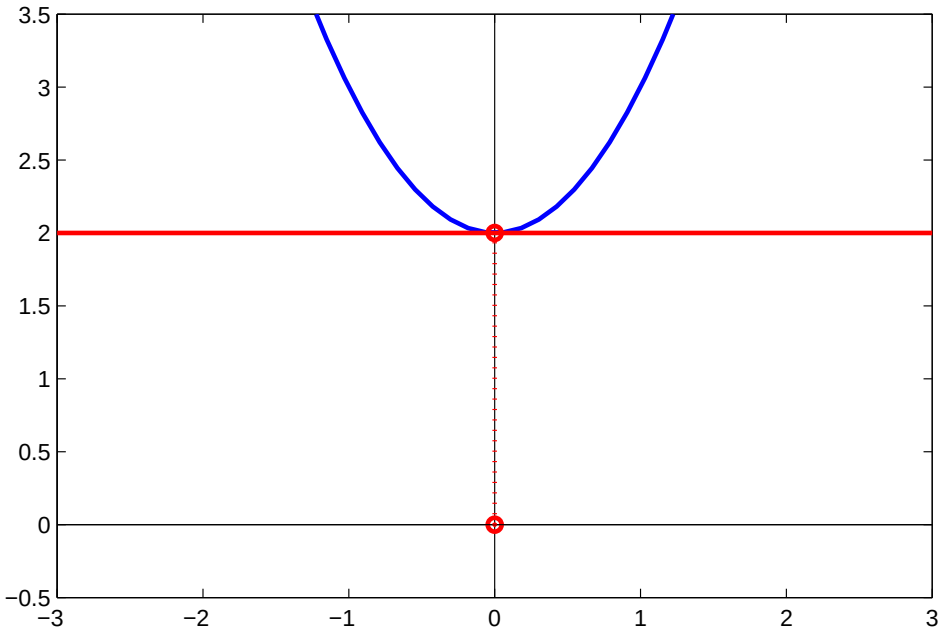
$x_1 = NaN$

Ez esetben a Newton-módszer nem találja meg a gyököt.

Mi történik akkor, ha a függvényünknek nincs valós gyöke?

2.4. Példa. (2.4. ábra) Tekintsük az $f(x) = x^2 + 2$ függvényt. Az iterációt az $x_0 = 1$ pontból indítva a következő eredményeket kapjuk:

```
NewMeth(0,1,0,2,1,40,'newton')
```



2.4. ábra. Az $f(x) = 2 + x^2$ függvény.

A módszer ilyenkor természetesen nem találja meg a gyököt.

2.3. Ellenpéldák egy lehetséges feloldása

A 2.2. Példában kezdőpontnak a $x_0 = 0$ pontot választottuk. Ez pedig az $R \circ R(x)$ leképezés egy fixpontja:

$$R \circ R(0) = R(R(0))$$

$$R(0) = 0 - \frac{0^3 - 2 \cdot 0 + 2}{3 \cdot 0^2 - 2} = 1$$

$$R(R(0)) = R(1) = 1 - \frac{1^3 - 2 \cdot 1 + 2}{3 \cdot 1^2 - 2} = 0$$

Ezen okból kifolyólag kaptunk a módszer elvégzése során egy 2 hosszú ciklust. Ebből látható, hogyha kezdőpontnak nem fixpontot, hanem ennél egy kis értékkel eltérő számot választunk, akkor ez a probléma jó eséllyel megszűnik és az iterációs eljárás meg fogja találni a gyököt. Ezt az eljárást nevezzük perturbálásnak.

2.5. Példa. Tekintsük az $f(x) = x^3 - 2x + 2$ függvényt. Az iterációt az $x_0 = 0$ pont helyett indítsuk az $x_0 = -0,13$ pontból (azért ezt választjuk, mert ez a legközelebbi érték, amit a 308 számjeggyel dolgozó programunk még kezelni tud). A következő eredményeket kapjuk:

```
NewMeth(1,0,-2,2,-0.13,40,'newton')
```

Első iterációs lépés:

$x_0 = -0,13$

$x_1 = 1,0283$

Második iterációs lépés:

$x_2 = 0,14188$

Harmadik iterációs lépés:

$x_3 = 1,0309$

...Negyvenötödik iterációs lépés:

$x_{45} = -1,8438$

...Negyvennyolcadik iterációs lépés:

$x_{48} = -1,7693$ $f(x) = 0$ (minimum 4 tizedesjegy pontossággal)

A megtalált gyök $-1,7693$.

A 2.5. Példa esetén a Newton-módszer a negyvennyolcadik iterációs lépésben találja meg az egyik gyököt (minimum 4 tizedesjegy pontossággal). Ez esetben tehát

a perturbálás segít és a módszernek sikerül gyököt találnia.

A 2.3. Példában kezdőpontnak a $x_0 = 0$ választottuk. A problémát a módszer képletében szereplő tört nevezőjében található függvény deriváltja és annak 0 értéke okozta. Most is azt várjuk, hogy a perturbálás segíteni fog.

2.6. Példa. Tekintsük az $f(x) = 2 - x^2$ függvényt. Az iterációt az $x_0 = 0$ pont helyett indítsuk az $x_0 = 0,01$ pontból. A következő eredményeket kapjuk:
`NewMeth(0,-1,0,2,0.01,40,'newton')`

Első iterációs lépés:
$x_0 = 0,01$
$x_1 = 100,005$

Második iterációs lépés:
$x_2 = 50,0125$

Harmadik iterációs lépés:
$x_3 = 25,02625$

... Tízedik iterációs lépés:
$x_{10} = 1,41422$

Tizenegyedik iterációs lépés:
$x_{10} = 1,41421$
$f(x) = 0$ (minimum 4 tizedesjegy pontossággal)

A megtalált gyök $1,41421$.

A 2.6. Példa esetén a Newton-módszer az tizenegyedik iterációs lépésben találja meg az egyik gyököt (minimum 4 tizedesjegy pontossággal). Ez esetben tehát a perturbálás ismét segít és a módszernek sikerül gyököt találnia.

