B3053

PLAZMIDSTABILITÁSSAL KAPCSOLATOS MATEMATIKAI MODELLEK ÉS NUMERIKUS ELEMZÉSÜK

Baranyi József

Gépipari és Automatizálási Műszaki Főiskola

EGYETEMI DOKTORI ÉRTEKEZÉS

1989.

B 3053

Előszó

A közönséges differenciálegyenletek numerikus megoldásának elméleti alapjaival dr. Balla Katalin kandidátus irányításával ismerkedtem meg. Ezúton is köszönöm, hogy vállalta tudományos ösztöndíjam témavezetését.

Az alkalmazott matematikus számára oly fontos, a dolgozatban elemzett inspiráló problémák először dr. Ballagi-P. Andrással folytatott együttműködésünk során kerültek felszínre. Hálával gondolok ezért barátomra.

A dolgozat nagy része az Uppsalai Egyetem Mikrobiológiai Tanszékén eltöltött 6 hónap alatt készült. Köszönöm dr. Tibor Illeninek, hogy munkatársául választott. A munkát támogatta a National Swedish Board for Technical Development (Nr. 732-88-01813).

Végül, de nem utolsósorban köszönettel tartozom munkatársaimnak a Gépipari és Automatizálási Műszaki Főiskolán, hogy feladataim egy részének átvállalásával időt biztosítottak kutatómunkámra.

Kecskemét, 1989. július 18.

Baranyi József

Baranyi József

Rövid tartalom

A dolgozat a biotechnológiában jólismert, plazmidstabilitásnak nevezett problémakör néhány matematikai modelljét és azok numerikus matematikai kérdéseit vizsgálja.

Az első rész röviden vázolja a szükséges biológiai, biotechnológiai ismereteket, majd bemutatja a dolgozatban fontos szerepet játszó, è tudományterületen közismert Monod-féle differenciálegyenleteket.

A második rész az ún. pozitív és negatív sejtek szaporodásának és "egymásbaalakulásá"-nak egy differenciaegyenletre vezető leírásával foglalkozik. Kiindulási alapunk Kim és Ryu (1984) modellje, amelyet azzal módosítunk, hogy figyelembe vesszük azt a lehetőséget is, hogy - bár nagyon kicsi valószínűséggel - a negatív sejtek is visszaalakulhatnak pozitívokká. A kapott általánosabb modell és a rávonatkozó megállapítások jelen értekezés szerzőjének még publikálatlan, új eredményei.

a.) Bemutatjuk, hogy így valóban Kim és Ryu modelljének általánosításához jutunk.

b.) Formulát vezetünk le a plazmidstabilitásra jellemző hányadosnak (negatív és pozitív sejtek aránya) időbeni alakulására illetve asszimptotikus viselkedésére. Bevezetjük a "saját növekedési faktor" fogalmát és megmutatjuk, hogy ha δ jelöli a negatív illetve pozitív sejtek saját növekedési faktorának különbségét, P pedig azt az értéket amihez a plazmidstabilitást jellemző hányados időben tart,

akkor a $\delta \to P$ függvénynek $\delta=0$-nál Kim és Ryu modelljében szakadása van, ellentétben a mi általánosabb modellünk megfelelő $\delta \to P$ folytonos függvényével.

c.) Numerikus becslést adunk δ azon értékére, amely mellett a pozitív és a negatív sejtek között hozzávetőleges egyensúly alakul ki ($P \approx 1$).

A harmadik rész ismerteti Monod differenciál-egyenleteinek azt az - irodalomban egyre terjedő - általánosítását, amely megkülönbözteti a pozitív és a negatív sejteket (Imanaka-Aiba (1981), Parulekar et al. (1987)). Megmutatjuk, hogy a Kim-Ryu féle illetve az általánosított Monod (más néven Monod-Aiba) modellben a szegregációt jellemző paraméterek nem ekvivalensek és gyakorlatban is használható becslést adunk a kétféle szegregációs paraméter numerikus eltérésére; ez szintén még publikálatlan, új eredmény.

Megállapítjuk, hogy a Monod egyenletek (így a Monod-Aiba egyenletek is) stiff közönséges differenciálegyenletek (KDE-k), s a negyedik rész a stiff KDE-k numerikus megoldásának elméleti alapjait tekinti át.

Az ötödik részben a Monod-Aiba modell szimulációjához egy, a Rosenbrock módszerek családjába tartozó algoritmust javasolunk (Kaps-Rentrop (1979)). Az eljárás lépésvezérlése - kis módosítással - Gottwald-Wanner (1981) technikájára vezethető vissza. Tesztfeladaton mutatjuk be, hogy programunk megbízhatóan működik ott is, ahol más, az irodalomból ismert program nem teljesíti az előírt pontosságot. Demonstráljuk az alkalmazott "visszalépéses" technika működését, végül bemutatjuk a Monod-Aiba modell kapott szimulációs görbéit.

A további matematikai vizsgálatok néhány lehetséges irányát vázoljuk befejezésképp. A plazmideloszlás függelékben ismertetett modellje alapján arra az eredményre jutunk, hogy érdemes a Monod-Aiba modell szegregációs rátáját $\sigma=e^{-\alpha N}$ alakban feltételeznünk, ahol N a sejtekben lévő átlagos plazmid-kópiaszám, α pedig modellparaméter.

A függelék egy közlésre még el nem fogadott dolgozat, amelynek tételei jelen értekezés szerzőjétől származnak. Ebben a plazmidok replikációjára és partíciójára egy sztochasztikus modellt ismertetünk, amely - mint erre ott példát is adunk - explicit eredményei miatt alkalmas arra, hogy más, hasonló modellek megfelelő paramétereit megbecsüljük.

TARTALOMJEGYZÉK

	old.
Bevezetés	1
1. Biotechnológiai alapok	3
1.1. Plazmidstabilitás	3
1.2. Sejtpopuláció folytonos tenyészetben; a klasszikus Monod-modell	4
2. Egy differenciaegyenlet-modell elemzése	7
2.1. A modell felállítása	7
2.2. A differenciaegyenlet megoldása	9
2.3. Az eredmények elemzése	14
2.4. A differenciaegyenlet-modell alkalmazhatósága	17
3. A Monod-Aiba modell	18
3.1. A modell ismertetése	18
3.2. Összehasonlítás a differenciaegyenlet-modellel	19
3.3. A Monod-egyenletek és a stiff KDE-k	23
4. A stiff KDE-k numerikus megoldásának alapjai	26
4.1. KDE-k numerikus megoldásának alapjai	26
4.2. Stiff KDE-k és A-stabilitás	29
4.3. A Rosenbrock típusú módszerek	31
5. A ROW4 eljárás alkalmazása	33
5.1. A ROW4B algoritmus	33
5.2. A ROW4B FORTRAN szubrutin	36
5.3. ROW4B tesztelése	42
5.4. A differenciálegyenlet-modell szimulációja a ROW4B programmal	47
6. Továbbfejlesztési lehetőségek	51
IRODALOM	
FÜGGELÉK	

Bevezetés

A biotechnológia az elmúlt étvizedekben empirikus tudományból fokozatosan egyre egzaktabb tudománnyá válik, ennek következtében egyre jobban igényli és felhasználja a matematika lehetőségeit. A matematika, ezen belül is a közönséges differenciálegyenletek (KDE) numerikus megoldása sok motiváló problémát kapott az utóbbi években biotechnológusoktól, vegyészektől. Példának elég megemlíteni, hogy a stiff KDE-k megoldási algoritmusainak fejlődésében nagy szerepe volt azoknak az enzimkinetikai problémáknak, amelyeket Enright et al. (1975) is felvett a 25 probléma közé, amelyeket a stiff KDE-k megoldási algoritmusainak teszteléséhez javasolt; hasonlóan az A-stabilitás fogalmának bevezetője, E. Dahlquist is vizsgált munkatársaival biotechnológiában felmerült problémákat tesztfeladatai között (Bjurel et al. 1971). Az újabb és újabb problémák újabb és újabb algoritmusok, illetve az ismert módszerek módosítását, finomítását követelik meg, s ez a gyümölcsöző együttmüködés az alkalmazott matematikával különösen jelentőssé vált azzal, hogy a személyi számítógép mindennapivá válása nyomán lehetőség nyílt igen kifinomult számítási eljárások kidolgozására egyedi esetekben is.

Ebben a folyamatban az alkalmazott matematikus feladata kettős: egyrészt a biológiai, technológiai problémák megértése után a matematikai modell alkotása, illetve régi modellek finomítása, módosítása; másrészt a matematikai modell numerikus elemzése és hatékony számítási eljárások kimunkálása.

Jelen dolgozat is ezt a felépítést követi: először ismertetjük azokat a biológiai, technológiai alapokat, amelyek elengedhetetlenül szükségesek a matematikai probléma megfogalmazásához, majd vizsgálatokat végzünk lehetséges modellekkel kapcsolatban. Végül elméleti alapokkal támogatva megkíséreljük kiválasztani a legmegfelelőbb numerikus módszert a modell számítógépes szimulációjához.

1. Biotechnológiai alapok

1.1. Plazmidstabilitás

A "biotechnológiai gyár" alapját a kívánt géntermék előállítására képes sejtek képezik.

A baktériumok és az élesztők - amelyek a termelés leggyakrabban alkalmazott alanyai - a mesterséges, génsebészeti úton bejuttatott genetikai információt általában a bennük levő plazmidokon hordozzák. A plazmidok a kromoszómákhoz viszonyítva kicsiny DNS molekulák, amelyek a gazdasejttől többé-kevésbé független, önálló replikációra képesek; számuk egy sejtben egytől ezres nagyságrendig is terjedhet.

Sejtosztódásnál többé-kevésbé véletlenül előfordulhat, hogy a két leánysejt közül az egyikbe nem jut plazmid; szegregáció történik. A plazmidmentes, ún. negatív sejt természetesen az "előírt" géntermék előállítására nem képes, viszont ugyanúgy fogyaszt szubsztrátot, szaporodik (sőt általában nagyobb sebességgel), mint a plazmidhordozó, pozitív sejt. A plazmidstabilitási vizsgálatok arra irányulnak, miként lehetne megakadályozni, hogy a termelés számára értéktelen negatív sejtek kiszorítsák a pozitív sejteket. A folyamat megismeréséhez és az azt befolyásoló paraméterek szerepének feltárásához elengedhetetlen a jelenség matematikai leírása.

A negatív sejtek dominánssá válásának tehát két fő oka van:

1./ Szegregáció a sejtosztódásnál. A szegregáció valószínüsége kicsiny (10^{-3} nagyságrendű vagy kisebb), mégsem elhanyagolható.

2./ A negatív sejtek általában gyorsabban nőnek mint a pozitívok. Specifikus növekedési sebességük a pozitívokénak 3-szorosát is elérheti.

A dolgozatban a fent leírt, plazmidstabilitásnak nevezett problémakör néhány matematikai kérdését elemezzük. Ehhez szükségünk van még a klasszikus Monod-modell ismertetésére, amelyre a következőkben térünk ki.

1.2 Sejtpopuláció folytonos tenyészetben; a klasszikus Monod-modell

A folytonosan tenyésztett homogén sejtpopulációról a legismertebb matematikai leírás az ún. Monod-modell. Eszerint a folyamatot - itt nem részletezett egyszerűsítő feltételezések mellett - egy bioreaktorban két mennyiség jellemzi: az $x(t)$ sejtkoncentráció és az ún. limitáló szubsztrát $S(t)$ koncentrációja.

A reaktorba kívülről szabályozható D intenzitással (neve: hígítási sebesség) és S_{in} koncentrációval érkezik a szubsztrátoldat, amely a reaktorban lévő $S(t)$ koncentrációjú szubsztrátoldattal elkeveredik. A szubsztrátot a sejtek növekedésre használják fel. Ugyancsak D intenzitással folyik el a reaktorból a fermentlé, amelyben együtt van sejt és szubsztrát.

