Povzetek

Večina zakonitosti, ki jih odkrijejo obstoječi programi za avtomatsko odkrivanje zakonitosti, opisuje statične situacije, kjer je modelirani sistem dosegel ravnotežje. Zakonitosti imajo obliko algebraičnih enačb, ki povezujejo spremenljivke modeliranega sistema.

V diplomskem delu obravnavamo problem avtomatskega modeliranja dinamičnih sistemov. Ker je problem modeliranja dinamičnih sistemov zelo pogost, bi uspešna metoda za avtomatsko modeliranje le-teh imela daljnosežne posledice.

Obstoječe metode za avtomatsko odkrivanje zakonitosti smo prilagodili tako, da bodo uporabne pri odkrivanju zakonitosti v dinamičnih domenah. Tako prilagojene metode smo implementirali v programu LAGRANGE, ki odkriva množico diferencialnih in algebraičnih enačb na osnovi časovnega zaporedja meritev spremenljivk dinamičnega sistema.

V dosedanjih poskusih je LAGRANGE uspešno rekonstruiral sedem modelov dinamičnih sistemov iz dinamike tekočin, kemičnih reakcij, populacijske rasti in mehanske dinamike. Med njimi so tudi nelinearni modeli oscilatornih procesov ter dinamični model inverznega nihala, ki je standardni problem za testiranje različnih metod vodenja dinamičnih sistemov.

Kazalo

1.

Uvod

1.1 Motivacija in cilji

Ker je problem odkrivanja zakonitosti obnašanja oz. modeliranja dinamičnih sistemov zelo pogost, bi uspešna metoda za avtomatsko modeliranje le-teh imela daljnosežne posledice. Večina obstoječih metod iz področja avtomatske regulacije predpostavlja znano strukturo modela oz. obliko diferencialnih enačb, ki opisujejo dinamiko sistema. Na osnovi izmerjenega obnašanja modeliranega sistema poskušajo omenjene metode določiti optimalne vrednosti parametrov v enačbah dinamike.

Čeprav obstaja več različnih programov za avtomatsko odkrivanje zakonitosti (BACON [Langley et al. 1987], FAHRENHEIT [Żytkow and Zhu 1991], ABACUS [Falkenheiner and Michalski 1990]), se nobeden ni uveljavil kot orodje za modeliranje realnih sistemov. Vzrok za to je tudi nezmožnost teh programov za odkrivanje zakonitosti v dinamičnih domenah [Bielecki 1992]. Večina zakonitosti, ki so jih taki sistemi odkrili, opisuje statične situacije, kjer je opazovani sistem dosegel ravnotežje. Te zakonitosti imajo večinoma obliko algebraičnih enačb, ki povezujejo opazovane spremenljivke.

Metode, ki jih uporabljajo obstoječi programi za avtomatsko odkrivanje zakonitosti, so neustrezne v kontekstu dinamičnih sistemov iz dveh razlogov. Prvi je ta, da pri dinamičnih sistemih nimamo eksperimentalne kontrole nad posameznimi spremenljivkami sistema. Drugi (ki delno izvira iz prvega) pa je, da ni eksplicitnega razlikovanja med odvisnimi in neodvisnimi spremenljivkami dinamičnega sistema.

V diplomskem delu smo želeli metode, uporabljene v obstoječih programih za avtomatsko odkrivanje zakonitosti, prilagoditi tako, da bodo uporabne tudi pri odkrivanju zakonitosti v dinamični domeni. Na osnovi teh metod smo želeli implementirati program, ki bi bil uporabno orodje za modeliranje dinamičnih sistemov.

1.2 Rezultati dela

V okviru tega dela sem razvil program LAGRANGE, ki avtomatsko odkriva zakonitosti v dinamični domeni. Program je implementiran v programskem jeziku C pod operacijskim sistemom UNIX in teče na delovni postaji Sun SPARC IPC.

Zmožnost programa sem pokazal s poskusi avtomatskega modeliranja sedmih dinamičnih sistemov iz dinamike tekočin, kemičnih reakcij, populacijske rasti in mehanske dinamike. V vseh primerih je LAGRANGE uspešno rekonstruiral originalne modele, med katerimi so tudi nelinearni modeli oscilatornih procesov ter dinamični model inverznega nihala, ki je standardni problem za testiranje različnih metod vodenja dinamičnih sistemov.

1.3 Pregled vsebine ostalih poglavij

Drugo poglavje vsebuje definicijo osnovnih pojmov iz področja dinamičnih sistemov in opis različnih pristopov modeliranja le-teh.

V tretjem poglavju je podana definicija naloge avtomatskega odkrivanja zakonitosti. Z opisom delovanja programov BACON in FAHRENHEIT opisujem tipične metode, ki se uporabljajo pri reševanju te naloge.

Četrto poglavje zajema podrobni opis algoritma LAGRANGE in njegove implementacije, analizo njegove časovne kompleksnosti ter primer poteka izvajanja algoritma na enostavni dinamični domeni.

Opis in rezultate poskusov, ki sem jih izvedel s programom LAGRANGE, podajam v petem poglavju.

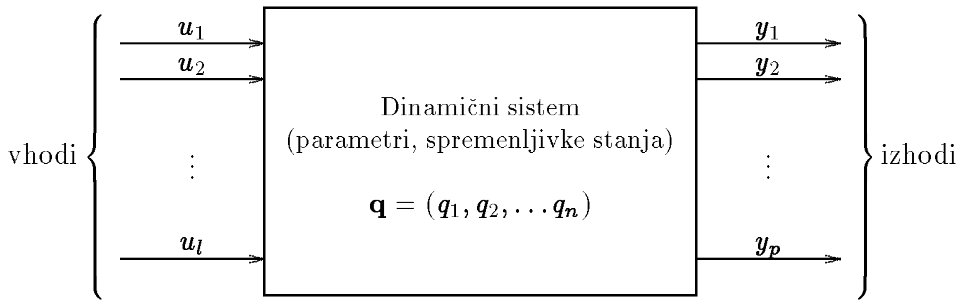
Šesto poglavje povzema rezultate diplomskega dela in nakazuje smernice za nadaljnje delo.

V dodatku A je podan formalni opis vhodnih datotek programa LAGRANGE, v dodatku B pa vhodne in izhodne datoteke programa LAGRANGE za domene uporabljene pri poskusih iz petega poglavja.

2.

Modeliranje dinamičnih sistemov

Dinamični sistem lahko shematično opišemo kot kaže Slika 2.1 [Shearer and Kulakowski 1990]. Vhodi oz. vhodne spremenljivke izvirajo iz okolja in niso direktno odvisne od dogajanja znotraj dinamičnega sistema. Spremenljivke, s katerimi dinamični sistem vpliva na okolje, so izhodne spremenljivke, ali na kratko izhodi. Stanje dinamičnega sistema je definirano z minimalno množico spremenljivk, za katere velja, da njihove vrednosti v času $t = t_0$ ter vrednosti vhodnih spremenljivk za $t > t_0$, enolično določajo obnašanje dinamičnega sistema v času $t \geq t_0$. Elementi minimalne množice, ki opisuje stanje, so spremenljivke stanja dinamičnega sistema.