A 2.4. Példában szereplő függvényünknek nincs valós gyöke, ezért az előző két példában (2.5. Példa, 2.6. Példa) működő perturbálás itt nem vezet eredményre és a Newton-módszer továbbra sem talál gyököt.

3. fejezet

Newton-módszer a komplex esetben

3.1. Newton-módszer

Az előző fejezetben tárgyalt 2.4. példában, mivel a függvénynek nincs valós gyöke, ezért a Newton-módszer semmilyen valós kezdőpontból nem talált gyököt, így a perturbálás sem segített. Mi történik viszont akkor, ha a kezdőponthoz nem egy kicsi valós számot, hanem egy kicsi képzetes részt adunk.

3.1. Példa. (3.1. ábra) Tekintsük az $f(z) = z^2 + 2$ függvényt. Indítsuk az iterációt az $z_0 = 1 + 0,01i$ pontból.

```
NewMeth(1,0,i,-1-i,2+i,40,'newton',2)
```

Első iterációs lépés (3.1. ábra):

$z_0 = 1 + 0,01i$

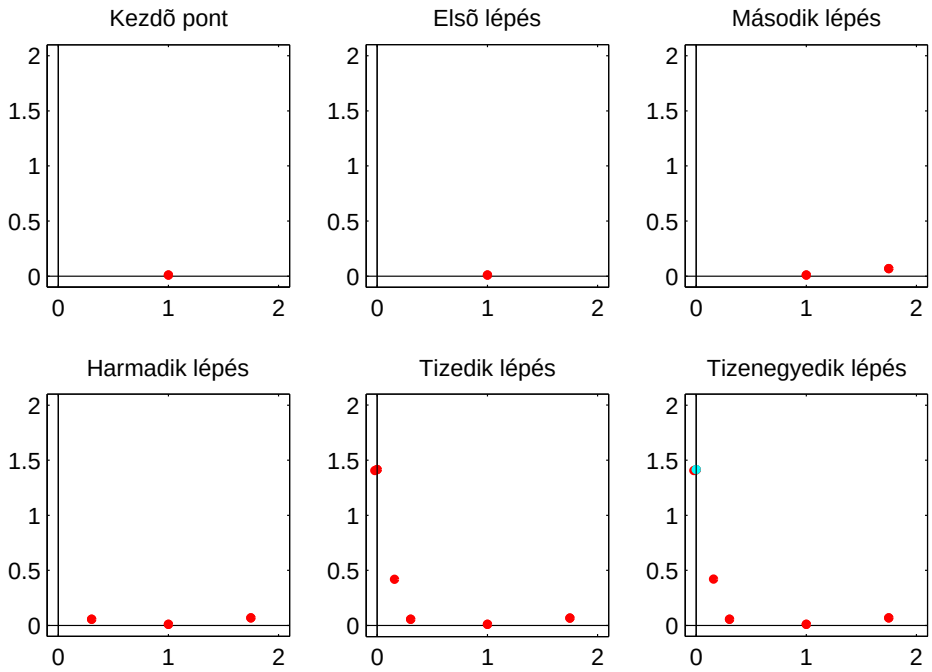
$z_1 = -0,4999 + 0,015i$

Második iterációs lépés:

$z_2 = 1,7486 + 0,0675i$

Harmadik iterációs lépés:

$z_3 = 0,3033 + 0,0558i$



3.1. ábra. Az $f(z) = z^2 + 2$ függvény megoldása Newton-módszerrel. Iterációs lépések pirossal, gyök ciánnal jelölve.

... Tízedik iterációs lépés:

$z_{10} = 0,0001 + 1,414i$

Tizenegyedik iterációs lépés:

$z_{11} = 1,4142i$

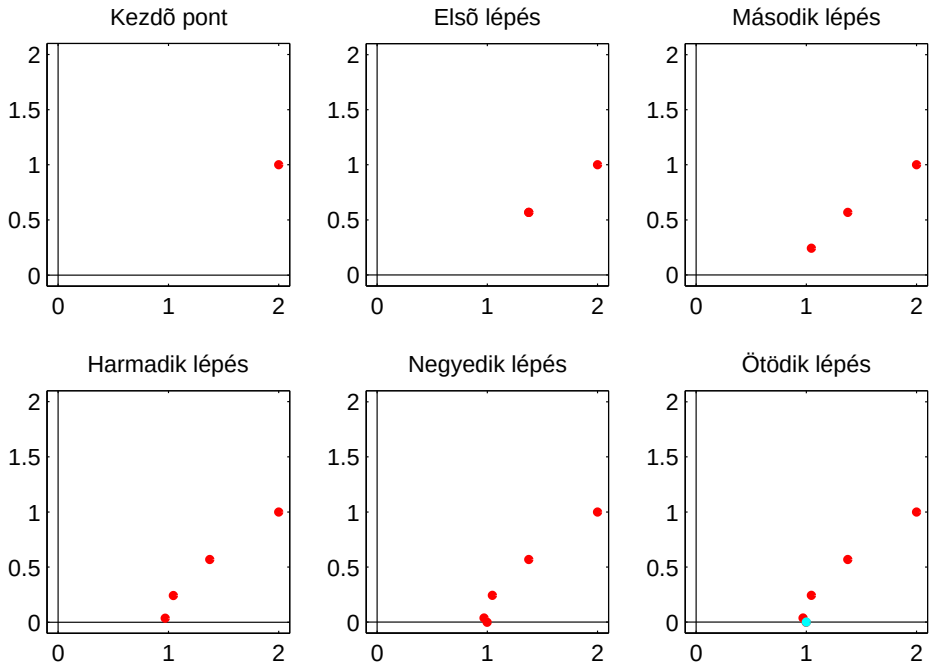
$f(z_{11}) = 0$

A megtalált gyök $1,4142i$, melyet a Newton-módszer az tizenegyedik iterációs lépésben talál meg (minimum 4 tizedesjegy pontossággal).

A Newton-módszer azonban nem csak valós, hanem komplex együtthatós polinomokra is működik. A következőkben erre fogunk néhány példát látni.