Tehát a sejtek az osztódás miatt egyrészt szaporodnak a reaktorban, másrészt csökkennek a kimosódás miatt. Jelöljük μ-vel a sejtnövekedés intenzitását (specifikus növekedési sebesség). Mind D mind pedig μ mértékegységét [1/h]-nak választjuk, míg a sejtkoncentráció mértékegysége

legyen [g/l]. (Feltételezzük, hogy minden sejtnek ugyanakkora a tömege, így a sejtszám és a sejttömeg között egyszerü lineáris kapcsolat van.) Tehát $x(t)$-re vonatkozó egyenletünk:

$$\frac{dx(t)}{dt} = \mu\, x(t) - D\, x(t) \qquad (1.1.1)$$

A reaktorban lévő $S(t)$ szubsztrátkoncentrációt növeli a D intenzitással befolyó szubsztrátoldat (koncentrációja: S_{in}); csökkenti a kimosódás (szintén D intenzitással), valamint a sejtek táplálása:

$$\frac{dS(t)}{dt} = D\,(\,S_{in} - S(t)\,) - \frac{\mu}{Y}\, x(t) \qquad (1.1.2)$$

ahol Y az úgynevezett "hozamkonstans": egységnyi tömegű szubsztrát elfogyasztásával Y egységnyit nő a sejttömeg.

Az (1.1.1) - (1.1.2) differenciálegyenlet rendszert az teszi nehézzé általános esetben, hogy a μ specifikus növekedési sebesség nem konstans; elsősorban a reaktorban uralkodó szubsztrátkoncentrációtól függ. E kapcsolat leírására leggyakrabban az ún. Michaelis-Menten kinetikát alkalmazzák:

$$\mu(S) = \mu_{max}\, \frac{S}{K_s + S} \qquad (1.1.3)$$

ahol μ_{max} a maximális specifikus növekedési sebesség, K_s pedig a telítési szubsztrátkoncentráció. Mindkettő a tenyésztési körülményektől (alkalmazott szubsztrát, baktériumtörzs, hőméréséklet, stb.) függő paraméter.

Hogy numerikus szempontból is képet kapjunk a Monod-féle differenciálegyenletekről, példát adunk a bennük szereplő paraméterek gyakorlatban előforduló értéktartományára:

D	:	0.05 - 1.5 [1/h]
S_{in}	:	2 - 50 [g/l]
μ_{max}	:	0.2 - 1.5 [1/h]
K_s	:	0.001-2 [g/l]

Jegyezzük meg rögtön, hogy a szubsztrátbetáplálás fajlagos sebessége (vagyis az általunk választható D hígítási sebesség) kisebb kell, hogy legyen mint μ_{max} , különben a sejtek kimosódnak.

A Monod- illetve Michaelis-Menten kinetikát igen sok bírálat érte az elmult években (Heineken et. al. (1967), Frederickson et al. (1970), Bailey-Ollis (1984)), mindazonáltal pontosabbat, mégis kellően egyszerűt nem sikerült még adni helyette. Ez az oka annak, hogy a modell tovább él és plazmidstabilitási problémák modellezésénél szintén Monod szolgáltatja a kiindulópontot az irodalomban (Lauffenburger (1985) Parulekar et al., (1987)).

2. Egy differenciaegyenlet-modell elemzése

2.1. A modell felállítása

Először szükségünk lesz az exponenciális növekedés egy tulajdonságára:

Legyen $f(t)$ exponenciálisan növő a $[t_0,\infty)$ intervallumban:

$$f(t)=f_0e^{\mu t} \qquad (f_0>0,\ \mu>1) \qquad (2.1.1)$$

Belátható, hogy ha $t_d = \ln 2/\mu$, akkor $f(t+t_d)=2f(t)$ teljesül minden $t \in [t_0,\infty)$ -re. Ez azt a közismert tényt fejezi ki, hogy az exponenciális növekedés duplázási ideje konstans. Ugyancsak egyszerü számolással adódik, hogy ha $\alpha > 0$ konstans, akkor

$$f(t+\alpha t_d) = 2^{\alpha}f(t) \qquad (2.1.2)$$

Modellünkben feltételezzük a baktériumtenyészet exponenciális növekedését. Legyen t_0 a kezdeti időpont, és legyen $t>t_0$-ra a tenyészet pozitív illetve negatív sejtjeinek száma $N^+(t)$ illetve $N^-(t)$. Legyen t_d^+ a pozitív, t_d^- a negatív sejtek duplázódási ideje (más néven: generációs idő), és tegyük fel, hogy ezek konstansok. Legyen t_d^+ időmérésünk alapja és így $k=0,1,2,\ldots$-re legyen

$$N_k^+ = N^+(\ t_0 + kt_d^+\ ) \qquad (2.1.3)$$

$$N_k^- = N^-(\ t_0 + kt_d^+\ ) \qquad (2.1.4)$$

Tegyük fel, hogy t_d^+ idő alatt a pozitív sejtek a-ad részében fordul elő szegregáció, ahol $a<1$ konstans. Ezt a következő öszefüggéssel írjuk le:

$$N_{k+1}^+ = aN_k^+ + 2(1-a)N_k^+ = (2-a)N_k^+ \qquad (2.1.5)$$

Vagyis feltételezésünk szerint a pozitív sejtek k-dik generációjának a-ad része csak egyik leánysejtjét közvetíti át a pozitív sejtek (k+1)-dik generációjába, a többi pozitív sejt pedig mindkét leánysejtet.

Az irodalomban megemlítik, hogy bár a negatív sejt is visszaalakulhat pozitívvá azáltal, hogy plazmidot vesz fel a sejtfalon keresztül, azonban ennek valószínűsége még a-hoz képest is elhanyagolható, így a modellekben ezt nem veszik figyelembe. Mi most mégis számolunk ezzel a visszaalakulással, hogy képet kapjunk az elhanyagolás jogosságáról.

Tegyük fel tehát, hogy egy negatív sejt a generációs idején belül b (konstans) valószínűséggel alakul vissza pozitív sejtté.(2.1.5)-höz analóg módon:

$$N^-(t_0+(k+1)t_d^-) = (2-b)\ N^-(t_0+kt_d^-) \qquad (2.1.6)$$

(2.1.2)-t figyelembe véve kapjuk:

$$N_{k+1}^- = aN_k^+ + (2-b)^{\alpha}N_k^- \qquad (2.1.7)$$

ahol $\alpha = t_d^+/t_d^-$

Hogy (2.1.5)-öt is kiegészítsük a negatívból újra pozitívvá alakult sejtekkel, vegyük számításba, hogy (2.1.2) alapján elvárható:

$$N_{k+1}^+ + N_{k+1}^- = 2N_k^+ + 2^{\alpha}N_k^- \qquad (2.1.8)$$

Tehát modellünk a következő differenciaegyenlettel írható le:

$$N^+_{k+1} = (2-a)N^+_k + (G-g)N^-_k \qquad (k=0,1,2,...) \qquad (2.1.9)$$

$$N^-_{k+1} = a\,N^+_k + g\,N^-_k \qquad (k=0,1,2,...) \qquad (2.1.10)$$

ahol $\alpha = t^+_d/t^-_d$, $G = 2^\alpha$, $g = (2-b)^\alpha$ és $0 \le b << a << 1$.

Ha $b = 0$ (vagyis $G = g = 2^\alpha$), akkor Kim és Ryu (1984) modelljét kapjuk vissza.

2.2 A differenciaegyenlet megoldása

a./ $b>0$

Vizsgáljuk először a $g = (2-b)^\alpha < 2^\alpha = G$ (vagyis $b>0$) esetet.

Bevezetve az $\underline{N}_k = [\,N^+_k\,,\,N^-_k\,]^T$ vektort (2.1.9) és (2.1.10) a következő alakba írható:

$$\underline{N}_{k+1} = A\,\underline{N}_k \qquad (k=0,1,2,...) \qquad (2.2.1)$$

ahol

$$A = \begin{bmatrix} (2-a) & (G-g) \\ a & g \end{bmatrix} \qquad (2.2.2)$$

Nyilvánvalóan

$$\underline{N}_k = A^k\,\underline{N}_0 \qquad (k=0,1,2,...) \qquad (2.2.3)$$

Ellenőrizhető, hogy A pozitív definit ha $0 < b < a < 2g/G$ amely relációk esetünkben természetesen teljesülnek. Így a sajátértékek is pozitívok.

$$0 < v = \frac{(g+2-a)}{2} + \sqrt{\left(\frac{g-2+a}{2}\right)^2 + a(G-g)} \qquad (2.2.4)$$

$$0 < w = \frac{(g+2-a)}{2} - \sqrt{\left(\frac{g-2+a}{2}\right)^2 + a(G-g)} \qquad (2.2.5)$$

Definiáljuk a T mátrixot a következőképpen:

$$T = \begin{bmatrix} (v-g) & (w-g) \\ a & a \end{bmatrix} \qquad (2.2.6)$$

Belátható, hogy mivel $v \neq w$, így T nemszinguláris.

A számítások megkönnyítésére vezessük be az

$$M = \begin{bmatrix} c & 0 \\ 0 & d \end{bmatrix} \qquad (2.2.7)$$

segédmátrixot és számítsuk ki M-nek a következő függvényét:

$$F(M) = \frac{1}{a(v-w)} \begin{bmatrix} (v-g) & (w-g) \\ a & a \end{bmatrix} \begin{bmatrix} c & 0 \\ 0 & d \end{bmatrix} \begin{bmatrix} a & (w-g) \\ -a & (v-g) \end{bmatrix} \qquad (2.2.8)$$

Felhasználva a $(v-g)(w-g) = -a(G-g)$ összefüggést belátható, hogy

$$F(M) = \frac{1}{a(v-w)} \begin{bmatrix} cv-dw-g(c-d) & (G-g)(c-d) \\ a(c-d) & dv-cw+g(c-d) \end{bmatrix} \qquad (2.2.9)$$

Ez utóbbi egyenlőséggel bizonyítjuk, hogy

i)
$$T^{-1} = \frac{1}{a(v-w)} \begin{bmatrix} a & -(w-g) \\ -a & (v-g) \end{bmatrix} \tag{2.2.10}$$

(helyettesítsünk be c=d=1 -et (2.2.8) illetve (2.2.9)-be.)

ii)
$$A = T \begin{bmatrix} v & 0 \\ 0 & w \end{bmatrix} T^{-1} \tag{2.2.11}$$

(számoljunk c=v, d=w helyettesítési értékekkel (2.2.8), (2.2.9)-ben.

Ezek szerint T olyan nemszinguláris mátrix, amely az A-t Jordan alakjára transzformálja: $A = TJT^{-1}$, ahol $J = \mathrm{diag}(v,w)$

iii)
$$A^k = F(J^k) = \frac{1}{v-w} \begin{bmatrix} v^{k+1}-w^{k+1}-g(v^k-w^k) & (G-g)(v^k-w^k) \\ a(v^k-w^k) & vw^k-wv^k+g(v^k-w^k) \end{bmatrix} \tag{2.2.12}$$

(számoljunk $c=v^k$, $d=w^k$ helyettesítési értékekkel (2.2.8), (2.2.9)-ben, és vegyük figyelembe, hogy $A^k = TJ^kT^{-1}$.

(2.2.3)-ból és (2.2.12)-ből :

$$N_k^+ = \frac{1}{(v-w)} \left[\left(v^{k+1}-w^{k+1}-g(v^k-w^k)\right)N_0^+ + \left((G-g)(v^k-w^k)\right)N_0^- \right] \tag{2.2.13}$$

$$N_k^- = \frac{1}{(v-w)} \left[\left(a(v^k-w^k)\right)N_0^+ + \left(vw^k-wv^k+g(v^k-w^k)\right)N_0^- \right] \tag{2.2.14}$$

A plazmidstabilitásra jellemző $P_k = N_k^-/N_k^+$ mennyiségre kapjuk:

$$P_k = \frac{a(v^k-w^k)+\left[vw^k-wv^k+g(v^k-w^k)\right]P_0}{v^{k+1}-w^{k+1}-g(v^k-w^k)+(G-g)(v^k-w^k)P_0} =$$

$$= \frac{a\left(1-\left[\frac{w}{v}\right]^k\right)+\left(v\left[\frac{w}{v}\right]^k-w+g\left(1-\left[\frac{w}{v}\right]^k\right)\right)P_0}{v-\left[\frac{w}{v}\right]^k-g\left(1-\left[\frac{w}{v}\right]^k\right)+(G-g)\left(1-\left[\frac{w}{v}\right]^k\right)P_0} \qquad (2.2.15)$$

Mivel $0 < w < v$, így ha $k \to \infty$ akkor

$$P_k \to P = \frac{a+(v-w+g)P_0}{v-g+(G-g)P_0} \qquad (2.2.16)$$

A gyakorlatban oltásnál $(t=t_0)$ még nincsenek negatív sejtek, tehát $P_0=0$-val számolhatunk:

$$P = \frac{a}{\sqrt{\left(\frac{g-2+a}{2}\right)^2+a(G-g)}-\frac{g-2+a}{2}} \qquad (2.2.17)$$

b./ $b=0$

Tekintsük most a $g=(2-b)^\alpha = 2^\alpha=G$ $(b=0)$ esetet.

i) Ha $G=g=2-a$ akkor a fenti meggondolások nem alkalmazhatók mert a két sajátérték egyenlő és így a T transzformációs mátrix szinguláris lesz.