Slika 2.1: Dinamični sistem

Model je poenostavljena predstavitev realnega sistema. Modeli so lahko fizični (pomanjšane kopije realnih sistemov in prototipi), ali pa abstraktno izražajo relacije med spremenljivkami modeliranega sistema. V slednjem primeru lahko dinamični sitem opišemo z besedami, z diagrami poteka in grafi, ali pa z matematičnimi relacijami. Matematični modeli so najbolj razširjeni predvsem zaradi natančnega izražanja relacij in splošnosti. Obstajata dve obliki matematičnih modelov dinamičnih sistemov: vhodno-izhodni modeli in modeli stanja. Čeprav sta obe obliki enakovredni glede vsebovane informacije o dinamičnem sistemu, so modeli stanja bolj primerni pri modeliranju in simulaciji [Shearer and Kulakowski 1990]. Modeli stanja imajo obliko sistema diferencialnih enačb prvega reda:

$$\begin{aligned}
\dot{q}_1 &= f_1(q_1, q_2, \ldots q_n, u_1, u_2, \ldots u_l, t) \\
\dot{q}_2 &= f_2(q_1, q_2, \ldots q_n, u_1, u_2, \ldots u_l, t) \\
&\vdots \\
\dot{q}_n &= f_n(q_1, q_2, \ldots q_n, u_1, u_2, \ldots u_l, t).
\end{aligned}$$

Izhodne spremenljivke dinamičnega sistema izrazimo iz vhodov in spremenljivk stanja s sistemom algebraičnih enačb:

$$\begin{aligned}
y_1 &= g_1(q_1, q_2, \ldots q_n, u_1, u_2, \ldots u_l, t) \\
y_2 &= g_2(q_1, q_2, \ldots q_n, u_1, u_2, \ldots u_l, t) \\
&\vdots \\
y_n &= g_n(q_1, q_2, \ldots q_n, u_1, u_2, \ldots u_l, t).
\end{aligned}$$

Modeliranje je proces poenostavljanja realnega dinamičnega sistema z matematičnim modelom in je lahko teoretično ali eksperimentalno [Karba 1993].

Pri teoretičnem modeliranju razčlenimo dinamični sitem na čimbolj enostavne podsisteme, katerih medsebojne relacije potem določamo na osnovi ravnotežnih zakonov iz ustreznega področja. Tako dobimo sistem diferencialnih enačb, ki predstavljajo model dinamičnega sistema z določeno strukturo in določenimi parametri. Ker je ta model pogosto obsežen in zapleten, ga pred nadaljno uporabo poenostavimo.

Pri eksperimentalnem modeliranju, ki ga imenujemo tudi identifikacija, dobimo model na osnovi izmerjenega obnašanja modeliranega dinamičnega sistema. Pri tem predstavljajo osnovo za modeliranje tudi a-priori znanja o sistemu, ki jih pridobimo s teoretično analizo, ali iz predhodnih meritev in opazovanj. Bistveno znanje o dinamičnem sistemu je njegova struktura, ker večina obstoječih identifikacijskih metod predpostavlja znano stukturo modela oz. diferencialnih enačb, ki opisujejo dinamiko sistema. Na osnovi izmerjenega obnašanja modeliranega sistema poskušajo omenjene metode določiti optimalne vrednosti parametrov v enačbah dinamike.

V praksi se največkrat uporablja kombinirano modeliranje, ki skuša izkoristiti prednosti obeh omenjenih pristopov. Pri tem strukturo modela v glavnem določamo s pomočjo teoretičnega pristopa, medtem ko parametre ocenimo z metodami identifikacije dinamičnih sistemov.

3.

Avtomatsko odkrivanje zakonitosti

Avtomatsko učenje je podpodročje umetne inteligence, ki preučuje metode za pridobivanje novega in organiziranje obstoječega znanja [Carbonell and Langley 1987]. Različne oblike učenja se vrstijo od učenja po nasvetu (*learning by being told*) do učenja z odkrivanjem (*learning by discovery*) [Bratko 1990]. Značilnost učenja z odkrivanjem je v tem, da učni program nima na razpolago nobenega učitelja, ki bi usmerjal učni proces. Učenje (odkrivanje) poteka samo na osnovi rezultatov poizkusov in opazovanja.

Raziskovalci na področju učenja z odkrivanjem kot neusmerjenega učenja pristopajo problemu odkrivanja iz treh različnih vidikov: pojmovno razvrščanje v skupine (*conceptual clustering*), oblikovanje pojasnjevalnih teorij (*formulation of explanation theories*) in odkrivanje opisnih zakonov (*induction of descriptive laws* ali *empirical discovery*) [Langley and Żytkow 1989]. V nadaljevanju se bomo omejili na tretji vidik odkrivanja, s katerim se ukvarjajo programi za avtomatsko odkrivanje zakonitosti. Nalogo avtomatskega odkrivanja zakonitosti lahko neformalno definiramo z naslednjim predpisom:

- **Podano** imamo množico opazovanj (podatki)
- **Poiskati** moramo eno ali več zakonitosti, ki veljajo za te podatke

Obstajajo različne statistične metode, ki bolj ali manj uspešno rešujejo to nalogo, vendar niso uveljavljene kot izključno orodje programov za avtomatsko odkrivanje zakonitosti iz dveh poglavitnih razlogov [Langley and Żytkow 1989]. Prvi je ta, da znanstveniki ne uporabljajo samo statističnih metod, drugi pa je, da hevristične metode lahko obravnavajo večji razred numeričnih zakonitosti, kot obstoječe statistične metode.

V nadaljevanju bom podal pregled programov BACON in FAHRENHEIT, ki avtomatsko odkrivata zakonitosti z uporabo hevrističnih metod. Podrobno so opisane metode, ki jih programa BACON in FAHRENHEIT uporabljata. Te, so namreč najbližje metodam, ki jih uporablja program LAGRANGE, razvit v okviru diplomskega dela.

3.1 Pregled programa BACON

Program za avtomatsko odkrivanje zakonitosti BACON [Langley et al. 1987] je poimenovan po filozofu znanosti iz šestnajstega stoletja Sir Francisu Baconu, ki je preučeval proces znanstvenega odkrivanja. Obstaja šest različic tega programa (BACON.1 do BACON.6). V tem pregledu bomo opisali različico BACON.3, ki zajema najbolj pomembne ideje programa BACON.

Vhod v program BACON.3 je množica vrednosti neodvisnih in odvisnih spremenljivk modeliranega pojava. Neodvisne spremenljivke so tiste, katerih vrednosti lahko neposredno spreminjamo. Spreminjanje vrednosti neodvisnih spremenljivk povzroči ustrezno spremembo odvisnih spremenljivk. BACON.3 odkriva algebraične enačbe, ki povezujejo vrednosti odvisnih in neodvisnih spremenljivk modeliranega pojava.