3.2. Példa. (3.2. ábra) Tekintsük az $f(z) = (z-1)(z+i)(z+1-i)$ függvényt. Indítsuk az iterációt az $z_0 = 2 + i$ pontból.

```
NewMeth(1,0,i,-1-i,2+i,40,'newton',2)
```



3.2. ábra. Az $f(z) = (z-1)(z+i)(z+1-i)$ függvény megoldása Newton-módszerrel. Iterációs lépések pirossal, gyök ciánnal jelölve.

Első iterációs lépés (3.2. ábra):

$z_0 = 2 + i$

Következő pont: $z_1 = z_0 - \frac{f(z_0)}{f'(z_0)} = 2 + i - \frac{f(2+i)}{f'(2+i)} = 1,376 + 0,568i$

Második iterációs lépés:

$z_2 = 1,045 + 0,2421i$

Harmadik iterációs lépés:

$z_3 = 0,9703 + 0,037i$

Negyedik iterációs lépés:

$z_4 = 0,9988 - 0,0018i$

Ötödik iterációs lépés:

$z_5 = 1$

$f(z_5) = 0$

A gyök 1 (3.2. ábrán ciánnal jelölve), melyet a Newton-módszer az ötödik iterációs lépésben talál meg (minimum 4 tizedesjegy pontossággal).

3.2. Newton-módszer különböző kezdőpontokból indítva

A következőkben különböző polinomok gyökeit keressük Newton-módszer segítségével. Ehhez az általunk a `MatLab` programcsomagban készített `NewMeth` programot használjuk (ld. A. függelék). A különböző gyökhöz konvergáló kezdőpontokat különböző színnel ábrázolja a komplex számsíkon a program.

3.3. Példa. (3.3. ábra) Tekintsük az $f(z) = (z-1)(z+i)(z+1-i)$ függvényt.

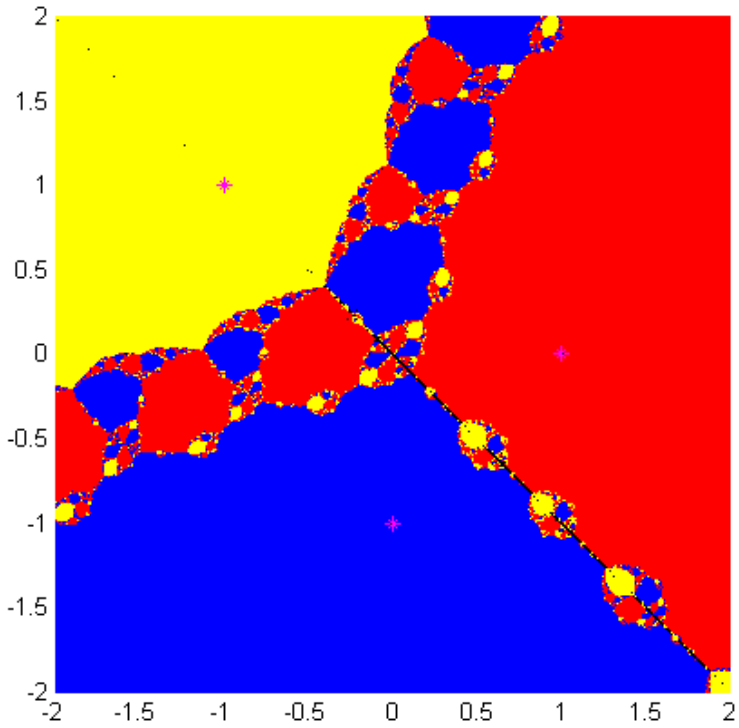
```
NewMeth(1,0,i,-1-i,testarea(-2,2,-2,2,100),40,'newton')
```

A 3.3. ábrán jól láthatóak a különböző színű részek, ahonnan a három gyök valamelyikébe konvergál az iteráció. A sárgával jelölt részből a $-1+i$, a kékkel jelölt részből $-i$, a pirossal jelölt részből pedig a 1 gyökhöz konvergál az iteráció. A fekete kezdőpontokból 40 iterációs lépés alatt nem konvergál, de ezeknél a kezdőpontoknál is segít a perturbálás, mert egy kis számot hozzáadva már valamelyik színes részbe kerülünk.

3.4. Példa. (3.4. ábra) Tekintsük az $f(z) = (z-1)(z+0,5-0,33i)(z+0,5+0,33i)$ függvényt.

```
NewMeth(1,0,-0.6411,-0.3589,testarea(-2,2,-2,2,100),40,'newton')
```

A 3.4. ábrán a már korábban látott részeken kívül megjelentek kisebb-nagyobb fe-



3.3. ábra. Az $f(z) = (z-1)(z+i)(z+1-i)$ függvény megoldása Newton-módszerrel. Sárga kezdőpontokból $-1+i$, kék kezdőpontokból $-i$, piros kezdőpontokból 1 gyökhöz konvergál az iteráció. A fekete kezdőpontokból 40 iterációs lépés alatt nem konvergál.

kete színű csomók. Ezekből a részekből indított iterációk nem konvergálnak egyik gyökhöz sem. Most nagyítsuk ki az egyik ilyen fekete részt (3.5. ábra).

```
NewMeth(1,0,-0.6411,-0.3589,testarea(0.4,1.2,1.2,2,450),40,'newton')
```