Ekkor azonban közvetlenül levezethető:

$$N_{k+1}^{+} = GN_{k}^{+} \tag{2.2.18}$$

$$N_{k+1}^{-} = aN_{k}^{+} + GN_{k}^{-} \tag{2.2.19}$$

és így:

$$N_{k}^{+} = G^{k}N_{0}^{+} \tag{2.2.20}$$

$$N_{k}^{-} = kaG^{k-1}N_{0}^{+} + G^{k}N_{0}^{-} \tag{2.2.21}$$

amiből nyerjük:

$$P_{k} = P_{o} + k(a/G) \quad (k=0,1,2,...) \tag{2.2.22}$$

ii) Ha $b=0$, $G=g\neq 2-a$, akkor a korábbi megfontolások érvényben maradnak.

Ha $G>2-a$, akkor a sajátértékek: $v=G$, $w=2-a$. Behelyettesítve (2.2.13)-(2.2.14)-be Kim és Ryu eredményét speciális esetként kapjuk:

$$N_{k}^{+} = (2-a)^{k}N_{0}^{+} \tag{2.2.23}$$

$$N_{k}^{-} = \frac{a}{g-2+a}\left[G^{k}-(2-a)^{k}\right]N_{0}^{+} + G^{k}N_{0}^{-} \tag{2.2.24}$$

$$P_{k} = \left(\frac{G}{2-a}\right)^{k}P_{0} + \frac{a}{G+a-2}\left[\left(\frac{G}{2-a}\right)^{k} - 1\right] \tag{2.2.25}$$

Ebből kitűnik, hogy $P_k \to \infty$ $(k \to \infty)$, ami összevág azzal, hogy ha a $v=g$, $P_0=0$ helyettesítéssel élünk (2.2.16)-ban akkor $a/0$ alakú határértékhez jutunk.

Ha $G<2-a$ akkor $v=2-a$, $w=G$. Ezt behelyettesítve (2.2.12)-be, ugyancsak (2.2.23)-(2.2.25)-öt kapjuk. $P_0=0$ mellett (2.2.25)-ben $P_k \to P=a/(2-a-G)$, s ugyanezt az eredményt kapjuk (2.2.16)-ból is.

2.3. Az eredmények elemzése

Számításaink eredményét a - $b>0$ esetet tükröző - (2.2.17) és a - $b=0$ melletti - (2.2.22), (2.2.25) képletek foglalják össze. Tekintve, hogy $b=0$ mellett $g=(2-b)^{\alpha}=2^{\alpha}=G$, így a $b=0$ és $b>0$ eseteket együtt tárgyalva megállapítható, hogy a $P_k=N_k^-/N_k^+$ plazmidstabilitást jellemző hányados asszimptotikus viselkedésében a döntő szerepet a $(g-2+a)$ mennyiség játssza.

Emlékezzünk rá, hogy $g=(2-b)^{\alpha}$ illetve $(2-a)$ azok a faktorok amelyekkel a negatív illetve pozitív sejtek az időmérés általunk választott alapja, t_d^+ alatt megnövelték saját alpopulációjukat. Nevezzük hát a fenti két mennyiséget a negatív illetve pozitív sejtek saját növekedési faktorának, és jelöljük különbségüket δ-val:

$$\delta = g+a-2 = (2-b)^{\alpha} - (2-a) \qquad (2.3.1)$$

a./ $b=0$

(2.2.25) szerint a $b=0$ esetben a generációk k számának növekedésével

ha $\delta>0$ akkor $P_k \to \infty$ exponenciálisan

ha $\delta=0$ akkor $P_k \to \infty$ lineárisan

ha $\delta<0$ akkor $P_k \to P=-a/\delta$

b./ $b>0$

Ekkor (2.2.17) átrendezve:

$$P = \sqrt{\left(\frac{\delta}{2(G-g)}\right)^2 + \frac{a}{G-g}} + \frac{\delta}{2(G-g)} \qquad (2.3.2)$$

Ebből kitűnik, hogy

ha $\delta>0$ akkor $P_k \to P > a/(G-g)$
ha $\delta=0$ akkor $P_k \to P = a/(G-g)$
ha $\delta<0$ akkor $P_k \to P < a/(G-g)$

Az $a/(G-g)$ mennyiség nagyságáról képet kaphatunk ha figyelembe vesszük, hogy b kicsi G és g-hez képest, így írható:

$$G-g = 2^{\alpha}-(2-b)^{\alpha} \approx b(\alpha 2^{\alpha-1}) \qquad (2.3.3)$$

vagyis P kritikus értéke (amikor $\delta=0$):

$$P_{crit} = \frac{a}{G-g} \approx \sqrt{\frac{2^{1-\alpha}}{\alpha}\,\frac{a}{b}} \qquad (2.3.4)$$

Minthogy a gyakorlatban α nincs túl messze az egységtől, így azt kaptuk, hogy $\delta=0$ esetben elég nagy generációszám után a tenyészetben körülbelül $\sqrt{a/b}$-szer annyi negatív sejt lesz mint pozitív; $\delta>0$ esetén ennél nagyobbnak $\delta<0$ esetén ennél kisebbnek várható a negatív sejtek aránya a pozitívokéhoz képest.

Hogy szemléletes képet kapjunk arról, mennyivel kell δ-nak kisebbnek lennie 0-nál hogy még használható tenyészetünk legyen, vizsgáljuk meg mikor kapunk

egyensúlyt a két sejtpopuláció között. A $P=1$ feltételből rendezés után kapjuk (2.2.17)-ből:

$$\delta = G - g - a \qquad (2.3.5)$$

Mint (2.3.3)-ból kitűnik $G-g$ és b nagyságrendje megegyezik; tehát szemléletesen azt mondhatjuk, hogy a negatív sejtek - legalábbis nagyságrendben - nem növik túl a pozitív sejteket, ha a saját növekedési faktorok közötti különbség (δ) és az "egymásba alakulás" faktorai közötti különbség ($a-b$) közelítőleg kompenzálják egymást:

$$\delta = (2-b)^{\alpha}-(2-a) \approx b-a \qquad (2.3.6)$$

Nyilvánvaló, hogy ez $0 < b << a$ miatt csak $\alpha<1$-re teljesülhet.

Ha valamilyen módon (pl. mesterséges beavatkozással) elérjük, hogy a negatív sejtek jóval lassabban nőjenek mint a pozitívok ($\delta<<-a$), akkor (2.3.2)-ből elsőrendű közelítéssel

$$P \approx -a/\delta$$

ugyanúgy mint a $b=0$ esetben. Ez a becslés is bizonyítja, hogy a $b>0$ lehetőség számbavételével mintegy folytonossá tettük azt a $\delta \to P$ függést, amelynek Kim és Ryu modelljében a $\delta=0$ helyen szakadása volt.

Érdemes megemlíteni azt az esetet, amikor $\alpha=1$. Ekkor ugyanis (2.2.17)-ből átrendezés után a

$$P = a/b$$

határérték adódik, amely szerint ha a két sejttípus azonos sebességgel szaporodik, akkor számuk aránya a generációk növekedtével az egymásba alakulás valószínűségeinek

arányához tart. Ez a mellékeredmény megegyezik más populációdinamikai modellek eredményével.

2.4. A differenciaegyenlet-modell alkalmazhatósága

A fenti modell egyszerübb változatát (b=0) Kim és Ryu (1984) vizsgálta azzal a különbséggel, hogy csak $\alpha=t_g^+/t_g^-$ állandóságát követelték meg; külön a t_g^+ és t_g^- generációs időkét nem Ez azt jelenti, hogy a sejtek növekedési sebessége (μ^+ ill. μ^-) nem feltétlen konstans, és így (2.1.2) nem használható fel annak igazolására, hogy a pozitív sejtek egy generációs ideje alatt a negatív sejtek $G = 2^{\alpha}$ -szorosukra növekedtek. Ezért kellett a jóval erősebb t_g^+ , t_g^- = const feltételezéssel élnünk. Ezzel viszont szűkítetettük modellünk érvényességi körét, s azt csak olyan esetekben alkalmazhatjuk, ahol a növekedési sebességek állandók. Ilyen elérhető például a turbidosztát technológiával ahol a sejtek maximális növekedési sebességgel szaporodnak, s így ez a technológia - mint azt Ballagi P. et al. elemzi - alkalmas lehet a szegregációs együttható kisérleti megállapítására a fenti modell segítségével.

3. A Monod-Aiba modell

3.1. A modell ismertetése

A szegregáció melletti baktériumnövekedés matematikai leírása Imanaka és Aiba (1981) munkájára nyúlik vissza. A modell azóta elterjedt az irodalomban és széles körben használják a Monod-modell szegregációval való módosításához hivatkozási alapként (Bailey és Ollis (1984), Parulekar et al. (1987) stb.). Ebben a kinetikai modellben az alap differenciálegyenletek a következők:

$$\frac{dx^+(t)}{dt} = (1-\sigma)\mu^+(S(t))x^+(t) - Dx^+(t) \qquad (3.1.1)$$

$$\frac{dx^-(t)}{dt} = \sigma\mu^+(S(t))x^+(t) + \mu^-(S(t))x^-(t) - Dx^-(t) \qquad (3.1.2)$$

$$\frac{dS(t)}{dt} = D(S_{in}-S(t)) - \frac{1}{Y^+}\mu^+(S(t))x^+(t) - \frac{1}{Y^-}\mu^-(S(t))x^-(t) \qquad (3.1.3)$$

ahol $\mu^+(S)$ és $\mu^-(S)$ eleget tesznek a Michaelis-Menten kinetikának:

$$\mu^+(S) = \mu^+_{max}\frac{S}{K^+_s + S} \qquad (3.1.4)$$

$$\mu^-(S) = \mu^-_{max}\frac{S}{K^-_s + S} \qquad (3.1.5)$$

A modell időfüggő változói:

- $x^+(t)$, $x^-(t)$: a pozitív ill. negatív sejtek koncentrációja
- $S(t)$: szubsztrátkoncentráció

Kivülről változtatható paraméterek:

- S_{in} : befolyó szubsztrát koncentrációja
- D : hígítási sebesség

Belső modellparaméterek:

- σ : szegregációs ráta
- μ^+_{max} , μ^-_{max} : pozitív ill. negatív sejtek maximális növekedési sebessége
- K^+_s , K^-_s : pozitív ill. negatív sejtek egyensúlyi szubszrátkoncentrációja
- Y^+ , Y^- : pozitív ill. negatív sejtek hozamkonstansai

3.2. Összehasonlítás a differenciaegyenlet-modellel

A fenti Monod-Aiba modellben elhanyagoljuk annak lehetőségét, hogy a negatív sejt is visszaalakulhat pozitívvá. Ettől eltekintve a fenti, Michaelis-Menten kinetikával ellátott modell nyilvánvalóan szélesebb körben alkalmazható, mint a 2. részben ismertetett differenciaegyenlet modell (nem követeljük meg μ^+ és μ^- állandóságát). Alkalmazzuk (3.1.1)-(3.1.3)-at az exponenciális növekedés fázisában $S \equiv 0$, $D \equiv 0$ mellett ("batch" fermentáció), és hasonlítsuk azt össze differenciaegyenlet-modellünkkel.