BACON.3 odkriva enostavne zakonitosti s pomočjo štririh hevristik za odkrivanje tendenc in konstantnosti v podatkih, ki jih lahko zapišemo v obliki **če-potem** pravil [Langley et al. 1987]:

1. **if** $Y = c$ za veliko število podatkov
   then postavi hipotezo $Y = c$

2. **if** $Y = aX + b$ za veliko število podatkov
   then postavi hipotezo $Y = aX + b$

3. **if** X narašča, takrat kadar Y pada
   then uvedi novo spremenljivko $X \cdot Y$

4. **if** X narašča hkrati z Y
   then uvedi novo spremenljivko X/Y

Hevristiki 3 in 4, ki uvajata nove spremenljivke, BACON.3 ponavljajoče uporablja na podatkih, dokler ni izpolnjen vsaj eden od pogojev iz prvih dveh hevristik. BACON.1 je samo s pomočjo teh preprostih hevristik odkril Ohmov zakon za električni krog, Keplerjev tretji zakon o gibanju planetov okrog Sonca ter zakon prostega pada.

Metoda opisnih ravni (*levels of description*), ki omogoča programu BACON.3 odkrivanje bolj kompleksnih zakonitosti, sloni na predpostavki, da imamo eksperimentalno kontrolo nad posameznimi neodvisnimi spremenljivkami. Na najnižji opisni ravni program dovoli spreminjanje samo ene neodvisne spremenljivke ter išče enostavno zakonitost, ki povezuje odvisno in spreminjajočo se neodvisno spremenljivko. Ko BACON.3 odkrije tako zakonitost, preide na naslednjo opisno raven, na kateri dovoli spreminjanje še ene neodvisne spremenljivke. Parametri zakonitosti odkritih na prejšnji ravni postanejo odvisne spremenljivke na tekoči ravni. BACON.3 uvaja nove opisne ravni toliko časa, dokler ne najde zakonitosti, ki vsebuje vse neodvisne spremenljivke.

Opisana metoda je zelo močna in omogoča programu BACON.3 odkrivanje različnih fizikalnih in kemičnih zakonov, odpove pa v primerih, ko nimamo eksperimentalne kontrole nad posameznimi spremenljivkami.

Delovanje programa BACON.3 bomo pokazali na primeru odkrivanja poenostavljenega Blackovega toplotnega zakona [Langley and Żytkow 1989]:

$$T_f = \frac{M_1 T_1 + M_2 T_2}{M_1 + M_2} \tag{3.1}$$

pri čemer sta T_1 in T_2 temperaturi dveh vzorcev iste snovi, M_1 in M_2 ustrezni masi ter T_f končna temperatura, ki nastane potem ko vzorce pomešamo.

BACON.3 začne tako, da zahteva rezultate različnih poskusov, v katerih se vrednosti M_1, M_2 in T_1 ne spreminjajo. S spreminjanjem vrednosti neodvisne spremenljivke T_2 se ustrezno spremeni odvisna spremenljivka T_f. Dobljeni rezultati poskusov opisujejo odvisnost T_f od T_2 v primeru konstantnosti ostalih nedvisnih spremenljivk. Iz teh podatkov BACON.3 s pomočjo druge hevristike, postavi hipotezo:

$$T_f = aT_2 + b. \tag{3.2}$$

Parametra a in b postaneta odvisni spremenljivki na drugi opisni ravni, na kateri BACON.3 zahteva rezultate poskusov, v katerih se spreminja tudi T_1, ter najde dve odvisnosti $a = c$ in $b = dT_1$. Če ti dve odvisnosti upoštevamo skupaj z enačbo 3.2, dobimo:

$$T_f = cT_2 + dT_1. \tag{3.3}$$

Na tretji opisni ravni spreminjamo maso M_2 in opazujemo njen vpliv na c in d. Ker ni linearne odvisnosti med c oz. d in M_2, BACON.3 uporabi tretjo in četrto hevristiko ter vpelje novi spremenljivki c/M_2

in dM_2. V naslednjem koraku uporabe hevristik najdemo zakonitosti $M_2 = e(M_2/c) + f$ in $dM_2 = gd + h$. Iz njih in 3.3 dobimo:

$$T_f = \frac{eM_2}{M_2 - f} T_2 + \frac{h}{M_2 - g} T_1. \tag{3.4}$$

Na zadnjem opisnem nivoju spreminjamo M_1 in BACON.3 najde odvisnosti $e = 1.0$, $f = -M_1$, $g = -M_1$ ter $h = M_1$. Če upoštevamo te odvisnosti skupaj s 3.4, dobimo poenostavljen Blackov toplotni zakon iz enačbe 3.1.

3.2 Pregled programa FAHRENHEIT

FAHRENHEIT [Żytkow and Zhu 1991], [Langley and Żytkow 1989] je razširitev programa BACON v smeri odkrivanja zakonitosti, ki ne veljajo v celotnem območju, temveč samo v podmnožici območja vrednosti spremenljivk. Podobno kot pri programu BACON, je vhod v FAHRENHEIT množica neodvisnih spremenljivk $X_1, X_2, \ldots X_n$ ter odvisna spremenljivka Y. Zakonitosti imajo obliko algebraičnih enačb, ki povezujejo neodvisne z odvisno spremenljivko. Vsaki enačbi FAHRENHEIT pridruži pogoj, ki določa območje njene veljavnosti.

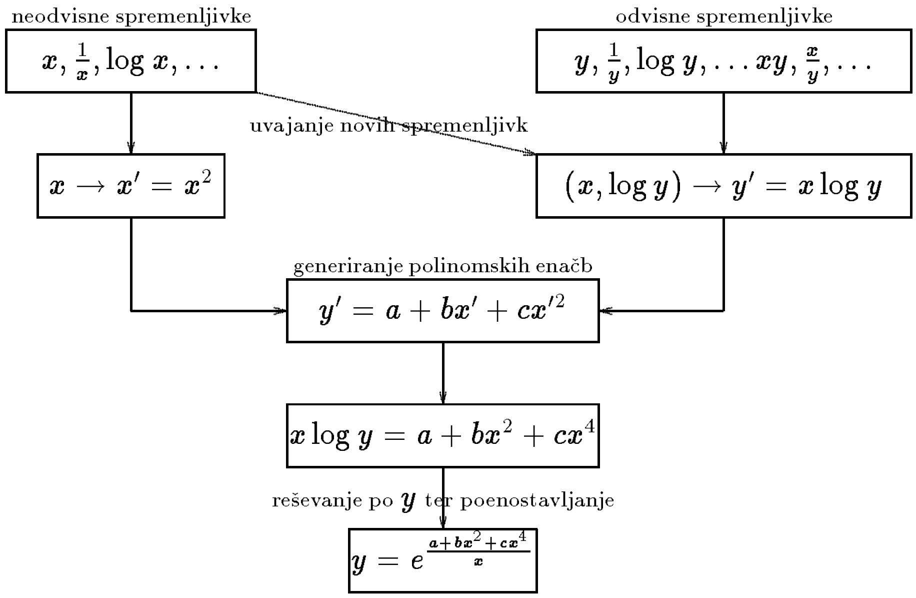
FAHRENHEIT odkriva kompleksne zakonitosti z metodo opisnih ravni programa BACON.3. Torej odkrivanje kompleksnih zakonitosti ponovno sloni na predpostavki, da imamo popolno eksperimentalno kontrolo nad posameznimi neodvisnimi spremenljivkami.