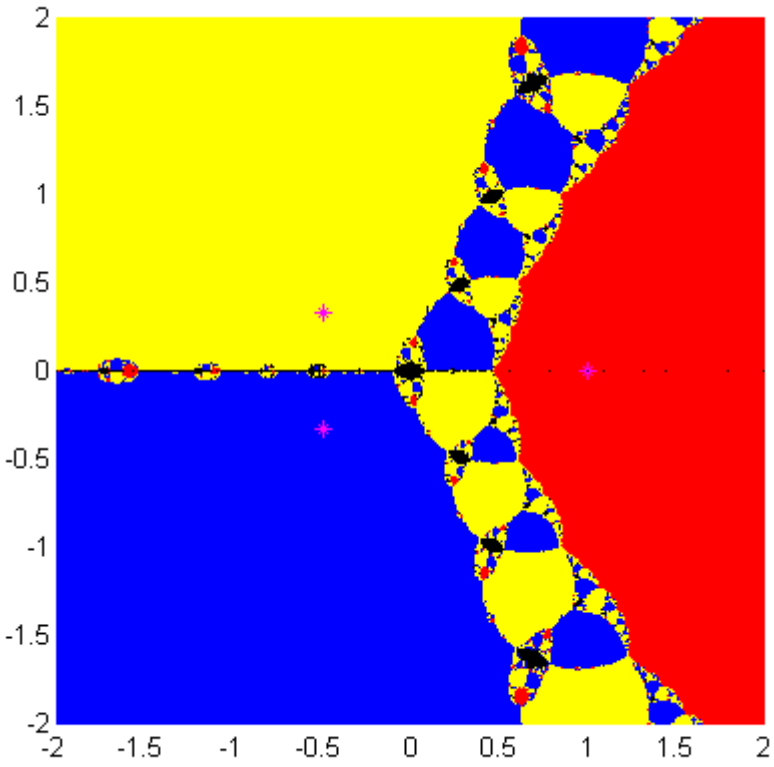
A 3.5. ábrán jól láthatók a különböző fekete színű, azaz nem konvergáló belső pontokkal rendelkező részek. A következőkben vizsgáljuk meg, hogy mi történik pontosan, ha egy ilyen fekete részből indítjuk az iterációt.

3.5. Példa. (3.6. ábra) Tekintsük az $f(z) = (z-1)(z+0,5-0,33i)(z+0,5+0,33i)$ függvényt. Indítsuk az iterációt az $z_0 = 0,7+1,6i$ pontból.

```
NewMeth(1,0,-0.6411,-0.3589,0.7+1.6i,40,'newton')
```

Első iterációs lépés (3.6. ábra):

$z_0 = 0,7+1,6i$



3.4. ábra. Az $f(z) = (z-1)(z+0,5-0,33i)(z+0,5+0,33i)$ függvény megoldása Newton-módszerrel. Sárga kezdőpontokból $-0,5+0,33i$, kék kezdőpontokból $-0,5-0,33i$, piros kezdőpontokból 1 gyökhöz konvergál az iteráció. A fekete kezdőpontokból 40 iterációs lépés alatt nem konvergál.

Következő pont: $z_1 = z_0 - \frac{f(z_0)}{f'(z_0)} = 0,7 + 1,6i - \frac{f(0,7+1,6i)}{f'(0,7+1,6i)} = 0,4676 + 0,9678i$

Második iterációs lépés:

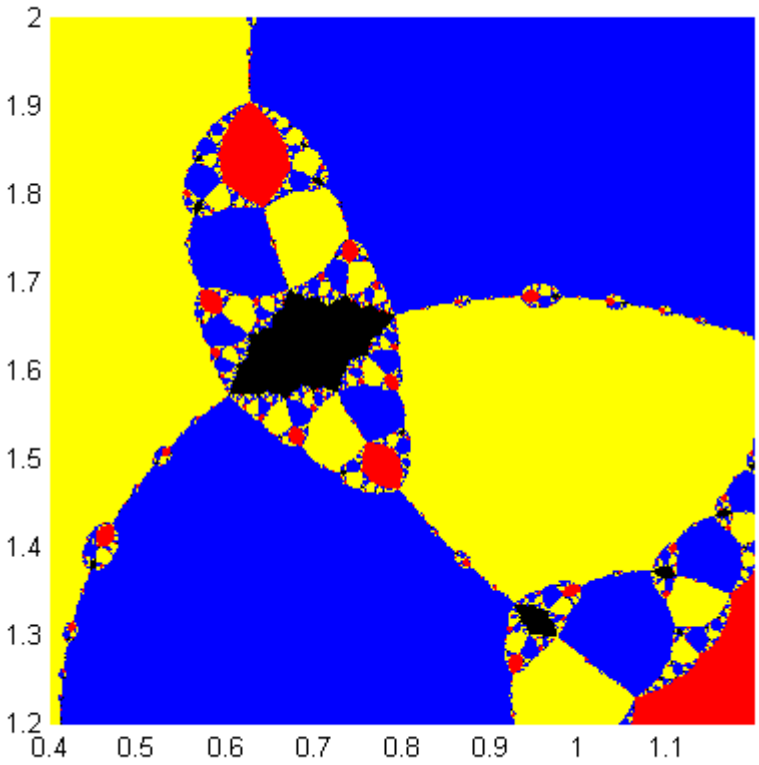
$z_2 = 0,2828 + 0,4692i$

Harmadik iterációs lépés:

$z_3 = -0,01 - 0,025i$

Negyedik iterációs lépés:

$z_4 = -0,5585 - 0,0013i$



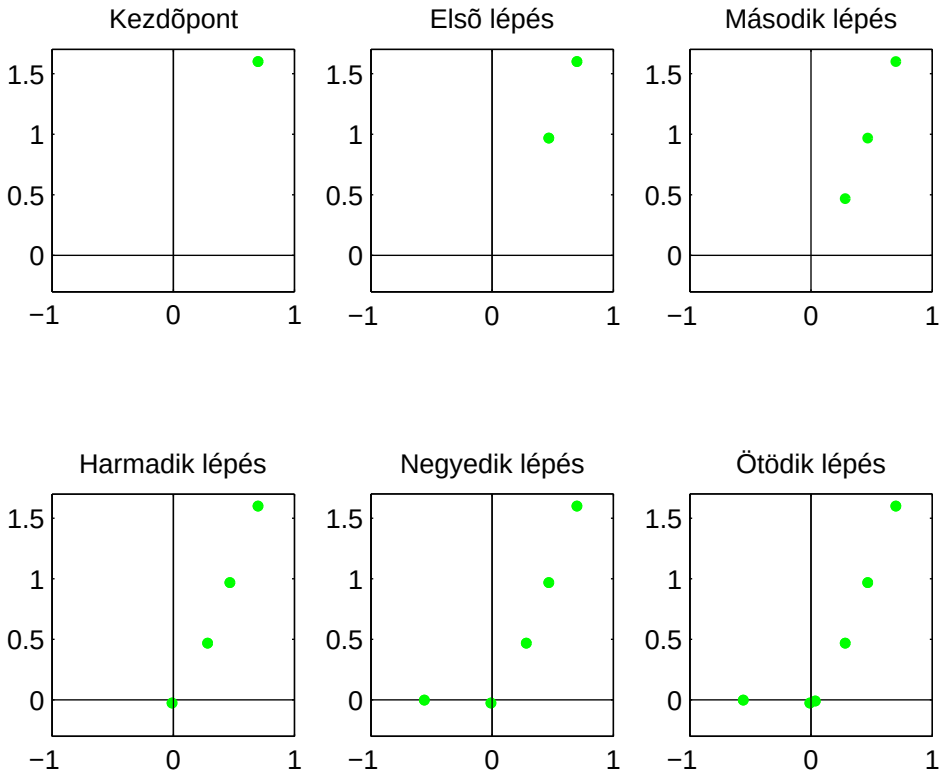
3.5. ábra. Az $f(z) = (z-1)(z+0,5-0,33i)(z+0,5+0,33i)$ függvény megoldása Newton-módszerrel. Sárga kezdőpontokból $-0,5+0,33i$, kék kezdőpontokból $-0,5-0,33i$, piros kezdőpontokból 1 gyökhöz konvergál az iteráció. A fekete kezdőpontokból 40 iterációs lépés alatt nem konvergál.