Ekkor $\mu^+(s) = \mu^+ = \text{const}$, $\mu^-(s) = \mu^- = \text{const}$, tehát:

$$\frac{d}{dt} x^+(t) = (1-\sigma)\mu^+ x^+(t) \tag{3.2.1}$$

$$\frac{d}{dt} x^-(t) = \sigma \mu^+ x^+(t) + \mu^- x^-(t) \tag{3.2.2}$$

A differenciálegyenlet rendszer megoldása:

$$x^+(t) = x^+(0)\, e^{(1-\sigma)\mu^+ t} \tag{3.2.3}$$

$$x^-(t) = x^+(0) \frac{\sigma\mu^+}{\mu^- - (1-\sigma)\mu^+} \left[e^{\mu^- t} - e^{(1-\sigma)\mu^+ t} \right] + x^-(0)\, e^{\mu^- t} \tag{3.2.4}$$

Bevezetve a plazmidstabilitásra jellemző $Q(t) = \frac{x^-(t)}{x^+(t)}$ hányadost, kapjuk (feltéve, hogy $x^+(t) \neq 0$):

$$Q(t) = \left[\frac{\sigma\mu^+}{\mu^- - (1-\sigma)\mu^+} + Q(0) \right] e^{(\mu^- - (1-\sigma)\mu^+)t} - \frac{\sigma\mu^+}{\mu^- - (1-\sigma)\mu^+} \tag{3.2.5}$$

Hasonlítsuk össze ezt az eredményt (2.2.25)-tel. Emlékeztetőül: ott k jelölte a pozitív sejtek generációinak számát, vagyis $t = t_0 + k t_d^+$. Időfüggővé téve (2.2.25)-öt, $t_0 = 0$ mellett kapjuk:

$$P(t) = \left(\frac{G}{2-a} \right)^{t/t_d^+} P(0) + \frac{a}{G+a-2} \left[\left(\frac{G}{2-a} \right)^{t/t_d^+} - 1 \right]$$

(3.2.6)

$$P(t)=\exp \left[\left(\ln \frac{G}{2-a} \right) \frac{\mu^+}{\ln 2} t \right] \left[\frac{a}{G+a-2} + P(0) \right] - \frac{a}{G+a-2}$$

(3.2.7)

Ha ugyanarra a tenyészetre alkalmazzuk a két modellt, azt kapjuk, hogy a következő relációknak kellene teljesülniük:

$$\frac{a}{2^{\alpha}+a-2} = \frac{\mu^+ \sigma}{\mu^- - (1-\sigma)\mu^+} \qquad (3.2.8)$$

$$\frac{\mu^+}{\ln 2} \ln \frac{2^{\alpha}}{2-a} = \mu^- - (1-\sigma)\mu^+ \qquad (3.2.9)$$

Behelyettesítve α definícióját:

$$\frac{a}{2^{\alpha}+a-2} = \frac{\sigma}{\alpha-1+\sigma} \qquad (3.2.10)$$

$$\frac{1}{\ln 2} \ln \frac{2^{\alpha}}{2-a} = (\alpha-1+\sigma) \qquad (3.2.11)$$

Egyszerűsítve:

$$(\alpha-1)a = (2^{\alpha}-2)\sigma \qquad (3.2.12)$$

$$2-a = 2^{1-\sigma} \qquad (3.2.13)$$

Értelmezzük a

$$\varphi(\alpha) = \frac{2^{\alpha}-2}{\alpha-1}$$

függvényt az $\alpha = 1$ helyen határértékével:

$$\varphi(1) = 2 \ln 2 \quad .$$

(3.2.12), (3.2.13)-ból kapjuk:

$$a = \varphi(\alpha)\ \sigma = [2 \ln 2 + O(\alpha-1)]\sigma \qquad (3.2.14)$$

$$a = 2 - 2^{1-\sigma} = [2 \ln 2 + O(\sigma)]\sigma \qquad (3.2.15)$$

Az utóbbi összefüggésekből kiolvasható, hogy bár nem található a és σ között tetszőleges α mellett érvényes összefüggés (hiszen (3.2.12) függ α-tól, (3.2.13) pedig nem), mindazonáltal mondhatjuk, hogy ha $\alpha \approx 1$ – ami a gyakorlatban nem ritka eset –, akkor

$$a \approx (2 \ln 2)\ \sigma \qquad (3.2.16)$$

A fenti tétel jelentősége abban áll, hogy ha a differenciaegyenlet-modell 'a' szegregációs együtthatóját meghatározzuk (pl. turbidosztát technológiával mért adatokból), akkor ugyanolyan tenyésztési körülmények mellett $\sigma = \frac{a}{2\ln 2} \approx 0.7a$ szegregációs rátával számolhatunk, ha a Monod-Aiba modellel dolgozunk.

3.3. A Monod-tipusú egyenletek és a stiff KDE-k

Térjünk vissza az eredeti (szegregáció nélküli) (1.1)-(1.2) Monod-egyenletekhez. Mint említettük, az ottani differenciálegyenletek numerikus megoldásában a nehézséget a $\mu=\mu(S)$ függés okozza. A μ-re adott Michaelis-Menten féle

$$\mu = \mu(S) = \mu_{max} \frac{S}{K_s+S}$$

összefüggés egy enzimkinetikai reakciókat leíró differenciálegyenlet rendszer pszeudo-állandósult állapotának közelítéséből származik (ld. Heineken et al. 1967). Ha ezeket a reakciókinetikai differenciál-egyenleteket az általunk vizsgált értéktartományból vett reakciósebességekkel elemezzük, kiderül, hogy bizonyos reakciók a rendszerben nagyságrendben gyorsabban zajlanak le mint mások. Ez az ún. stiff tulajdonság öröklődik a Michaelis-Menten összefüggés segítségével leírt Monod-egyenletekre is.

A stiff KDE-k vizsgálata az 50-60-as évekre vezethető vissza. "Stiff" angolul "merev"-et jelent; az elnevezés onnan származik, hogy a jelenséget először különböző merevségű rugók által vezérelt mozgást leíró differenciálegyenlet rendszereknél észlelték.

A stiff-ség problémája a következőképpen foglalható össze: Legyen adva az alábbi n egyenletből álló ($n\geq 1$) KDE rendszer és kezdeti feltétele:

$$y'(x) = f(x,y(x)) \qquad (f: S=[a,b]\times R^n \to R^n;\ a,b\in R) \qquad (3.3.1)$$

$$y(a) = y_0 \qquad (3.3.2)$$

ahol f kellően sima ahhoz, hogy (3.3.1) (3.3.2)-nek egyértelmű megoldása létezzen (ld. pl. Coddington-Levinson (1953)).

A fenti kezdeti érték probléma stiff az $[a,b]$ intervallumon, ha minden $x\in[a,b]$-re a $\frac{\partial f}{\partial y}$ Jacobi mátrix $\lambda_i(x)$ sajátértékeire:

i) $\quad \mathrm{Re}\ \lambda_i(x) < 0 \qquad (i=1,2,\ldots,n)$

ii) $\quad \max_i |\mathrm{Re}\ \lambda_i(x)| \gg \min_i |\mathrm{Re}\ \lambda_i(x)|$

Bár a fenti definició matematikailag nem egészen szabatos, jobbat nem sikerült még találni helyette. Mint kitűnik, a stiff-ség magától a keresett megoldástól is függ; tehát nem csak az f függvénytől, hanem az $[a,b]$ intervallumtól és a kezdeti értéktől is. Elképzelhető tehát olyan differenciálegyenlet rendszer, amely egyik intervallumon stiff, másikon nem; iletve amely egyik kezdeti értékre stiff, másikra nem.

A stiffség azt eredményezi, hogy a megoldások alaprendszerében vannak gyorsan és nagyon lassan változó tagok, és eredőjük numerikus megoldásánál a lépésköz meghatározásához a lassú tranzienst is figyelembe kell venni - noha az gyakorlatilag alig járul hozzá a megoldáshoz. Ez a fölöslegesen kicsi lépésköz a numerikus eljárás hibáját elviselhetetlenül megnövelheti.

Számítsuk ki ezek után az (1.1.1), (1.1.2) Monod-egyenletek Jacobi mátrixának sajátértékeit a $\mu(S)=\mu_{max}\frac{S}{K_s+S}$ Michaelis-Menten kinetika mellett:

$$\frac{d}{dt} x = (\mu(S) - D)\, x \tag{3.3.3}$$

$$\frac{d}{dt} S = D(S_{in} - S) - \frac{\mu(S)}{Y} x \tag{3.3.4}$$

$$J = \frac{\partial f}{\partial x} = \begin{bmatrix} \mu(S)-D & x \dfrac{\partial \mu(S)}{\partial S} \\ -\dfrac{\mu(S)}{Y} & -\dfrac{x}{Y} \dfrac{\partial \mu(S)}{\partial S} - D \end{bmatrix}$$

$$\lambda_1 = -D$$

$$\lambda_2 = (\mu(S) - D) - \frac{x}{Y} \frac{\partial \mu(S)}{\partial S} = -D - \frac{\mu_{max}}{K_s+S} \left[\frac{x}{Y} \frac{K_s}{K_s+S} - S \right]$$

A paramétereknek az 1.1. fejezetben említett értéktartományait figyelembe véve megállapítható, hogy a (3.3.3) (3.3.4) KDE stiffé válik ha $x(t)$ megnő. Érdemes tehát a szimulációs programokhoz rögtön stiff KDE-re készült programokat igénybe venni. A fenti KDE-t használja Valkó-Vajda (1987) is stiff KDE-re alkalmas - M72 - programjának bemutatására.

Mivel a (3.1.1)-(3.1.3) KDE-k a (3.3.3)-(3.3.4) egyenleteknek bizonyos értelemben általánosításai, így biztosak lehetünk abban, hogy (3.1.1)-(3.1.3)-at is érdemes stiff-ként kezelnünk.

4. A stiff KDE-k numerikus megoldása

4.1. KDE-k numerikus megoldásának alapjai

Célunk az, hogy a legmegfelelőbb módszert válasszuk ki a Monod-féle egyenletek numerikus megoldására. Ehhez először áttekintjük a KDE-k numerikus megoldásának elméleti alapjait (Stoer-Bulirsch (1980)).

Adott (3.3.1) (3.3.2) alakú kezdeti érték problémához egylépéses közelítő módszert határoz meg a $\phi(x,y;h)$ függvény, amennyiben az (x_i,η_i) pontokat tekintjük a keresett görbe közelítésének, ahol

$$\begin{aligned} x_0 &= a \quad , \quad \eta_0 = y_0 \\ x_{i+1} &= x_i + h \quad , \quad \eta_{i+1} = \eta_i + h\phi(x_i,\eta_i;h) \qquad (i=0,1,2,\dots) \end{aligned} \tag{4.1.1}$$

Legyen $(x,y)\in S$ tetszőleges, de rögzített. Legyen $z(t)$ a

$$\begin{aligned} \dot{z}(t) &= f(t,z(t)) \\ z(x) &= y \end{aligned} \tag{4.1.2}$$

kezdeti érték probléma pontos megoldása. Legyen

$$\Delta(x,y;h) = \begin{cases} \dfrac{z(x+h)-y}{h} & (h = 0) \\ f(x,y) & (h \neq 0) \end{cases} \tag{4.1.3}$$

Ekkor a

$$\tau(x,y;h) = \Delta(x,y;h) - \phi(x,y;h)$$

függvényt a módszer (x,y)-hoz tartozó lokális diszkretizációs (vagy képlet-) hibájának mondjuk. (Más szerzők lokális képlethibának inkább a $z(x+h)-(y+h\phi(x,y;h))$ függvényt nevezik.)

Elvárható, hogy egy közelítő módszerre teljesüljön:

$$\tau(x,y;h) \to 0 \qquad (h \to 0) \qquad ((x,y)\in S) \tag{4.1.4}$$

Ez ekvivalens azzal, hogy

$$\phi(x,y;h) \to f(x,y) \qquad (h \to 0) \qquad ((x,y)\in S) \tag{4.1.5}$$

A módszer konzisztens, ha (4.1.5) teljesül minden $f\in C_1[a,b]$ esetén, ahol

$$C_N[a,b] = \{f(x;y):S=[a,b]\times R^n\to R^n;\ f\ N\text{-szer folytonosan differenciálható } S\text{-en}\} \tag{4.1.6}$$

A módszer rendje p, ha

$$\tau(x,y;h) = O(h^p) \qquad (x,y\in S,\ f\in C_p[a,b]) \tag{4.1.7}$$

Legyen a (3.3.1), (3.3.2) kezdeti érték probléma pontos megoldása $y(x)$. Definiáljon $\phi(x,y;h)$ egy egylépéses módszert (4.1.1) szerint. Legyen $\eta(x)$ a módszer által szolgáltatott, a megoldásgörbét közelítő poligon:

$$\eta(x;h)=\begin{cases} \eta_i & (x=x_0+ih) \\ \eta_i+\dfrac{\eta_{i+1}-\eta_i}{h}(x-x_i) & (x_0+ih\le x<x_0+(i+1)h) \end{cases} \tag{4.1.8}$$

Ekkor rögzített $x\in S$-re az

$$e(x;h) = \eta(x;h) - y(x)$$

függvényt a módszer x-hez tartozó globális diszkretizációs (vagy képlet-) hibájának mondjuk. A módszer konvergens ha

$$e(x;h) \to 0 \qquad (h \to 0) \qquad (x\in S,\ f\in C_1[a,b]) \tag{4.1.9}$$

Bebizonyítható (pl. Stoer-Bulirsch (1980)), hogy $p>0$ rendü módszer konzisztens, sőt konvergens; éspedig $e(x;h)=O(h^p)$.