Za odkrivanje enostavnih zakonitosti na vsaki opisni ravni uporablja FAHRENHEIT modul EF (EQUATION FINDER) [Zembowitz and Żytkow 1992]. Enostavne zakonitosti povezujejo odvisno spremenljivko y in spreminjajočo se neodvisno spremenljivko x. Vhod v EF je množica točk (x_i, y_i, σ_i), $i = 1, 2, \ldots N$, pri čemer so x_i in y_i ustrezne vrednosti neodvisne in odvisne spremenljivke, ter σ_i ocene za napake vrednosti y_i. Izhod je množica algebraičnih enačb oblike $y = f(x)$, ki se dobro prilegajo podanim podatkom. Delovanje EF lahko razdelimo na dva koraka.

V prvem koraku EF uvaja nove spremenljivke s transformacijami, ki jih poda uporabnik. Transformacije uporablja rekurzivno, najprej nad osnovnimi spremenljivkami x in y:

$$\begin{aligned} x &\rightarrow x' = T_1(x) \\ (y, \sigma) &\rightarrow (y' = T_2(x, y), \sigma' = \left|\frac{\partial T_2}{\partial y}\right| \sigma), \end{aligned}$$

zatem pa nad že prej uvedenimi spremenljivkami. EF uvaja nove spremenljivke do uporabniško določene globine. Globina spremenljivke je enaka številu transformacij, ki jih uporabimo pri njenem uvajanju. Privzeta množica transformacij je $\mathcal{T} = \{^{-1}, \log, \exp, \sqrt{}, /, *\}$, uporabnik pa lahko doda množici $\mathcal{T}$ poljubne transformacije. Tako ima na primer, spremenljivka x globino 0 ter spremenljivki $x \log y$ in $x^2 e^x \sqrt{y}$ globini 2 (log in $*$) in 5 (exp, $\sqrt{}$ in tri $*$).



Slika 3.1: Primer delovanja modula EF

EF v drugem koraku generira polinomske enačbe oblike $y' = p_q(x')$ za vse pare spremenljivk (x', y') in ocenjuje njihovo signifikantnost glede na podane meritve. Če je signifikanost nad uporabniško določenim pragom, se signifikantna enačba z reševanjem po y transformira v obliko $y = f(x)$ in doda množici odkritih enačb. Maksimalno stopnjo polinomov poda uporabnik. EF določa konstantne koeficiente polinomskih enačb EF z metodo najmanjših kvadratov, ki minimizira uteženo napako:

$$E = \sum_{i=1}^{N} \left(\frac{y'_i - p_q(x'_i)}{\sigma_i} \right)^2, \quad p_q(x'_i) = \sum_{j=1}^{q} a_j (x'_i)^{j-1}.$$

Za ocenjevanje signifikantnosti generiranih enačb uporablja EF stopnjo zaupanja, ki se izračuna na osnovi vrednosti utežene napake E ter verjetnosne porazdelitve χ^2 z $N - q$ prostostnimi stopnjami.

Slika 3.1 prikazuje delovanje modula EF na primeru odkrivanja enačbe $y = e^{\frac{a+bx^2+cx^4}{x}}$ [Zembowitz and Żytkow 1992].

4.

Lagrange

Nalogo avtomatskega odkrivanja zakonitosti v dinamični domeni neformalno definiramo na naslednji način. Na osnovi izmerjenega obnašanja dinamičnega sistema v nekem končnem časovnem intervalu je treba poiskati matematični model, ki opisuje dinamični sistem. Lagrange na vhodu zahteva meritve spremenljivk dinamičnega sistema v ekvidistančnih časovnih točkah (Tabela 4.1). Matematični model, ki ga odkrije Lagrange, je množica diferencialnih in algebraičnih enačb.

	spremenljivke din. sistema			
čas	X_1	X_2	$\ldots$	X_n
t_0	x_{11}	x_{21}	$\ldots$	x_{n1}
$t_0 + h$	x_{12}	x_{22}	$\ldots$	x_{n2}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$t_0 + (N-1)h$	x_{1N}	x_{2N}	$\ldots$	x_{nN}

Tabela 4.1: Vhodni podatki za program Lagrange

V nadaljevanju tega poglavja je podan formalni in neformalni opis algoritma Lagrange, primer poteka izvajanja na enostavni dinamični domeni, analizo časovne kompleksnosti algoritma in podrobnosti implementacije.

4.1 Opis algoritma

Algoritem LAGRANGE, katerega formalni opis je podan v Tabeli 4.2, v grobem razdelimo na tri korake:

1. Računanje časovnih odvodov
2. Uvajanje novih spremenljivk
3. Iskanje enačb z linearno regresijo

4.1.1 Računanje časovnih odvodov

V prvem koraku LAGRANGE izračuna časovne odvode spremenljivk dinamičnega sistema do uporabniško določene stopnje o. Privzeta vrednost za o je 1. Časovne odvode računamo z numeričnim odvajanjem po formuli [Bohte 1991] (str. 73):

$$\dot{x}_i = \frac{1}{12h}(x_{i-2} - 8x_{i-1} + 8x_{i+1} - x_{i+2}), \tag{4.1}$$

pri čemer so x_i in $\dot{x}_i$ ustrezne vrednosti spremenljivke X in njenega odvoda $\dot{X}$ v točkah $t_0 + (i-1)h$. Če je δ napaka pri merjenju spremenljivke X, potem je napaka odvoda $\dot{X}$, izračunanega po formuli 4.1, enaka

$$\frac{\delta}{h} + \frac{1}{30}h^4 M_5. \tag{4.2}$$

Obrazec za napako kaže na to, da moramo biti previdni pri določanju časovnega koraka h med zaporednimi meritvami. Če poznamo napako δ, ki jo zagrešimo pri merjenju spremenljivke X, in zgornjo mejo M_5 za peti odvod X na intervalu $[t_0, t_0 + (N-1)h]$, potem lahko izračunamo optimalni h, za katerega je napaka iz obrazca 4.2 najmanjša. V primerih, ko je možno bolj natančno merjenje odvodov spremenljivk dinamičnega sistema, kot pa njihovo računanje po zgornji formuli, podamo meritve odvodov na vhodu programa LAGRANGE in jih ne računamo numerično.

4.1.2 Uvajanje novih spremenljivk

LAGRANGE uvaja nove spremenljivke na dva načina.