Ötödik iterációs lépés:
$z_5 = 0,0355 - 0,0088i$

Látható, hogy a módszer során az első két lépésben egy-egy következő fekete foltba jut az iteráció, majd ezután a valós tengely közelében kezdenek el ugrálni a pontok. Hogy jobban lássuk mi is történt nézzünk meg négy további iterációs lépést.

3.6. Példa. (3.7. ábra) Tekintsük az $f(z) = (z-1)(z+0,5-0,33i)(z+0,5+0,33i)$ függvényt. Indítsuk az iterációt az $z_5 = 0,0355 - 0,0088i$ pontból.

Hatodik iterációs lépés:



3.6. ábra. Az $f(z) = (z-1)(z+i)(z+1-i)$ függvény megoldása Newton-módszerrel. Iterációs lépések zölddel jelölve.

$z_6 = -0,563 + 0,0018i$

Hetedik iterációs lépés:

$z_7 = 0,0062 + 0,0112i$

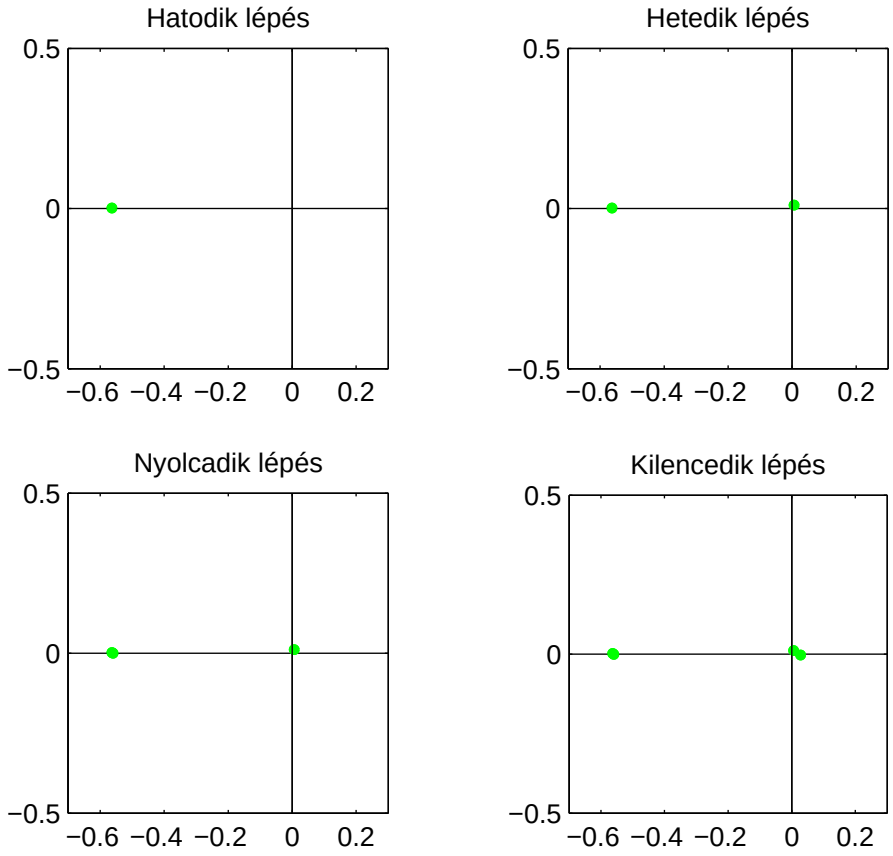
Nyolcadik iterációs lépés:

$z_8 = -0,5596 - 0,0004i$

Kilencedik iterációs lépés:

$z_9 = 0,0282 - 0,0026i$

Megfigyelhető, hogy a negyedik lépéstől kezdődően a $-0,5$ és 0 közelében felváltva



3.7. ábra. Az $f(z) = (z-1)(z+i)(z+1-i)$ függvény megoldása Newton-módszerrel. Iterációs lépések zölddel jelölve.

ugrál az iteráció.

Látható, hogy ezeknek a fekete részeknek az előzőeknél jóval nagyobb a kiterjedésük, így ezen belső pontokon nem segít a perturbálás, mert továbbra is fekete részben maradunk.