Legismertebb egylépéses módszerek az r-pontos Runge-Kutta tipusú közelítő eljárások, amelyek általános alakja:

$$\phi(x,y;h) = \sum_{i=1}^{r} w_i k_i \qquad (4.1.10)$$

ahol

$$k_i = f\left[x_n+\alpha_i h, y_n+h\sum_{j=1}^{r} \beta_{ij} k_j\right] \quad (i=1,2,\ldots r) \qquad (4.1.11)$$

(w_i, α_i, β_i, valós paraméterek, $w_r \neq 0$)

Ha $i \geq j$-re $\beta_{ij}=0$, akkor a módszer explicit; egyébként implicit.

Legyen $p_{max}(r)$ az a maximális rend, amelyet az r pontos módszer paramétereinek optimális választásakor kaphatunk. Bebizonyítható (pl. Móricz (1981)), hogy

a./ explicit módszerek esetében

$p_{max}(r) \leq r$

és egyenlőség csak $r \leq 4$ esetén teljesül

b./ implicit módszerek esetében $p_{max}(r) = 2r$ mindig elérhető.

Az implicit módszerek hátránya az, hogy egy újabb közelítő η_i érték kiszámításához egy (általában nemlineáris) egyenletrendszert is meg kell oldanunk. Ez a futási időt igen megnövelheti, ha f kiszámítása nagyon műveletigényes. A nemlineáris egyenletrendszerek iteratív megoldásánál az egyik nehézség a jó kezdőérték megadása. A fenti implicit módszerek esetében azonban az η_{i+1}-re kapott egyenletrendszer iteratív megoldásakor az előző η_i értéket célszerű kiindulópontként felhasználni. Ezzel általában jó kezdőértéket biztosítunk az iterációhoz.

Az ismertetett alapfogalmak többlépéses módszerekre természetes módon adaptálhatók. Itt csak a legismertebb, a lineáris k-lépéses módszer formáját mutatjuk be:

$$\eta_{n+1}=\sum_{j=1}^{k} a_j\eta_{n+1-j} - h\sum_{j=0}^{k}\beta_j f(x_{n+1-j},\eta_{n+1-j}) \qquad (4.1.12)$$

ahol α_j, β_j valós konstansok, $|\alpha_k|+|\beta_k|\neq 0$. A $\beta_0=0$ esetben explicit (prediktor), egyébként implicit (korrektor) a módszer. Jegyezzük meg, hogy a módszer nem "önindító", ha $k\geq 2$, hiszen több korábbi közelítő értéket is felhasznál. Tehát $x_0=a$-nál kiinduláshoz mindenképpen egylépéses módszerre van szükség.

4.2. Stiff KDE-k és A-stabilitás

Az A-stabilitás fogalmát G. Dahlquist vezette be (Dahlquist (1963)), majd Widlund fejlesztette tovább, megalkotva az $A(\alpha)$ stabilitás fogalmát (Widlund (1967)):

Egy módszer $A(\alpha)$ stabil ($\alpha\in(0,\pi/2)$), ha a módszer által szolgáltatott közelítő $\eta(x,h)$ megoldás tetszőleges rögzített $h>0$ mellett zérushoz konvergál ($x\to\infty$), midőn a módszert az

$$y'(x) = \lambda y(x) \qquad (y:R\to R) \qquad (4.2.1)$$

$$y(0) = y_0 \qquad (y_0\in R) \qquad (4.2.2)$$

tesztegyenletre alkalmazzuk, ahol

$$\lambda \in S(\alpha) = \{ z\in C: |\arg(-z)| < \alpha \} \qquad (4.2.3)$$

$S(\alpha)$ a módszer stabilitási tartománya. Ha $\alpha=\pi/2$, akkor a módszer stabilitási tartománya a baloldali komplex félsík. Ekkor a módszert csak A-stabilnak, vagy abszolút stabilnak mondjuk.

Az A-stabilitásról az alaptételek Dahlquist-tól származnak:

a./ Nem létezik explicit lineáris többlépéses módszer, amely A-stabilis lenne.

b./ Egy implicit, A-stabilis, lineáris többlépéses módszer maximális rendje 2 .

A stiff KDE-k megoldására az A-stabilis módszerek a megfelelő numerikus eljárások.

Az $y'=f(y)$ alakú, ún. autonóm rendszerekre igen hatékony eljárások stiff esetben a Rosenbrock (1963) gondolatán alapuló módszerek. Stoer-Bulirsch (1980) is kiemeli ezek közül Kaps-Rentrop (1979) módszerét (ROW4), amely viszonylag egyszerűen programozható és hatékony hibabecsléssel rendelkezik. Ezen a módszeren javasolt Gottwald-Wanner (1981) egy "visszalépéses" lépésvezérlő technikát, s így született a ROW4A program. Mivel a Monod-, illetve Monod-Aiba egyenletek autonóm KDE-k így mi is egy negyedrendű Rosenbrock módszert választottunk. Az alkalmazott lépésvezérlés a "visszalépéses" technika egy változata lesz. Megjegyezzük, hogy Valkó-Vajda (1987) is a ROW4A módszert programozta le BASIC nyelven az M72-es modulban, így az ő progamjuk alkalmas lesz az általunk készített programmal való összehasonlításra. Erről az 5.3. fejezetben lesz szó.

4.3. A Rosenbrock tipusú módszerek

Legyen adva az

$$\left.\begin{aligned} y' &= f(y) \qquad (f: R^n \to R^n) \\ y(x_0) &= y_0 \end{aligned}\right\} \qquad (4.3.1)$$

autonóm kezdeti érték probléma, s vizsgáljuk az r pontos

$$\eta_{i+1} = \eta_i + \sum_{i=1}^{r} w_i k_i \qquad (w_i \in R,\ w_r \neq 0) \qquad (4.3.2)$$

alakú módszert, ahol i=1,2,...,r-re:

$$k_i = hf\left[\eta_i + \sum_{j=1}^{i-1} \beta_{ij} k_j\right] + hJ(\eta_i) \sum_{j=1}^{i} \gamma_{ij} k_j \qquad (4.3.3)$$

$(\beta_{ij},\ \gamma_{ij} \in R;\ J(y) = \frac{\partial f}{\partial y}$ az f Jacobi mátrixa)

A módszer Rosenbrock (1963)-ra vezethető vissza, később többen vizsgálták (Wolfbrandt (1977), Kaps-Rentrop (1979), Norsett-Wolfbrandt (1979), Kaps-Wanner (1981), Gottwald-Wanner (1981), Kaps-Rentrop (1984)). A módszer alapötlete a szemi-implicit Runge-Kutta tipusú módszerekből származik, ahol k_i nem függ $k_{i+1}, \ldots, k_r$-től (i=1,2,...,r). (Innen a szemi-implicit elnevezés.) Az ötlet az, hogy f-et elsőrendű közelítésével helyettesítjük (innen a Jacobi mátrix megjelenése), és így végül mégis explicit formulát kapunk:

$$(I - h\gamma_{ii} J_i) k_i = hf(\eta_i + \sum_{j=1}^{i-1} \beta_{ij} k_j) + hJ_i \sum_{j=1}^{i-1} \gamma_{ij} k_j \qquad (4.3.4)$$

ahol I az n*n-es egységmátrix, $J_i = J(\eta_i)$ (i=1,2,3,...r).

Vegyük észre, hogy az i-dik lépésben az $A_i(h)=I-h\gamma_{ii}J(\eta_i)$ mátrixegyütthatónak elég egyszer megállapítani az LU felbontását, hiszen a különböző k_i értékek kiszámításánál csak a lineáris egyenletrendszer jobboldala változik. Így a műveletigény jelentősen csökkenthető.

Ugyancsak érdemes megjegyezni, hogy elég kicsi h-ra $A_i(h)$ biztosan nem szinguláris, hiszen $A_i(h)\to I$ $(h\to 0)$.

A (4.3.2)-(4.3.4) típusú módszert szokás Rosenbrock-Wanner, röviden ROW-módszernek is nevezni.

Ha (4.3.4)-et az ismert Taylor-sor módszerrel kiértékeljük, a módszer paramétereire különböző összefüggéseket állapíthatunk meg attól függően, hányadrendű módszert szeretnénk kapni. Hogy A-stabil módszerünk legyen minél kisebb műveletigénnyel és minél hatékonyabb hibabecsléssel, Kaps és Rentrop (1979) egy negyedrendű módszert javasolt:

$$\eta_{i+1} = \eta_i + b_1k_1 + b_2k_2 + b_3k_3 + b_4k_4$$

ahol

$$A_i(h)k_1 = hf(\eta_i)$$

$$A_i(h)k_2 = hf(\eta_i+a_1k_1)+c_1k_1$$

$$A_i(h)k_3 = hf(\eta_i+a_2k_1+a_3k_2)+c_2k_1+c_3k_2$$

$$A_i(h)k_4 = hf(\eta_i+a_2k_1+a_3k_2)+c_4k_1+c_5k_2+c_6k_3)$$

$A_i(h)=I-h\gamma J(\eta_i)$;
(az $a_1,a_2,a_3,c_1,c_2,c_3,c_4,c_5,c_6,b_1,b_2,b_3,b_4,\gamma$ konstansok értékei az 5.2. fejezetben közölt programlistából kiolvashatók.)

A továbbiakban a fenti módszerre ROW4 néven hivatkozunk.

5. A ROW4 eljárás alkalmazása

5.1. A ROW4B algoritmus

A 3.3. fejezetben elmondottakból következik, hogy a lépésvezérlés a stiff KDE numerikus megoldásának pontosságában döntő szerepet játszik. Az i-dik lépésben a h aktuális lépésköz, az err aktuális ("ha h lépést tennénk") becsült hiba, és az ep előírt pontosság (relatív tolerancia) függvényében állapítják meg általában a lépésvezérlő eljárások a h_{new} új lépésközt.

Valkó-Vajda (1987) a ROW4 módszer i-dik lépésében a következő hibabecslést alkalmazta (Kaps-Rentrop (1979) nyomán):

Legyen η_i az x_i helyen kapott közelítés, és tegyük fel, hogy kiszámoltuk az

$$\eta_{i+1}^{(h)} = \eta_i + h\phi(x_i, y_i; h)$$

értéket az aktuális lépésközzel. Ekkor $\eta_{i+1}^{(h)}$ relatív hibájának becslése:

$$err = \max_{1 \le j \le n} \frac{2|e_1 k_1(j) + e_2 k_2(j) + e_3 k_3(j) + e_4 k_4(j)|}{\frac{ep}{1000} + |\eta_i(j)| + |\eta_{i+1}(j)|} \qquad (5.1.1)$$

ahol az e_1, e_2, e_3, e_4 konstansok értékei az 5.2. fejezetben közölt programlistából kiolvashatók.

Legyen

$$er = \max\left(\frac{ep}{16}, err\right) \qquad (5.1.2)$$

(melynek szerepéről később lesz szó), és legyen

$$h_{new} = 0.9h \sqrt[4]{\frac{ep}{er}} \qquad (5.1.3)$$

Ekkor azt várjuk, hogy az $x_{i+1}=x_i+h_{new}$ helyen

$$\eta_{i+1} = \eta_i+h_{new}\phi(x_i,\eta_i;h_{new}) \qquad (5.1.4)$$

értékkel számolva - itt nem részletezett simasági feltételek mellett - annak relatív hibája az előírt hibakorláton belül lesz. (A 0.9-es faktornak biztonsági szerepe van.)

Gottwald és Wanner (1981) nyomán a következő finomítást végezzük el a ROW4 eljárás lépésszabályozásán:

$i\geq 1$ esetén legyen $h_{old}=x_i-x_{i-1}$ $(i\geq 1)$.