Pri prvem načinu uporablja informacijo o merskih enotah spremenljivk dinamičnega sistema. Če je merska enota spremenljivke X radian, uvedemo dve novi spremenljivki $sinX$ in $cosX$.

$$
\text{1.}\quad \left\{ \begin{array}{l}
V := V_S \\
D_0 := V_S \\
\textbf{for } i := 1 \textbf{ to } o \textbf{ do} \\
\quad D_i := \emptyset \\
\quad \textbf{forall } x \in D_{i-1} \textbf{ do} \\
\quad\quad D_i := D_i \cup \{\frac{d}{dt}x\} \\
\quad V := V \cup D_i
\end{array}\right.
$$

$$
\text{2.}\quad \left\{ \begin{array}{l}
\textbf{forall } x \in V_S \wedge \dim(x) = [\mathrm{rad}] \textbf{ do} \\
\quad V := V \cup \{\sin x, \cos x\} \\
V_1 := V \\
\textbf{for } l := 2 \textbf{ to } d \textbf{ do} \\
\quad V_l := \emptyset \\
\quad \textbf{forall } (x, y) \in V_0 \times V_{l-1} \textbf{ do} \\
\quad\quad V_l := V_l \cup \{x \cdot y\} \\
\quad\quad (*\ \text{duplikati spremenljivk niso dovoljeni}\ *) \\
\quad V := V \cup V_l
\end{array}\right.
$$

$$
\text{3.}\quad \left\{ \begin{array}{l}
M := \emptyset \\
\textbf{for } i := 1 \textbf{ to } r \textbf{ do} \\
\quad \textbf{forall } R \in \mathcal{P}(V) \wedge |R| = i \textbf{ do} \\
\quad\quad \text{izberi odvisno spremenljivko } y \in R \\
\quad\quad R := R \setminus \{y\} \\
\quad\quad \textbf{if } LR(y, R) \text{ je signifikantna } \textbf{then} \\
\quad\quad\quad R' := R \\
\quad\quad\quad \textbf{forall } x \in R \wedge |c_x| < \sigma_{c_x} \textbf{ do} \\
\quad\quad\quad\quad R' := R' \setminus \{x\} \\
\quad\quad\quad \textbf{if } (R' \neq R) \textbf{ then} \\
\quad\quad\quad\quad \text{ponovi } LR(y, R' \setminus \{y\}) \\
\quad\quad\quad (*\ \text{zbriši nepomembne neodvisne}*) \\
\quad\quad\quad (*\ \text{spremenljivke iz enačbe}\ *) \\
\quad\quad\quad M := M \cup \{y = c_0 + \sum_{x \in R' \setminus \{y\}} c_x x\} \\
\quad\quad\quad (*\ \text{dodaj enačbo v model}\ *)
\end{array}\right.
$$

Tabela 4.2: Formalni opis algoritma LAGRANGE

Pri drugem načinu uvajamo nove spremenljivke z množenjem starih. Pri tem globina spremenljivk postopoma narašča. Maksimalno globino novih spremenljivk d določi uporabnik, privzeta vrednost je 2. Globina spremenljivke $Y = \prod_{i=1}^{n} X_i^{\alpha_i}$ $(\alpha_i \geq 0)$ je enaka $\sum_{i=1}^{n} \alpha_i$. Tako ima na primer, stara spremenljivka X_1 globino 1 ter novi spremenljivki $X_1 X_2$ in ${X_1}^2 {X_2}^3$ globini 2 in 5.

Pri dodajanju novih spremenljivk z množenjem se lahko zgodi, da isto spremenljivko dobimo na več načinov. Tako lahko spremenljivko $X_1{X_2}^2$ dobimo kot produkt spremenljivk X_1 in ${X_2}^2$, ali pa spremenljivk X_2 in X_1X_2. LAGRANGE uvede take spremenljivke samo enkrat ter sproti poskrbi za brisanje duplikatov.

4.1.3 Iskanje enačb z linearno regresijo

LAGRANGE v tretjem koraku generira vse možne linearne enačbe, ki vsebujejo največ r spremenljivk in ocenjuje njihovo signifikantnost glede na podane meritve. Če je signfikantnost nad pragom, ki ga določi uporabnik, dodamo enačbo v model. Vrednost r določi uporabnik, privzeta vrednost je 3.

Konstantne člene linearnih enačb računamo z linearno regresijo [Volk 1958]. Linearna regresija na osnovi meritev m neodvisnih spremenljivk $X_1, X_2, \ldots, X_m$ ter ene odvisne Y, določi koeficiente $c_0, c_1, \ldots, c_m$ tako, da je kvadratna napaka

$$E = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2, \quad \hat{y}_i = c_0 + \sum_{j=1}^{m} c_j x_{ji}$$

čim manjša. V zgornji formuli so $y_i, x_{1i}, \ldots, x_{mi}$ ustrezne meritve odvisne spremenljivke Y ter neodvisnih spremenljivk $X_1, X_2, \ldots, X_m$ v točkah $t_0 + (i-1)h$. Če definiramo

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{21} & \ldots & x_{m1} \\ 1 & x_{12} & x_{22} & \ldots & x_{m2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{1N} & x_{2N} & \ldots & x_{mN} \end{bmatrix}, \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \mathbf{c} = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_m \end{bmatrix},$$

potem vektor $\mathbf{c}$ izračunamo po naslednji formuli:

$$\mathbf{c} = \mathbf{V}\mathbf{y}, \quad \mathbf{V} = (\mathbf{X}^T\mathbf{X})^{-1}.$$

Iz diagonalnih elementov matrike $\mathbf{V}$ lahko izračunamo odstopanja $\sigma_{c_0}, \sigma_{c_1}, \ldots, \sigma_{c_m}$ koeficientov linearne enačbe po formuli [Volk 1958] (str. 260–264):

$${\sigma_{c_j}}^2 = v_{(j+1)(j+1)} \frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{N - m}.$$

Odstopanja koeficientov uporabljamo pri brisanju nepomembnih neodvisnih spremenljivk iz enačb, ki jih dodajamo v model.

LAGRANGE uporablja dva kriterija za ocenjevanje signifikantnosti enačb dobljenih z linearno regresijo:

1. Korelacijski koeficient R [Volk 1958] (str. 274):

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \overline{y})^2}, \quad \overline{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$$

   R zavzame vrednosti iz intervala $[0,1]$. Večja vrednost R pomeni večjo signifikantnost enačbe in obratno. Enačbo dodamo v model, če je $R > 1 - t_R$. Prag t_R določi uporabnik, privzeta vrednost je 0.01.

2. Standardizirano odstopanje S:

$$S^2 = \frac{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\overline{y}^2}$$

   S zavzame vrednosti iz intervala $[0,\infty)$. Čim manjši je S, tem večja je signifikantnost enačbe in obratno. Enačbo dodamo v model, če je $S < t_S$. Prag t_S določi uporabnik, privzeta vrednost je 0.01. Pri zgornji formuli za S lahko nastanejo problemi z numeričnim računanjem v primeru, ko je $\overline{y} \doteq 0$. Zato smo to formulo spremenili na naslednji način:

$$S^2 = \frac{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\overline{y}^2 + e^{-\overline{y}^2}}$$

   Člen $e^{-\overline{y}^2}$ bistveno spremeni vrednost S le pri $\overline{y}^2 < 1$. Takrat je namreč, S približno enak nestandardiziranemu odstopanju ($S^2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$). Za ostale vrednosti $\overline{y}$ je vpliv tega člena zanemarljiv.