4. fejezet

Curt McMullen javított algoritmusa

4.1. Javított algoritmus

Curt McMullen módszere [McMullen 1987] szerint, ha a függvényünk a következő alakú:

$$f(z) = z^3 + a_1 \cdot z + a_0$$

akkor az $n+1$.-ik pont kiszámításának képlete:

$$z_{n+1} = z_n - \frac{(z_n^3 + a_1 z_n + a_0)(3a_1 z_n^2 + 9a_0 z_n - a_1^2)}{3a_1 z_n^4 + 18a_0 z_n^3 - 6a_1^2 z_n^2 - 6a_1 a_0 z_n - 9a_0^2 - a_1^3}$$

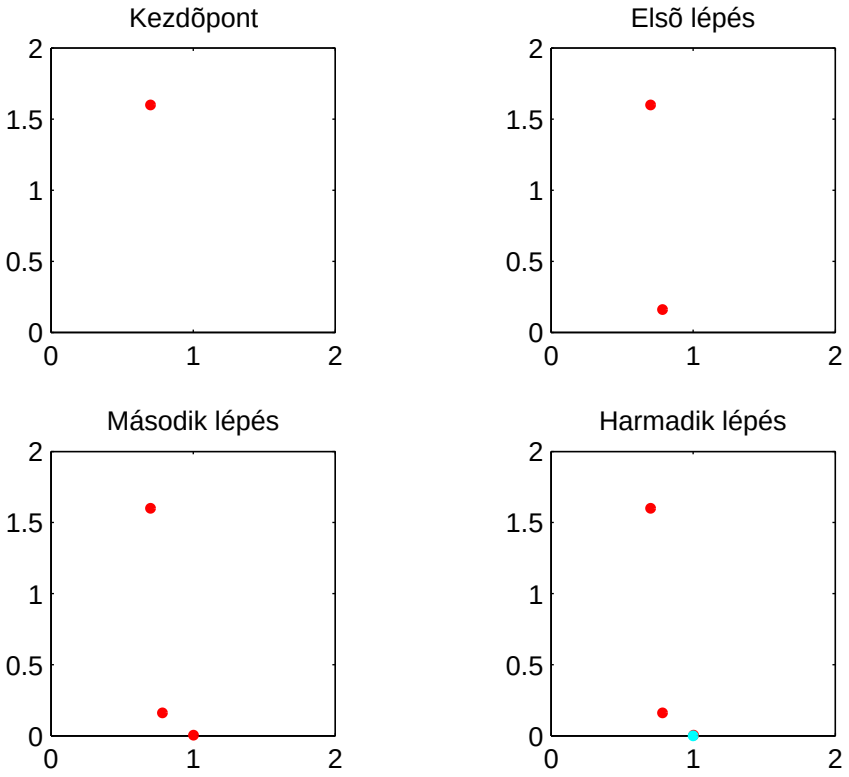
4.1. Megjegyzés. Általános harmadfokú polinomokra is létezik a módszer, de mivel minden harmadfokú polinom a fenti alakra hozható, ezért ezen hosszadalmas képlet közlésétől eltekintünk.

Nézzük meg most ezzel a módszerrel az előző fejezetben tárgyalt 3.4. Példát, melyben a Newton-módszer nem vezetett eredményre, mert a 3.4. ábrán látható fekete részekből még perturbáció segítségével sem sikerült az iterációnak gyököt találnia.

4.1. Példa. (4.1. ábra) Tekintsük az $f(z) = (z-1)(z+0,5-0,33i)(z+0,5+0,33i)$ függvényt. Indítsuk az iterációt az $z_0 = 0,7 + 1,6i$ pontból.

```
NewMeth(1,0,-0.6411,-0.3589,0.7+1.6i,40,'mcmullen')
```

Első iterációs lépés (4.2. ábra):



4.1. ábra. Az $f(z) = (z-1)(z+i)(z+1-i)$ függvény megoldása McMullen-módszerrel. Iterációs lépések pirossal, gyök ciánnal jelölve.

$z_0 = 0,7 + 1,6i$

Következő pont: $z_1 = 0,784 + 0,16i$

Második iterációs lépés:

$z_2 = 1,0029 + 0,0036i$

Harmadik iterációs lépés:

$z_3 = 1$

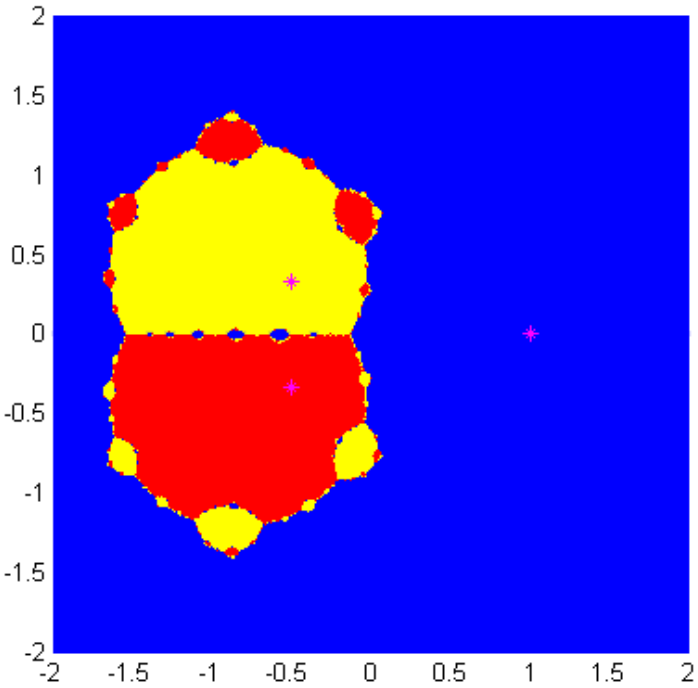
A gyök 1 (4.1. ábrán ciánnal jelölve), melyet a módszer a harmadik iterációs lépésben talál meg (minimum 4 tizedesjegy pontossággal).

4.2. McMullen-módszer különböző kezdőpontokból indítva

Az előző fejezethez hasonlóan most ezt a javított módszert is alkalmazzuk különböző kezdőpontokból.

4.2. Példa. (4.2. ábra) Tekintsük az $f(z) = (z-1)(z+i)(z+1-i)$ függvényt.

```
NewMeth(1,0,-0.6411,-0.3589,testarea(-2,2,-2,2,100),40,'mcmullen')
```



4.2. ábra. Az $f(z) = (z-1)(z+i)(z+1-i)$ függvény megoldása McMullen-módszerrel. Sárga kezdőpontokból $-0,5+0,33i$, piros kezdőpontokból $-0,5-0,33i$, kék kezdőpontokból 1 gyökhöz konvergál az iteráció

A 4.2. ábrán ismét látható a már megszokott három szín, valamint a fekete, azaz a nem konvergáló részek eltűnnek, eszerint tehát McMullen algoritmusa minden kezdőpontból indítva talál gyököt.