1./ Ha $er\leq ep$, akkor elfogadjuk az új közelítést (i-ről i+1-re lépünk). Ekkor könnyen megeshet (éppen stiff KDE-k esetében), hogy az (5.1.1)-ben definiált err sokkal kisebb mint ep, így (5.1.3) miatt h_{new} jóval nagyobb is lehetne mint h. Ezt ellensúlyozandó szerepel $\frac{ep}{16}$ (5.1.2) jobboldalán, ami biztosítja, hogy

$$h_{new} \leq 0.9 \sqrt[4]{16}\ h = 1.8h \qquad (5.1.5)$$

2.a./ Ha $er>ep$ és $i\geq 1$, akkor visszalépünk egyet i-ről (i-1)-re. (Ekkor az (5.1.3)-ban szereplő h_{new} kisebb mint h .) Újraszámoljuk x_{i-1}-nél az értékeket

$$\tilde{h} = \min(h_{old},h_{new}) \qquad (5.1.6)$$

lépésközzel, amely tehát kisebb vagy egyenlő, mind

az (i-1)-nél korábban alkalmazott lépésköznél (h_{old}), mind az új, javasolt lépésköznél (h_{new}), mindpedig az i-nél kipróbált h lépésköznél. Ezzel biztosítva vagyunk affelől, hogy elég kis lépésközt választottunk, hiszen egyszer már ennél nagyobb lépést (h_{old}) is elfogadott a hibateszt. Ezt a $\tilde{h}$ választott lépést nem engedjük megnövelni a következő (i-dik) lépésben.

b./ Ha i=0 és er>ep vagy éppen egy visszalépést követő első lépésnél vagyunk és er>ep, akkor egyszerűen $\tilde{h}=h_{new}$ lépésközzel újraszámoljuk az értékeket; (5.1.3) biztosítja, hogy ekkor $\tilde{h}<h$.

Gottwald és Wanner (1981) javasolt hasonló "visszalépéses" lépésszabályozást, hogy az eljárás időben reagáljon arra, ha az f függvény (a kezdeti érték probléma jobb oldalán) hirtelen változik. A mi lépés-szabályozásunknak is természetesen ez a célja. A különbség Gottwald-Wanner ROW4A programjától az, hogy a mi algoritmusunkban a visszalépés utáni lépésválasztás (5.1.6) szerint történik; valamint, hogy a 2.b pont miatt visszalépés után mindenképpen legalább kettő előrelépés következik.

Ezzel a lépésvezérléssel készült, ROW4 módszert használó programunkat ROW4B-nek nevezzük.

Az eljárást FORTRAN '77 nyelven programoztuk le, listáját a következő fejezetben közöljük.

5.2. A ROW4B FORTRAN szubrutin

Az alábbiakban megadjuk a FORTRAN '77 nyelven megírt ROW4B szubrutin listáját. A program algoritmusa kiolvasható a 4.3. és 5.1. fejezetekből, így azt külön nem részletezzük.

```
      SUBROUTINE ROW4B(N,T,Y,TE,EP,H,IM,M,P,IER)
C
C*** solving stiff system of ordinary differential equations
C*** ROW4 algorithm with Backstep
C
C  input:
C         n      number of equations
C         t      initial time
C       y(i)     initial values at t
C        te      upper bound of the integration
C        ep      relative tolerance for y(i)
C         h      initial steplength
C        im      maximum number of steps + back steps
C         m      number of parameters
C       p(j)     parameters
C  output:
C        ier     error code
C                   0  successful return
C                   1  max. number of steps is not enough
C                   2  initial time is not less then the upper bound
C         t      time at return
C       y(i)     solution at t
C         h      next steplength proposed
C  user supplied routines:
C         F  :   t,y(i) --> d(i)     (right side of the ODE)
C
C  The next arrays must be dimensioned by means of the value of N;
      DIMENSION E(2,2),A(2,2),B(2),IAO(2)
C
      DIMENSION Y(2),YR(2),YB(2),D(2),P(M),
     +          RK1(2),RK2(2),RK3(2),RK4(2),DD(2),YD(2)
      IF (T.LT.TE.AND.EP.GT.0.0.AND.N.LE.4) GOTO 10
      IER=2
      RETURN
```

```
C
C  initial values
C
   10  GAMMA=.39500000E+00
       A1=.43800000E+00
       A2=.93894868E+00
       A3=.73079542D-01
       C1=-.19434744E+01
       C2=.41695753E+00
       C3=.13239678E+01
       C4=.15195133E+01
       C5=.13537082E+01
       C6=-.85415150E+00
       B1=.72904488E+00
       B2=.54106977E-01
       B3=.28159936E+00
       B4=.25000000E+00
       E1=-.19085887E-01
       E2=.25560879E+00
       E3=-.86381628E-01
       E4=.25000000E+00
       IP=0
       IR=0
       ILB=0
       LB=0
       LE=0
   11  DO 12 I=1,N
   12  YR(I)=Y(I)
       HR=H
       TR=T
C
   14  CONTINUE
       WRITE (2,3) T,IP,IR,ILB,ES
       WRITE (*,1) T,(Y(I),I=1,N)
       IF (IP+ILB+IR.LT.IM) GOTO 16
       IER=1
       RETURN
   16  IF (T+H.LT.TE) GOTO 18
       LE=1
       H=TE-T
C
   18  CALL F(N,Y,D,M,P)
```

```
C
C   This loop to the label 182 computes the Jacobian.
C   It can be exchanged for its analytical computation:
C         CALL JFY(N,Y,E,M,P)
C
      DO 182 J=1,N
      DO 181 I=1,N
 181  YD(I)=Y(I)
      YDJ=ABS(Y(J))*1.E-3+1.E-7
      YD(J)=Y(J)+YDJ
      CALL F(N,YD,DD,M,P)
      DO 183 I=1,N
 183  E(I,J)=(DD(I)-D(I))/YDJ
 182  CONTINUE
C
C  computing the coefficient matrix
C
  22  DO 24 I=1,N
      DO 24 J=1,N
      IJ=0
      IF (I.EQ.J) IJ=1
      A(I,J)=-GAMMA*H*E(I,J)+IJ
  24  CONTINUE
      CALL MXLU(N,A,IAO,IA,IER,1.E-8)
      IF (IER.EQ.0) GOTO 26
      H=H/2
      GOTO 22
C
C   solving the linear systems
C
  26  CONTINUE
      DO 28 I=1,N
  28  B(I)=H*D(I)
      CALL SUBST(N,A,IAO,IA,B)
      DO 30 I=1,N
      RK1(I)=B(I)
  30  Y(I)=YR(I)+A1*RK1(I)
C
      CALL F(N,Y,D,M,P)
      DO 32 I=1,N
  32  B(I)=H*D(I)+C1*RK1(I)
      CALL SUBST(N,A,IAO,IA,B)
      DO 34 I=1,N
      RK2(I)=B(I)
  34  Y(I)=YR(I)+A2*RK1(I)+A3*RK2(I)
C
      CALL F(N,Y,D,M,P)
      DO 38 I=1,N
  38  B(I)=H*D(I)+C2*RK1(I)+C3*RK2(I)
```

```
      CALL SUBST(N,A,IAO,IA,B)
      DO 40 I=1,N
      RK3(I)=B(I)
   40 B(I)=H*D(I)+C4*RK1(I)+C5*RK2(I)+C6*RK3(I)
      CALL SUBST(N,A,IAO,IA,B)
      DO 42 I=1,N
      RK4(I)=B(I)
   42 Y(I)=YR(I)+B1*RK1(I)+B2*RK2(I)+B3*RK3(I)+B4*RK4(I)
      T=T+H
C
C        error estimation
C
      ES=EP/16.0
      DO 44 I=1,N
      S1=ABS(E1*RK1(I)+E2*RK2(I)+E3*RK3(I)+E4*RK4(I))
      S2=ABS(Y(I))+ABS(YR(I))+EP/1.E3
      S=2*S1/S2
      IF (S.GT.ES) ES=S
   44 CONTINUE
C
C       step control
C
      S=.9*(EP/ES)**.25
      IF (LB.EQ.-1.AND.ES.LT.EP) S=.9
      HN=S*H
      IF (ES.GT.EP) GOTO 54
C
C       accepted step
C
      IP=IP+1
      IF (LE.NE.1) GOTO 46
      H=HR
      IER=0
      WRITE (2,3) T,IP,IR,ILB,ES
      WRITE (*,1) T,(Y(I),I=1,N)
   3  FORMAT (E15.6,3I6,E10.3)
   1  FORMAT(5E15.6)
      RETURN
   46 DO 48 I=1,N
      YB(I)=YR(I)
   48 YR(I)=Y(I)
      TB=TR
      HB=HR
      HR=H
      TR=T
      H=HN
      IF (LB.LE.0) LB=LB+1
      GOTO 14
   54 LE=0
      IF (LB.EQ.1) GOTO 50
```

```
C
C   recomputation with smaller h
C
      IR=IR+1
      DO 56 I=1,N
   56 Y(I)=YR(I)
      T=TR
      H=HN
      GOTO 22
C
C   one step back
C
   50 CONTINUE
      DO 60 I=1,N
   60 Y(I)=YB(I)
      T=TB
      H=MIN(HN,HB)
      ILB=ILB+1
      IP=IP-1
      LB=-1
      GOTO 11
      END


      SUBROUTINE MXLU(N,A,IAO,IA,IER,EP)
C
C *** construction of the LU form of a square matrix A
C *** Gauss elimination with partial pivoting
C
C  input:
C    n          number of rows/columns
C    a(i,j)     matrix
C  output:
C    ier        error code
C                 0  successful return
C                 1  singular matrix  ( pivot element < ep )
C    a(i,j)       LU in concise form
C    iaO(i)     indices referring to the largest elements in the
C               columns ( --> permutation matrix)
C    ia         sign referring to the permutation matrix
C
C     IMPLICIT REAL *8 (A-H,O-Z)
C   The next arrays must be dimensioned explicitly
      DIMENSION A(2,2),IAO(2)
      IA=1
      DO 10 K=1,N-1
      M=K
      DO 12 I=K+1,N
      IF (ABS(A(I,K)).GT.ABS(A(M,K))) M=I
 12   CONTINUE
```

```
      IAO(K)=M
      AS=A(M,K)
      IF (M.LE.K) GOTO 14
      IA=-IA
      A(M,K)=A(K,K)
      A(K,K)=AS
14    IF (ABS(AS).GT.EP) THEN
                    AS=1/AS
      ELSE
                    IER=1
                    RETURN
      ENDIF
      DO 15 I=K+1,N
15    A(I,K)=-A(I,K)*AS
      DO 18 J=K+1,N
      AS=A(M,J)
      A(M,J)=A(K,J)
      A(K,J)=AS
      IF (AS.EQ.0.0) GOTO 18
      DO 16 I=K+1,N
16    A(I,J)=A(I,J)+A(I,K)*AS
18    CONTINUE
10    CONTINUE
      IER=0
      IF (ABS(A(N,N)).LE.EP) IER=1
      RETURN
      END


C
      SUBROUTINE SUBST(N,A,IAO,IA,B)
C
C *** substituting backward to get the solution of the system
C *** of linear equations Ax=b where the square matrix A is
C *** given in concise LU form
C
C input:
C       n                    size of A
C   a(i,j), iaO(i), ia       see output of MXLU
C     b(i)                   vector of the right side
C
C     IMPLICIT REAL *8 (A-H,O-Z)
C These arrays must be dimensioned explicitly
      DIMENSION A(2,2),B(2),IAO(2)
      DO 10 K=1,N-1
        I=IAO(K)
        AS=B(I)
        B(I)=B(K)
        B(K)=AS
        DO 12 I=K+1,N
12    B(I)=B(I)+A(I,K)*AS
10    CONTINUE
```

```
      DO 14 K=N,1,-1
      B(K)=B(K)/A(K,K)
      DO 15 I=1,K-1
15    B(I)=B(I)-A(I,K)*B(K)
14    CONTINUE
      RETURN
      END
```

A ROW4B által hívott MXLU szubrutin egy négyzetes mátrix LU felbontását végzi főelemkiválasztásos Gauss-eliminációval, a SUBST szubrutin pedig az LU alakban megadott mátrixú lineáris egyenletrendszert oldja meg. A rutinok Valkó-Vajda (1987) könyvében szereplő M14 illetve M15 BASIC nyelvű modulok FORTRAN változatai.