Če je linearna enačba oblike $y = c_0$ (ni neodvisnih spremenljivk, $m = 0$), prvi kriterij nima smisla, ker je za te enačbe vedno $R = 0$. Signifikantnost teh enačb ocenjujemo samo z drugim kriterijem. Za ocenjevanje signifikantnosti ostalih enačb Lagrange lahko uporablja enega od zgornjih kriterijev ali pa njihovo kombinacijo.

Preden dodamo signifikantno enačbo v model, iz nje zbrišemo tiste neodvisne spremenljivke X_i, za katere velja $|c_i| < \sigma_{c_i}$. Brisanje nekaj neodvisnih spremenljivk vpliva na vrednosti koeficientov ostalih spremenljivk v enačbi. Zato v tem primeru linearno regresijo ponovimo s preostalimi neodvisnimi spremenljivkami.

4.2 Primer

Algoritem iz Tabele 4.2 bomo demonstrirali na primeru odkrivanja zakonitosti za dinamični sitem sožitja volkov in zajcev (glej Poglavje 5.3.2). Spremenljivke tega dinamičnega sistema sta velikosti populacij volkov in zajcev N_1 in N_2 ($V_S = \{N_1, N_2\}$). Parametrom o, d in r smo dodelili privzete vrednosti 1, 2 in 3.

V prvem koraku algoritma numerično izračunamo časovne odvode $\dot{N}_1$ in $\dot{N}_2$ spremenljivk dinamičnega sistema.

Na začetku drugega koraka je množica vseh spremenljivk $V = \{N_1, N_2, \dot{N}_1, \dot{N}_2\}$. V drugem koraku ne uvajamo nobenih spremenljivk oblike $sinX$ in $cosX$, ker nobena spremenljivka iz S ni merjena v radianih. Z množenjem spremenljivk iz V dobimo deset novih spremenljivk globine 2 , kar da $V = \{N_1, N_2, \dot{N}_1, \dot{N}_2, {N_1}^2, N_1N_2, N_1\dot{N}_1, N_1\dot{N}_2, {N_2}^2, N_2\dot{N}_1, N_2\dot{N}_2, {\dot{N}_1}^2, \dot{N}_1\dot{N}_2, {\dot{N}_2}^2\}$.

V tretjem koraku preizkušamo linearno regresijo nad vsemi podmnožicami množice V, z največ tremi ($r = 3$) elementi (Tabela 4.3).

$\lvert R\rvert = 1$	$\lvert R\rvert = 2$	$\lvert R\rvert = 3$
$\{N_1\}$	$\{N_1,\ N_2\}$	$\{N_1,\ N_2,\ \dot{N}_1\}$
$\{N_2\}$	$\{N_1,\ \dot{N}_1\}$	$\{N_1,\ N_2,\ \dot{N}_2\}$
...	...	...
$\{{\dot{N}_2}^2\}$	...	$\{N_1,\ \dot{N}_1,\ N_1N_2\}$
	$\{\dot{N}_1\dot{N}_2, {\dot{N}_2}^2\}$	...
		$\{N_2,\ \dot{N}_2,\ N_1N_2\}$
		...
		$\{{\dot{N}_1}^2,\ \dot{N}_1\dot{N}_2,\ {\dot{N}_2}^2\}$

Tabela 4.3: Podmožice za linearno regresijo

Z linearno regresijo nad množicama $\{N_1, \dot{N}_1, N_1N_2\}$ in $\{N_2, \dot{N}_2, N_1N_2\}$ dobimo diferencialni enačbi, ki opišeta obnašanje dinamičnega sistema sožitja populacij volkov in zajcev:

$$\begin{aligned} N_1N_2 &= 160N_1 - 100\dot{N}_1 \\ N_1N_2 &= 200N_2 - 100\dot{N}_2. \end{aligned}$$

4.3 Analiza časovne kompleksnosti

Analizirali bomo časovno kompleksnost algoritma LAGRANGE v odvisnosti od naslednjih parametrov:

- števila spremenljivk dinamičnega sistema (n)
- števila meritev spremenljivk dinamičnega sistema (N)
- maksimalne stopnje časovnih odvodov (o)
- maksimalne globine novih spremenljivk (d)
- maksimalnega števila spremenljivk v enačbah (r)

V prvih dveh korakih algoritma ocenjujemo število spremenljivk $|V|$, v tretjem pa število izvedenih linearnih regresij. Izračun koeficientov linearne regresije ima časovno kompleksnost $\mathcal{O}((m+1)^3 + N(m+1)^2)$, kjer je m število neodvisnih spremenljivk. V algoritmu LAGRANGE velja $N \gg m$, tako da smo privzeli, da je časovna kompleksnost linearne regresije $\mathcal{O}(N(m+1)^2)$.

Po prvem koraku algoritma (Tabela 4.2) je množica $V = V_S \cup D_1 \cup D_2 \cup \ldots \cup D_o$. Vsaka izmed množic $S, D_1, D_2, \ldots, D_o$ vsebuje n elementov. Ker so te množice paroma disjunktne, velja $|V| = (o+1)n$.

V primeru ko je merska enota vseh spremenljivk dinamičnega sistema radian, uvedemo še $2n$ novih spremenljivk oblike $sinX$ in $cosX$. Torej je pred uvajanjem novih spremenljivk z množenjem $|V| = \mathcal{O}((o+3)n)$.

Med uvajanjem novih spremenljivk z množenjem velja odvisnost $|V_l| \leq |V_1||V_{l-1}|$, za $2 \leq l \leq d$. Če skupaj s to odvisnostjo upoštevamo $|V_1| = \mathcal{O}((o+3)n)$, dobimo $|V_l| = \mathcal{O}(((o+3)n)^l)$. Ker je na koncu drugega koraka algoritma $V = V_1 \cup V_2 \cup \ldots \cup V_d$, in so množice $V_1, V_2, \ldots, V_d$ paroma disjunktne, dobimo oceno za celotno število spremenljivk $|V| = \mathcal{O}(((o+3)n)^d)$.