5. fejezet

Összefoglalás

A dolgozatban áttekintettük a Newton-Raphson módszernek nevezett approximációs eljárást, mellyel harmadfokú polinomok gyökeit kerestük. Valós együtthatós polinomok esetén bebizonyítottuk, hogy az eljárás kvadratikusan konvergál (2.1. Tétel), mely kvadratikus konvergenciát példán is szemléltettük. Ezek után azokat a speciális eseteket vizsgáltuk, amikor a Newton-módszer nem találta meg a gyököt. Ekkor, ha a kiválasztott kezdőpont a leképezés egy fixpontja volt és ciklusba jutott a módszer (2.2. Példa), vagy ha a kezdőpontban a polinom deriváltja 0 volt (2.3. Példa), a perturbálás segített és a módszer talált gyököt. Azonban ha a polinomnak nem létezett valós gyöke (2.4. Példa), akkor ez a fajta perturbáció nem javított az iteráción, továbbra sem találtunk gyököt. Ezen a problémán csak a 3. fejezetben bemutatott képzetes rész hozzáadás a valós kezdőponthoz segített (3.1. Példa). Ekkor a Newton-módszer talált komplex gyököt.

Ezek után a valós együtthatós polinomokról áttértünk a komplex együtthatókra, amikor is a Newton-módszer az előzőekhez hasonlóan működött. A következő példában különböző komplex kezdőpontokból indítottuk az iterációt, így kaptuk a különböző színes ábrákat. Ezeken az ábrákon különböző színnel jelöltük a különböző gyökhöz konvergáló kezdőpontokat. Az ábrákon (3.4. ábra) megfigyelt nagy kiterjedésű fekete csomók azt jelezték, hogy ezeknél a belső pontokkal rendelkező részeknél nem konvergál az iteráció egyik gyökhöz sem. Ilyenkor a már korábban bevált perturbáció sem segített, mert perturbálás után is fekete részben maradtunk. Tüzetesebben megvizsgáltuk mi történik, ha egy ilyen kezdőpontból indítjuk az iterációt

(3.5. Példa), majd azt tapasztaltuk, hogy egyik fekete csomóból egy másikba ugrunk, míg nem két érték közelében kezd el oszcillálni a módszer.

A 4. fejezetben bemutattuk Curt McMullen javított algoritmusát, amelyik módszer harmadfokú polinomok esetében minden kezdőpontból talál gyököt. Ezzel az eljárással is megvizsgáltuk a korábban tárgyalt polinomokat és ezen esetekben mindig gyorsabb konvergenciát tapasztaltunk, mint a Newton-módszer esetében.

Összességében elmondható, hogy mind a Newton-módszer mind McMullen módszere egy kiváló eljárást ad harmadfokú polinomok gyökeinek megtalálására.

A. Függelék

A `NewMeth` program

A NewMeth program letölthető az ELTE TTK Matematika intézetének honlapjáról (`http://www.cs.elte.hu/~keppabt/documents/NewMeth.zip`).

A.1. A feladat

A `NewMeth` program a `MatLab` programcsomaggal készült, célja komplex együtthatós, legfeljebb harmadfokú polinomok gyökeinek megkeresése és azok ábrázolása a komplex számsíkon. A gyökök keresésére a program két matematikai eljárást, a Newton-Raphson módszert, és egy Curt McMullen által módosított módszert használ. A különböző gyökhöz konvergáló pontokat különböző színnel színezi.

A.2. Elemzés

- A program során lehetőségünk van beállítani kezdőpontokat és a maximális iterációs számot is (ha ezt a számot meghaladja az iteráció, ekkor az adott pontokat feketére színezi a program).
- A kapott adatokat egy mátrixban tároljuk melynek első oszlopában a pontok, második oszlopában pedig a pontokhoz tartozó gyökök találhatóak. A hatalmas memória igény miatt ezt a tömböt egy fájlba írjuk.
- Az iterációs lépést, a fájlba való kiírást, az onnan való beolvasást és képernyőre való megfelelő kirajzolást különböző alprogramok segítségével hajtjuk végre.

A.3. Input paraméterek

A program futatásához az alábbi input paraméterekre van szükségünk:

- **a3** - polinom együtthatója (szokásos jelölés) [komplex]
- **a2** - polinom együtthatója [komplex]
- **a1** - polinom együtthatója [komplex]
- **a0** - polinom együtthatója [komplex]
- **x0** - kezdőpont(ok) [komplex vektor]
- **maxit** - maximális iteráció szám [pozitív egész]. Ha ennyi lépés alatt nem találja meg a gyököt akkor a program feketével színezi az adott pontokat.
- **meth** - használni kívánt módszer neve [karakterlánc] - értéke ['newton' 'mcmullen']
- **marker_size** (opcionális) - ábrázolás során használt pontok mérete [pozitív egész]. Alapértelmezett érték: 1
- **filen** (opcionális) - az adatok tárolására használt fájl neve [karakterlánc]. Alapértelmezett érték: `'tempdata.txt'`

A.4. Implementáció

A `NewMeth.m` főprogram feladata az input paraméterek ellenőrzése és további alprogramok futtatása.