Ezen kívül ROW4B hívja még a felhasználó által megadott F szubrutint, amely a differenciálegyenlet jobboldalát számítja ki. A Jacobi mátrixot numerikus differenciálással állítja elő a program, de ez nyilvánvalóan helyettesíthető az analitikus deriválttal; ebben az esetben még egy felhaszálói szubrutin megadása szükséges.

5.3. ROW4B tesztelése

A ROW4B szubrutin tesztelésére tekintsük a következő kezdeti érték feladatot (Stoer-Bulirsch (1980) nyomán):

$$y_1' = -\frac{101}{2} y_1 + \frac{99}{2} y_2 \qquad (5.3.1)$$

$$y_2' = \frac{99}{2} y_1 - \frac{101}{2} y_2 \qquad (5.3.2)$$

$$y_1(a) = \alpha_1 \qquad (5.3.3)$$

$$y_2(a) = \alpha_2 \qquad (5.3.4)$$

Az (5.3.1), (5.3.2) KDE $y'=f(y)$ alakú. Jacobi mátrixának sajátértékei: $\lambda_1=-1$, $\lambda_2=-100$ (y_1, y_2-től függetlenül). A kezdeti érték probléma tehát stiff, függetlenül az [a,b] integrálási intervallumtól és az α_1, α_2 kezdeti értékektől. A KDE általános megoldása:

$$y_1(x) = C_1e^{-x} + C_2e^{-100x} \tag{5.3.5}$$

$$y_2(x) = C_1e^{-x} - C_2e^{-100x} \tag{5.3.6}$$

amelyből a C_1, C_2 együtthatók behelyettesítéssel adódnak, ha a kezdeti értékek által meghatározott partikuláris megoldást keressük.

Mi a tesztelés során különböző [a,b] intervallumokon az

$$\alpha_1 = e^{-a} + e^{-100a} \tag{5.3.7}$$

$$\alpha_2 = e^{-a} - e^{-100a} \tag{5.3.8}$$

kezdeti feltételekkel dolgozunk, amelyek mellett a megoldás nyilván ($C_1=C_2=1$):

$$y_1(x) = e^{-x} + e^{-100x} \tag{5.3.9}$$

$$y_2(x) = e^{-x} - e^{-100x} \tag{5.3.10}$$

Látható, hogy $x<-1$-re $y_i(x)\approx e^{-100x}$ $(i=1,2)$ $x>1$-re pedig $y_i(x)\approx e^{-x}$ $(i=1,2)$.

Tehát, ha az $\{e^{-x},e^{-100x}\}$ függvényrendszert tekintjük bázisnak, akkor a bal féltengelyen az egyik bázisfüggvény a domináns, a jobb féltengelyen a másik.

Hasonló dominancia igaz a deriváltakra is:

$$y_1'(x) \approx -100e^{-100x} \qquad (x < -1)$$

$$y_2'(x) \approx -e^{-x} \qquad (x > 1)$$

A megoldásgörbe deriváltja tehát rendkívül gyorsan változik x=0 körül. Mi éppen az ilyen tulajdonságú megoldások okozta buktatókat kívánjuk elkerülni a ROW4B program lépésvezérlési technikájával. Hogy ez sikerült vagy sem, s ha igen, mennyire, arról képet kaphatunk a következő táblázatból:

$a=-0.1 \qquad \alpha_1=y_1(a)=22027.57 \qquad \alpha_2=y_2(a)=-22025.36$

x	ROW4B	M72	Pontos érték	Rel.hiba (ROW4B)	Rel.hiba (M72)
-0.1	.2203E+5	.2203E+5	.2203E+5	0.0E-2	0.0E-2
	-.2203E+5	-.2203E+5	-.2203E+5	0.0E-2	0.0E-2
0.0	.2005E+1	.1990E+1	.2000E+1	0.1E-2	0.1E-2
	-.5180E-2	-.2450E-1	.0000E+0	0.5E-2	2.4E-2
0.1	.9049E+0	.8893E+0	.9049E+0	0.0E-2	1.6E-2
	.9047E+0	.8891E+0	.9048E+0	0.0E-2	1.6E-2
1.0	.3679E+0	.3615E+0	.3679E+0	0.0E-2	0.6E-2
	.3679E+0	.3615E+0	.3679E+0	0.0E-2	0.6E-2
10.0	.4575E-4	.4495E-4	.4540E-4	0.4E-2	0.5E-2
	.4541E-4	.4472E-4	.4540E-4	0.0E-2	0.7E-2

A táblázat a tesztfeladatnak két különböző eljárással (ROW4B és Valkó-Vajda (1987) M72 modulja) kapott numerikus megoldásának eredményét mutatja be. A helyes megoldást tudjuk:

$$y_1(x) = e^{-x}+e^{-100x}$$
$$y_2(x) = e^{-x}-e^{-100x}$$

Ezen függvények $x=-0.1$ helyen vett helyettesítési értékeit írtuk be mindkét programban 7 helyes jeggyel az $\alpha_1=y_1(-0.1)$ illetve $\alpha_2=y_2(-0.1)$ kezdőértékek helyére. A második illetve harmadik oszlopban a jelölt x helynek megfelelő, ROW4B illetve M72 programok által szolgáltatott $y_1(x)$, $y_2(x)$ közelítések találhatók egymás alatt négy jegyre kerekítve. A harmadik oszlopban a helyes értékek vannak szintén négy jeggyel. A negyedik, ötödik oszlop a programok által szolgáltatott eredmények hibáit mutatja.

Az előírt relatív pontosság ep=1.E-2 , a lépésköz kezdeti értéke h=0.1 volt mindkét programnál.

A ROW4B FORTRAN '77 program IBM-kompatibilis számítógépen szimpla pontosságú ($\approx$ 7 tizedes jegyű) aritmetikával futott, míg az M72-es program ugyanezen a gépen BASIC interpreterrel, ugyancsak szimpla pontosságú aritmetikával dolgozott.

A táblázatokból a következők olvashatók ki:

Az előírt pontosságot M72 nem tudta elérni az $x=0.0$, $x=0.1$ pontokban, míg ROW4B minden kiírt pontban teljesítette azt. ROW4B feltűnően pontos eredményt adott a [0,1] intervallumban.

A fentiek magyarázata a következő:

ROW4B és M72 csak a visszalépés technikájában különböznek. Ennek a technikának különösen ott van jelentősége, ahol a megoldásgörbe mentén f(y) gyorsan változik; vagyis - mint korábban elemeztük - esetünkben az x=0 pont környezetében. Az általunk módosított visszalépésnek tudható be, hogy ROW4B a hibakorláton belüli eredményt szolgáltatott a "veszélyes" [-0.1,0.1] intervallumon; ellentétben M72-vel, amely pedig ugyanolyan pontos aritmetikával dolgozott. Ezt a részletes kiíratások is mutatták, amelyekből kiderült, hogy ROW4B-nek ebben az intervallumban valóban többször is vissza kellett lépnie.

Hangsúlyozandó, hogy M72-nek nem az a hátránya ROW4B-vel szemben, hogy pontatlanabb eredményt szolgáltatott. A próbafuttatások mutatják, hogy pl. a=0.1-ből indulva (ettől jobbra már nincs az előzőhöz hasonló "veszélyes" rész) már nincs lényeges különbség a ROW4B illetve M72 által szolgáltatott output között.

M72-nek az a hátránya, hogy a [-0.1,0.1] intervallumban is 1.E-2 alatti relatív hibát becsül, s ezzel mintegy "félrevezeti" a felhasználót. Ezt küszöböli ki ROW4B-ben alkalmazott visszalépéses technika.

Megjegyezzük, hogy ROW4B duplapontos változata - mint az várható volt a fentiek után - valóban pontosabb eredményt szolgáltatott bármilyen kezdőérték és előírt pontosság esetén, mint M72.

5.4. A differenciálegyenlet-modell szimulációja a ROW4B programmal

Térjünk vissza a (3.1.1)-(3.1.3) Monod típusú differenciálegyenletekhez, és oldjuk meg azt a ROW4B eljárás duplapontos változatáva.

A megoldandó KDE jobboldalát a ROW4B által hívott F felhasználói szubrutinban, programoztuk le. A szubrutin, illetve a főprogram listáját az alábbiakban közöljük:

```
C
C*** MONROW
C     Main program to solve the Monod-equations
C
      IMPLICIT REAL *8 (A-H,O-Z)
      DIMENSION Y(3),P(12)
      OPEN (1,FILE='MON.PAR')
      OPEN (2,FILE='MON.OUT')
      M=9
      N=3
      READ (1,101) (P(I),I=1,9)
  101 FORMAT(/6F10.3)
      READ (1,101) Y(1),Y(2),Y(3),TE
      READ (1,101) T,H,EP
      IM=5000
      CALL ROW4B(N,T,Y,TE,EP,H,IM,M,P,IER)
      WRITE (2,105) IER
  105 FORMAT(/' IER=',I4)
      CLOSE (1)
      CLOSE (2)
      STOP
      END
C
      SUBROUTINE F(N,Y,D,M,P)
C
C*** right side of the O.D.E. y'=f(y)
C
C  input:
C          n     number of equations
C        y(i)    values of the dependent (vector-) variable at t
C          m     number of parameters
C        p(i)    vector of parameters
C  output:
C        d(i)    values of y'(i) at y
C
      IMPLICIT REAL *8 (A-H,O-Z)
      DIMENSION P(M),Y(N),D(N)
```

```
      SMP=P(5)*Y(3)/(P(3)+Y(3))
      SMN=P(6)*Y(3)/(P(4)+Y(3))
      D(1)=((1.D0-P(1))*SMP-P(2))*Y(1)
      D(2)=P(1)*SMP*Y(1)+(SMN-P(2))*Y(2)
      D(3)=P(2)*(P(7)-Y(3))-SMP*Y(1)/P(8)-SMN*Y(2)/P(9)
      RETURN
      END
C
```

A futtatáshoz az input file képe a következő (a paramétereket és a kezdőértékeket az irodalomnak illetve a gyakorlatnak megfelelően adtuk meg):

szigma	D	Ks+	Ks-	mu+	mu-
0.0002	0.6	0.008	0.007	1.0	1.2

Sin	Y+	Y-
20.0	0.5	0.6

X+(0)	X-(0)	S(0)	TE
0.01	0.0	2.0	100.0

TST	H	EP
0.0	0.1	0.001

A fenti input file első két sorából kapja a P paramétervektor az értékeit; a harmadik sorban a függő változók kezdőértékei valamint az integrációs intervallum jobb végpontja, a negyedik sorban a kezdeti időpont, a kezdeti lépésköz és az előírt pontosság kaptak helyet.