V i-tem prehodu skozi zanko iz tretjega koraka algoritma, preizkušamo linearno regresijo nad vsemi podmnožicami množice V, ki vsebujejo i spremenljivk. Teh je $\binom{|V|}{i} = \mathcal{O}(|V|^i)$. Število neodvisnih spremenljivk pri posamezni linearni regresiji je $i-1$, torej je časovna kompleksnost i-tega prehoda skozi zanko $\mathcal{O}(Ni^2|V|^i)$. Od tod dobimo kompleksnost tretjega koraka: $\mathcal{O}(Nr^2|V|^r)$. Če v tem obrazcu upoštevamo oceno za celotno število spremenljivk, dobimo naslednjo zgornjo mejo časovne kompleksnosti tretjega koraka algoritma LAGRANGE:

$$\mathcal{O}(Nr^2((o+3)n)^{dr}). \tag{4.3}$$

Čas, ki ga porabi LAGRANGE v prvih dveh korakih, je zanemarljiv v primerjavi s časom, ki ga porabi v tretjem. Zaradi tega je obrazec 4.3 tudi zgornja meja za časovno kompleksnost algoritma LAGRANGE. Če v tem obrazcu vstavimo privzete vrednosti parametrov ($o = 1$, $d = 2$, $r = 3$), dobimo zgornjo mejo časovne kompleksnosti $\mathcal{O}(9N(4n)^6) = \mathcal{O}(Nn^6)$.

4.4 Implementacija

LAGRANGE je implementiran v programskem jeziku C pod operacijskim sistemom UNIX in teče na delovni postaji tipa Sun SPARC IPC. Program poženemo z ukazom:

`lg [-R` t_R`] [-S` t_S`] [-o` o`] [-d` d`] [-r` r`]` *ime_datoteke*

S stikali določamo vrednost uporabniško določenih parametrov programa. Če stikalo ni podano, ustreznemu parameteru priredimo privzeto vrednost. V primeru stikal `-R` in `-S` veljata naslednji dve pravili:

- če podamo obe stikali, uporablja LAGRANGE oba kriterija za ocenjevanje signifikantnosti enačb, razen pri enačbah oblike $y = c_0$, pri katerih uporablja samo kriterij S. To pomeni, da je enačba signifikantna če velja $(R > 1 - t_R) \wedge (S < t_S)$.

- če podamo samo eno stikalo, se uporablja samo kriterij, ki ustreza podanemu stikalu. Izjema so ponovno enačbe oblike $y = c_0$.

Parameter *ime_datoteke* je obvezen in z njim podamo ime vhodne datoteke z opisom domene. Vhodna datoteka je sestavljena iz naslednjih treh delov:

1. Časovni korak med dvema zaporednima meritvima spremenljivk dinamičnega sistema (h).

2. Imena spremenljivk dinamičnega sistema in njihove merske enote. Imena se morajo začenjati s črko, kateri lahko sledi poljubna kombinacija črk, cifer in podčrtaja. Če v vhodni datoteki podamo meritve odvoda spremenljivke X dinamičnega sistema, potem odvod poimenujemo X'. LAGRANGE potem ne računa odvoda $\dot{X}$ numerično, pač pa vzame $\dot{X} = X'$.

3. Meritve spremenljivk (in odvodov spremenljivk) dinamičnega sistema v ekvidistančnih časovnih točkah (Tabela 4.1).

Formalni opis vhodne datoteke je podan v dodatku A.

Program LAGRANGE izpiše enačbe modela skupaj z vrednostmi kriterijev S in R. Na koncu izpisa so podani število signifikantnih enačb v modelu ter porabljeni resursi (čas in pomnilnik). Izpis je razumljiv programskemu paketu Mathematica, kar omogoča morebitno reševanje in analizo generiranih enačb.

5.

Poskusi s programom Lagrange

Program Lagrange sem preizkusil na problemih odkrivanja zakonitosti v domenah iz dinamike tekočin, kemičnih reakcij, populacijske rasti in mehanske dinamike.

Za vsako posamezno domeno je bil potek poskusa naslednji: najprej sem izbral množico diferencialnih enačb, ki opisuje dinamični sistem, nabor začetnih vrednosti njegovih spremenljivk, ter vrednosti parametrov, ki nastopajo v diferencialnih enačbah. Tako izbrane diferencialne enačbe sem z metodo Runge-Kutta četrte stopnje [Press et al. 1986] (str. 550–554) integriral v $N = 1000$ časovnih točkah s korakom $h = 0.01$. Dobljeni časovni potek vrednosti spremenljivk dinamičnega sistema sem podal na vhod programa Lagrange.

Domena	n	N	o	d	r	t_R	t_S	št. enačb	čas$[s]$
`Utube`	2	1000	1	1	3	0.01	0.01	6	1.82
`Chem`	3	1000	1	1	3	0.01	0.01	14	2.89
`Brus`	2	1000	1	3	3	0.01	0.01	4	102.91
`Pop`	2	1000	1	2	3	0.01	0.01	2	8.96
`Monod`	2	1000	1	2	3	0.01	0.01	4	9.01
`Spring`	1	1000	2	2	3	0.01	0.01	5	2.87
`Cart`	2	133	2	3	3	$--$	0.0001	5	1494.91

Tabela 5.1: Parametri poskusov

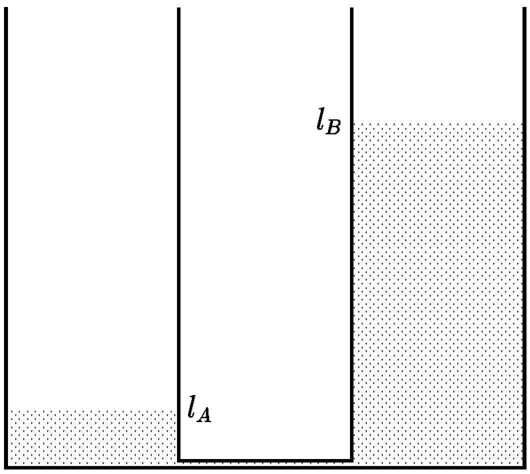
Vrednost parametrov o in d sem izbiral v odvisnosti od diferencialnih enačb dinamičnega sistema. Parameter r je imel pri vseh poskusih privzeto vrednost 3. Parameter o je imel privzeto vrednost 1 za vse domene, razen za domene iz mehanske dinamike, kjer v njihovih diferencialnih enačbah nastopajo drugi odvodi spremenljivk. Parameter d je imel privzeto vrednost 2 za vse domene, razen za domeni Bruselator (`Brus`) in palica na vozičku (`Cart`), kjer v diferencialnih enačbah nastopajo spremenljivke globine 3, ter v linearnih domenah `Utube` in `Chem`, kjer je bila vrednost 1. V domeni palice na vozičku sem uporabil za ocenjevanje signifikantnosti enačb samo kriterij S ($t_S = 10^{-4}$), za vse ostale domene pa kombinacijo obeh kriterijev.

Vhodne in izhodne datoteke programa LAGRANGE za vse poskusne domene sem podal v dodatku B.