A főprogramban található alprogramok:

- `make_title` - az adatokat tartalmazó fájl fejlécének elkészítése
  input paraméterek:
  - **K** - polinom együtthatóit tartalmazó vektor [komplex vektor]
  - **sum** - eddig vizsgált kezdőpontok száma [nem negatív egész]

  - **filen** - adatokat tartalmazó fájl neve [karakterlánc]

- `newton.m` - az iterációs lépést végrehajtó alprogram
  input paraméterek:

  - **a3** - polinom együtthatója (szokásos jelölés) [komplex]
  - **a2** - polinom együtthatója [komplex]
  - **a1** - polinom együtthatója [komplex]
  - **a0** - polinom együtthatója [komplex]
  - **x0** - kezdőpont(ok) [komplex vektor]
  - **k** - kezdőpont száma [pozitív egész]
  - **x0** - kezdőpontokat tartalmazó vektor [komplex vektor]
  - **maxit** - maximális iteráció szám [pozitív egész]
  - **meth** - használni kívánt módszer neve [karakterlánc] - értéke ['newton' 'mcmullen']

  output paraméterek:

  - **A** - pontokat tartalmazó vektor [komplex vektor]
  - **kon** - konvergál-e az adott kezdőpontból [0 nem tudjuk; 1 igen; 2 nem]

- `roots.m` - az
  input paraméterek:

  - **A** - pontokat tartalmazó vektor [komplex vektor]
  - **kon** - konvergál-e az adott kezdőpontból [0 nem tudjuk; 1 igen; 2 nem]
  - **V** - eddig megtalált különböző gyököket tartalmazó vektor [komplex vektor]

  output paraméter:

  - **V** - eddig megtalált különböző gyököket tartalmazó vektor [komplex vektor]

- `writting_to_file.m` - adatok kiírása egy fájlba
  input paraméterek:
  - **A** - pontokat tartalmazó vektor [komplex vektor]
  - **V** - eddig megtalált különböző gyököket tartalmazó vektor [komplex vektor]
  - **sum** - eddig vizsgált kezdőpontok száma [nem negatív egész]
  - **kon** - konvergál-e az adott kezdőpontból [0 nem tudjuk; 1 igen; 2 nem]
  - **fid** - fájl azonosítója
- `plotting_the_datas.m` - kapott pontok kirajzolása a komplex számsíkon a megfelelő módon
  input paraméterek:
  - **filen** - adatokat tartalmazó fájl neve [karakterlánc]
  - **marker_size** - ábrázolás során használt pontok mérete [pozitív egész]

A.4.1. További alprogram

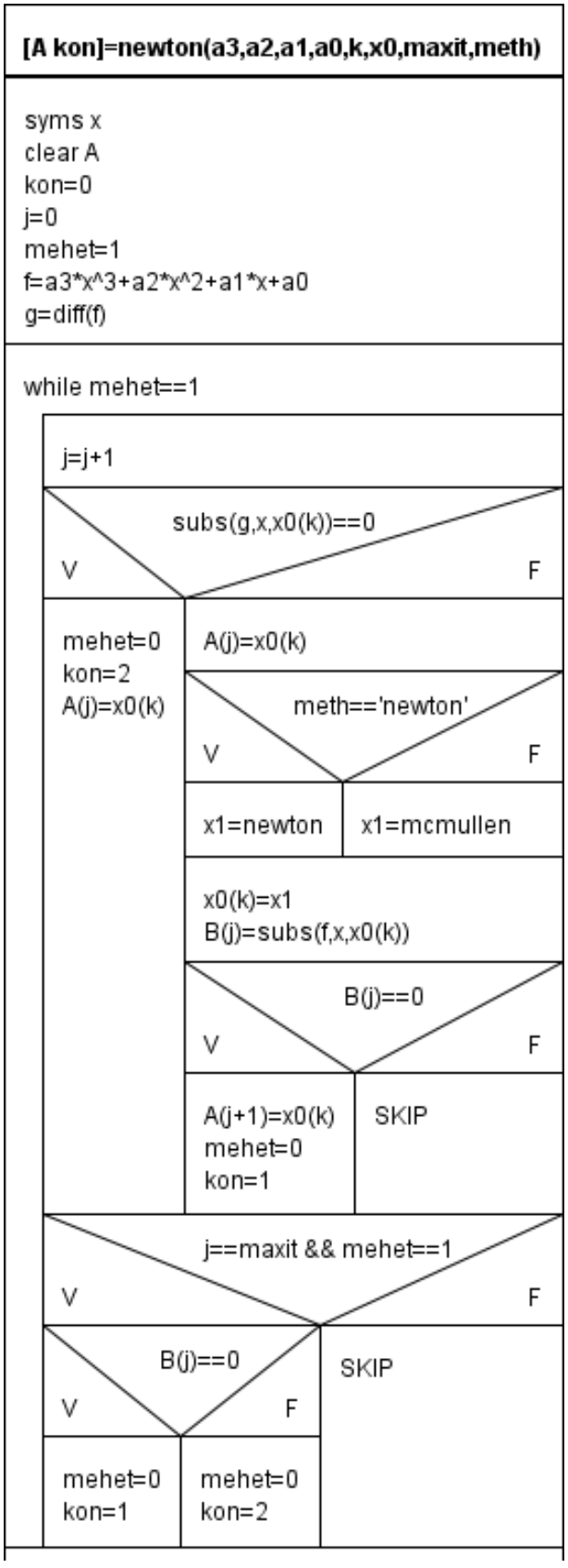
További alprogram a `testarea.m`, amely program egy téglalap alakú területen, adott távolságokként lévő pontokat helyezi el egy vektorban. Ezek a pontok lesznek az iteráció során használt kezdőpontok.

- `testarea.m` - kezdőpontok elkészítése
  input paraméterek:
  - **xmin** - x tengelyen vett baloldali végpont [pozitív valós]
  - **xmax** - x tengelyen vett jobboldali végpont [pozitív valós]
  - **ymin** - y tengelyen vett baloldali végpont [pozitív valós]
  - **ymax** - y tengelyen vett jobboldali végpont [pozitív valós]
  - **l** - egységnyi intervallumon lévő pontok száma [pozitív egész]

  output paraméter:
  - **A** - kezdőpontok [komplex vektor]

A.5. Absztrakt program

A program lényegi algoritmusa az iterációs lépést végrehajtó program (`newton.m`). Ezen program struktogramját ábrázolja a A.2. ábra.



A.1. ábra. `newton.m` alprogram struktogramja

Irodalomjegyzék

[1] McMullen, Curt *Families of Rational Maps and Iterative Root-Finding Algorithms*, Annals of Mathematics 125 (1987), 467-493.

[2] Gilbert, William J. *Generalizations of Newton's method*, Fractals, Vol 9., No. 3 (2001), 251-262.