A futás során elfogadott lépésköz esetén a ROW4B program az output file-ba írta az aktuális időt, a függő változók értékeit, azok pontosságát ($\| \cdot \|_\infty$ értelemben), a pozitív irányban megtett lépések, az újraszámolt lépések és a visszalépések számát. A program működését jól illusztrálja a futtatás során kapott output file néhány sora (amelyből most a függő változók értékeit elhagyjuk):

T	IP	IR	ILB	ERR
.163439E+02	29	0	0	.66E-03
.169998E+02	30	0	0	.72E-03
				.19E-01
.163439E+02	29	0	1	.66E-03
.166186E+02	30	0	1	.63E-04
.168658E+02	31	0	1	.63E-04
				.84E-01
.166186E+02	30	0	2	.63E-04

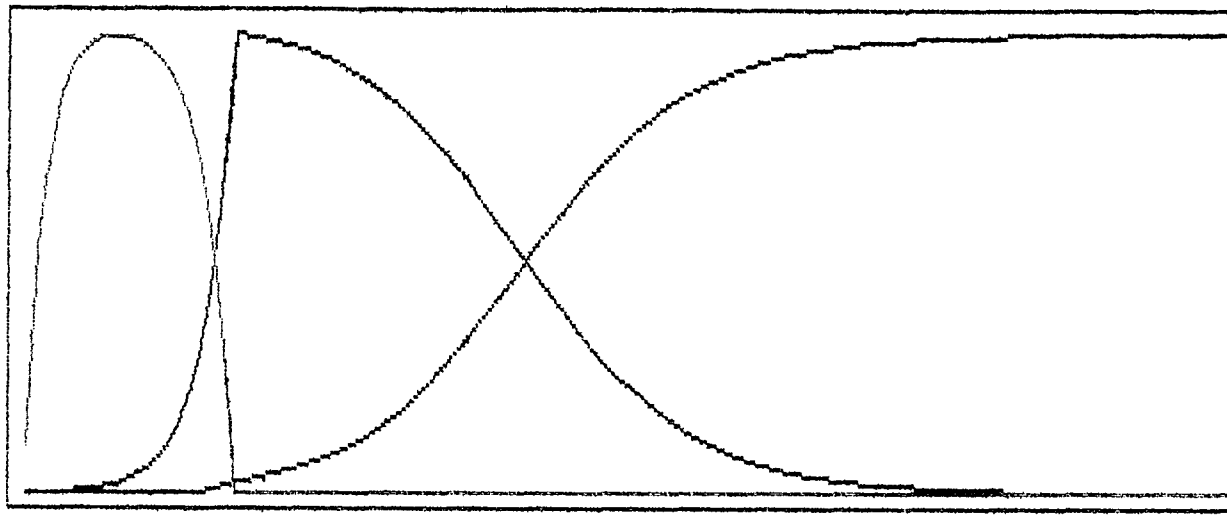
Az eljárás IP=30 lépésben jutott el a T=0.169998E+2 pontig, ahol a következő lépés becsült hibája nagyobb mint a megengedett 0.1E-1 (így a lépés eredményét ki sem írtuk). Ezért a program visszalépett az IP=29. lépésre (T=0.163439E+2-re), és kisebb lépéssel, mindössze T=0.166186E+2-re került. Így a visszalépést számláló ILB változó tartalma eggyel növekedett. A következő lépésnél, T=.168658E+2-nél ez megismétlődik, amiből kiderül, hogy a "veszélyes" szakasz valahol 16.8-17.0 körül kezdődik.

Az output file-ból kitűnik, hogy ilyen "veszélyes" szakasz csak az integrálási intervallum elején van; ennek illusztrálására lássuk az output file néhány sorát az integrálási intervallum közepéről:

.464647E+02	76	0	15	.68E-03
.484544E+02	77	0	15	.65E-03
.504478E+02	78	0	15	.63E-03
.524608E+02	79	0	15	.62E-03
.545046E+02	80	0	15	.61E-03
.565859E+02	81	0	15	.61E-03
.587084E+02	82	0	15	.61E-03
.608735E+02	83	0	15	.61E-03
.630813E+02	84	0	15	.61E-03

Később ugyanilyen nagy lépésközökkel számolhatott a program; hibakorláton belül maradt és nem volt szükség visszalépésre.

A futtatás outputjából szimulációs görbéket rajzoltunk ki, amelyek a következő oldalon láthatók.



	min	max	dimens.	m e a n i n g
T	0.00E+00	1.00E+02	(h)	time
X+	3.42E-03	9.75E+00	(g/l)	conc. of pos. cells
X-	0.00E+00	1.20E+01	(g/l)	conc. of neg. cells
S	7.00E-03	1.95E+01	(g/l)	substr. conc.

A (3.1.1)-(3.1.3) KDE-k szimulációs görbéi az mertetett input adatok mellett.

Jól látható, hogy T=17 h körül (ahol ROW4B többszöri visszalépéssel számolt) a megoldások deriváltja valóban rohamosan változik.

6. Továbbfejlesztési lehetőségek

A (3.1.1)-(3.1.3) Monod-típusú egyenletek szerint a folyamatot a pozitív és a negatív sejtek száma (koncentrációja) írja le. Számunkra azonban még érdekesebb a sejtekben lévő plazmidok számának (koncentrációjának) alakulása. Ennek modellezésére éppen az utóbbi időkben történtek kíséreletek (Satayagal-Agrawal (1988), Greenhalf et al. 1989). A kritikus kérdés, hogy ha $\nu(t)$ jelöli a plazmidok specifikus növekedési sebességét:

$$\dot{p}(t) = \nu(t)\, p(t)$$

ahol $p(t)$ a plazmidok koncentrációja (sejtektől függetlenül), akkor ν hogyan függ a modell többi paramétereitől (elsősorban μ-től).

A mellékletben található egy - e dolgozat szerzőjének részvételével készült még megjelenésre el nem fogadott - cikk, amelynek egyik eredménye a következő:

Jó okunk van feltételezni, hogy a szegregáció valószínűsége a következőképpen függ az egy pozitív sejtben lévő N átlagos plazmidszámtól:

$$\mathrm{Prob(szegregáció)} = e^{-\alpha N}$$

ahol α valamilyen (kb. $\frac{1}{2}$ és 1 közötti) konstans. Az átlagos plazmidszám:

$$N(t) = k\,\frac{p(t)}{x^{+}(t)}$$

ahol k egy sejt és egy plazmid tömegének aránya. Az pedig a Poisson folyamatot (ld. pl. Kleinrock (1979)) leíró KDE-ből következik, hogy a (3.1.1)-(3.1.3)-ban szereplő σ szegregációs ráta a szegregáció valószínűségének tekinthető, ha a sejtek osztódását Poisson folyamatként kezeljük. Tehát ésszerű feltételezni a $\sigma=e^{-\alpha N}$ (α modellparaméter) összefüggést.

A fenti meggondolások kísérleti ellenőrzése még hosszú időt vehet igénybe. A mérési adatok értékelésének problémája felveti a ROW4B program továbbfejlesztésének igényét. Ugyanis a mérések értékelése azt kívánja, hogy differenciálegyenlet paramétereit becsülni tudjuk mérési adatokból. Erre a numerikus matematika egyéb területeihez képest jóval kevesebb irodalmi anyag áll rendelkezésre. Bellman et al. (1967) az ún. kvázilinearizálás módszerét javasolta, amelyet azután többen is finomítottak (pl. Hwang-Seinfeld (1972) Kalogerakis-Luus (1983)). Stiff KDE-k paramétereinek becslése természetesen még nehezebb mint a "sima" differenciálegyenleteké; a KDE paraméterbecslése nyilván még pontatlanabb mint maga a KDE közelítő megoldása. Az újabb vizsgálatok a spline-függvények segítségével próbálják ezt a nehézséget leküzdeni (Yermakova et al (1982), Varah (1982)).

A KDE-k paramétereinek becslése nagyságrendekkel több gépidőt vesz igénybe mint csak maga a KDE megoldása. Így a numerikus továbbfejlesztés iránya adott: a pontosság növelése, illetve a futási idő csökkentése. Mivel a kettő együtt szinte biztosan nem megy, optimumot kell az adott feladathoz találni. A vizsgálatok ebben az irányban jelenleg is folynak.

IRODALOM

J.E.Bailey-A.Ollis: Biochemical Engineering Fundamentals
McGraw Hill (1984)

Ballagi-P.A.-Baranyi J.-Kováts S.-T.Illeni: Investigation of Some Simulaton Results and Mathematical Models Describing Plasmid Stability in Continuous Cultures of Recombinant Cells
(Elküldve a Biotechnolgy and Bioengineering című folyóiratnak (1989))

Baranyi J.-T.Illeni: A Simple Mathematical Model for Partition and Replication of Plasmids
(Elküldve a Mathematical Biosciences című folyóiratnak (1989))

R.Bellman-J.Jacquez-R.Kalaba-S.Schwimmer:
Quasilinearization and the Estimation of Chemical Rate Constants from Raw Kinetic Data
Mathematical Biosciences 1, 71-76 (1967)

G.Bjurel-G.Dahlquist-B.Lindberg-S.Lindé-L.Odén: Survey of Stiff Ordinary Differential Equations
Report NA 70.11, The Royal Institute of Technology, Stockholm, Sweden (1970)

A.Coddington-H.Levinson: Theory of Ordinary Differential Equations
McGraw Hill (1955)

G.G.Dahlquist: A Special Stability Problem for Linear Multistep Methods
BIT 3, 27-43 (1963)

W.H.Enright-T.E.Hull-B.Lindberg: Comparing Numerical Methods for Stiff Systems of O.D.E-s
BIT 15, 10-48 (1975)

A.G.Fredrickson-R.D.Megee III.,-H.M.Tsuchiya: Mathematical Models for Fermentation Processes
Chemical Engineering Department Report
University of Minnesota, Minneapolis (1970)

G.W.Gear: Numerical Initial Value Problems in Ordinary Differential Equations
Prentice Hall (1971)

B.A.Gottwald-G.Wanner: 'A Reliable Rosenbrock Integrator for Stiff Differential Equations
Computing 26, 355-360 (1981)

W.Greenhalf-D.C.J.Gardner-S.G.Oliver: A Mathematical Model for Plasmid Replication and Distribution in Microbial Populations
Biotechnology Letters 11, 93-98 (1989)

F.G.Heineken-H.M.Tsuchiya-R.Aris: On the Pseudo-Steady State Hypothesis of Biochemical Kinetics
Matehamatical Biosciences 1, 95-113 (1967)

H.Hemker-C.Hess: Analysis and Simulation of Biochemical Systems
North Holland (1972)

M. Hwang-J. H. Seinfeld: A New Algorithm for the Estimation of Parameters in Ordinary Differential Equations
AIChE Journal 18, 90-93 (1972)

T. Imanaka-S. Aiba: A Perspective on the Application of Genetic Engineering; Stability of Recombinant Plasmid
Ann. NY. Acad. Sci. 369, 1-14 (1981)

P. Kaps-P. Rentrop: Generalized Runge-Kutta Methods of Order Four with Stepsize Control for Stiff Ordinary Differential Equations
Numerische Mathematik 33, 55-68 (1979)

P. Kaps-P. Rentrop: Application of a Variable-Order Semi Implicit Runge-Kutta Method to Chemical Models
Computers & Chemical Engineering 8, 393-396 (1984)

P. Kaps-G. Wanner: A Study of Rosenbrock-Type Methods of High Order
Numerische Mathematik 38, 279-298 (1981)

S. H. Kim-D. Y. Ryu: Instability Kinetics of trp Operon Plasmid ColEl-trp in Recombinant E. Coli Mv12 (pVH5) and Mv12-trp T(pVH5)
Biotechnology and Bioengineering 26, 497-502 (1984)

L. Kleinrock: Sorbanállás, kiszolgálás
Műszaki Könyvkiadó (1979)

D. A. Lauffenburger: Stability of Colicin Plasmids in Continuous Culture; Mathematical Model and Analysis
Biotechnology Progress 1, 53-59 (1985)

Móricz F.: Numerikus analízis
Egyetemi jegyzet, JATE, Szeged, (1981)

S.J.Parulekar-Y.K.Chang-J.M.Modak-H.C.Lim: Analysis of Continuous, Culture of Recombinant Methylotrophs
Biotechnology and Bioengineering 29, 911-923 (1987)

H.H.Rosenbrock: Some General Implicit Processes for the Numerical Solution of Differential Equations
Computing Journal 5, 329-331 (1963)

V.N.Satayagal-P.Agrawal: A Generalized Model of Plasmid Replication
Biotechnology and Bioengineering 33, 1135-1144 (1989)

J.Stoer-R.Bulirsch: Introduction to Numerical Analysis
Springer-Verlag (1980)

H.M.Tsuchiya-A.G.Frederickson-R.Aris: Dynamics of Microbial Cell Populations
in: Advances in Chemical Engineering
McGraw Hill (1966)

T.B.Young-D.F.Bruley-H.R.Bungay III.: A Dynamic Mathematical Model of the Chemostat
Biotechnology and Bioengineering 12, 747-769 (1970)

Valkó P.-Vajda S.: Műszaki tudományos feladatok megoldása személyi számítógéppel
Műszaki Könyvkiadó (1987)

O.B.Widlund: A Note on Unconditionally Stable Linear Multistep Methods
BIT 7, (1967)

A.Wolfbrandt: A Study of Rosenbrock Processes with Respect to Order Conditions and Stiff Stability
Report 77.01, Göteborg University, Sweden (1977)

A szerzőnek e témához nem kapcsolódó publikációi

Baranyi J.: Néhány tétel konvex függvényekről
GAMF Közlemények (1983)

Dr. Kodácsy J. - Baranyi J.: A szerszáméltartam meghatározása változó forgácsolási paraméterekkel
Gépgyártástechnológia 26, 124-129 (1986)

N. Masszi - Baranyi J: A Linear and a Nonlinear Programming Approach to Recursive Digital Filter Design with Linear Phase
Uppsala University Reports (1989)