5.1 Dinamika tekočin

Sistem povezanih posod (`Utube`) [Kuipers 1986] je sestavljen iz dveh posod A in B, ki sta povezani z ozko cevjo in napolnjeni s tekočino do ravni l_A oz. l_B (Slika 5.1). Razlika v ravneh povzroči pretakanje tekočine skozi cev iz posode B v posodo A. Spremembo ravni tekočin, ki nastane zaradi pretakanja, opišemo z naslednjim sistemom linearnih diferencialnih enačb:

$$\dot{l_A} = c(l_B - l_A) \quad (5.1)$$
$$\dot{l_B} = -\dot{l_A}. \quad (5.2)$$



Slika 5.1: Sistem povezanih posod

Iz podatkov, ki sem jih dobil z integriranjem enačb za $c = 2$ z začetnimi vrednostmi $l_A(0) = 10$ in $l_B(0) = 210$, je LAGRANGE odkril naslednje enačbe:

$$
\begin{array}{lcl}
(1)\ l_B & = & 210 - l_A \\
(2)\ \dot{l_A} & = & 420 - 4l_A \\
(3)\ \dot{l_A} & = & -420 + 4l_B \\
(4)\ \dot{l_B} & = & -420 + 4l_A \\
(5)\ \dot{l_B} & = & 420 - 4l_B \\
(6)\ \dot{l_B} & = & -\dot{l_A}.
\end{array}
$$

Prva enačba izraža zakon o ohranitvi volumna tekočine, zadnja pa je ekvivalentna enačbi 5.2. Drugo in tretjo enačbo dobimo tako, da upoštevamo zakon o ohranitvi volumna ter ustrezno vrednost parametra c v enačbi 5.1. Enačbi (4) in (5), ki jih je LAGRANGE odkril, lahko dobimo z naslednjimi linearnimi kombinacijami:

$$
\begin{array}{lcl}
(4) & = & (2) - (6) \\
(5) & = & (3) - (6)
\end{array}
$$

5.2 Kemične reakcije

5.2.1 Enostavna kemična reakcija

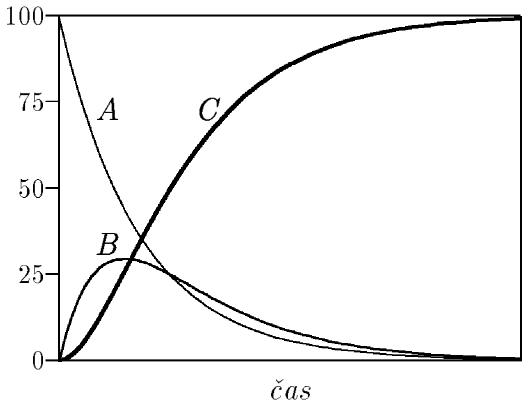
Na začetku enostavne kemične reakcije imamo neko koncentracijo A_0 nestabilne snovi A. Snov A s hitrostjo k_1 prehaja v snov B, ta pa s hitrostjo k_2 v snov C, ki je stabilna in ne razpada naprej.

Model enostavne kemične reakcije opisuje časovni potek koncentracij snovi A, B in C in se da zapisati kot sistem treh linearnih diferencialnih enačb [Babloyantz 1986] (str. 39):

$$
\begin{array}{lcll}
\dot{A} & = & -k_1 A & (5.3) \\
\dot{B} & = & k_1 A - k_2 B & (5.4) \\
\dot{C} & = & k_2 B. & (5.5)
\end{array}
$$

Začetni pogoj je $A(0) = A_0$, $B(0) = C(0) = 0$. Pri integriranju diferencialnih enačb sem za začetno koncentracijo vzel $A_0 = 100$, ter hitrosti reakcij $k_1 = 2$ in $k_2 = 3$. Na osnovi dobljenega časovnega poteka spremenljivk A, B in C (Slika 5.2), je LAGRANGE odkril štirinajst enačb:

$$
\begin{array}{lcl}
(1)\ \dot{A} & = & -2A \\
(2)\ \dot{B} & = & 2A - 3B
\end{array}
$$



Slika 5.2: Časovni potek koncentracij snovi A, B in C

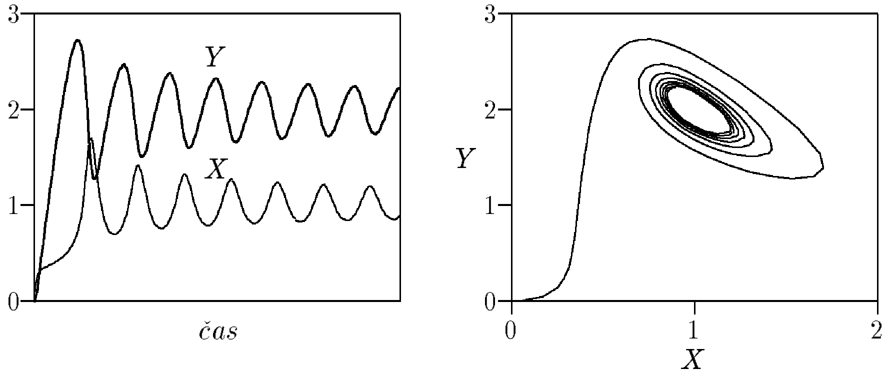
$$\begin{aligned}
(3)\ \dot{C} &= 3B \\
(4)\ C &= 100 - A - B \\
(5)\ \dot{C} &= -\dot{A} - \dot{B} \\
(6)\ \dot{A} &= -200 + 2B + 2C \\
(7)\ \dot{B} &= -300 + 5A + 3C \\
(8)\ \dot{B} &= 200 - 5B - 2C \\
(9)\ \dot{B} &= -3B - \dot{A} \\
(10)\ \dot{B} &= -300 + 3C - 2.5\dot{A} \\
(11)\ \dot{C} &= 300 - 3A - 3C \\
(12)\ \dot{C} &= 2A - \dot{B} \\
(13)\ \dot{C} &= 300 - 3C + 1.5\dot{A} \\
(14)\ \dot{C} &= 120 - 1.2C - 0.6\dot{B}.
\end{aligned}$$

Prve tri enačbe so ekvivalentne enačbam modela (5.3–5.5) za ustrezne vrednosti parametrov. Četrta enačba izraža zakon o ohranitvi mase ($A + B + C = const$), peto pa dobimo z odvajanjem obeh strani četrte enačbe. Ostale enačbe, ki jih je odkril LAGRANGE, so redundantne, ker jih lahko dobimo z linearnimi kombinacijami prvih petih enačb:

$$\begin{aligned}
(6) &= (1) - 2(4) \\
(7) &= (2) - 3(4) \\
(8) &= (2) - 2(4) \\
(9) &= (3) - (5)
\end{aligned}$$

$$\begin{aligned}(10) &= (2) - 3(4) + 2.5(1)\\ (11) &= (3) + 3(4)\\ (12) &= (1) - (5)\\ (13) &= (3) + 3(4) - 1.5(1)\\ (14) &= 0.6(2) + (3) + 1.2(4).\end{aligned}$$

5.2.2 Bruselator



Slika 5.3: Nihanje koncentracij snovi X in Y

Bruselator [Babloyantz 1986] (str. 175–177) je kemični reaktor, v katerem koncentraciji dveh snovi X in Y nihata okrog nekega ravnovesnega stanja (Slika 5.3). Spreminjanje koncentracij snovi lahko opišemo z naslednjima nelinearnima enačbama:

$$\begin{aligned}\dot{X} &= A - (B+1)X + X^2Y\\ \dot{Y} &= BX - X^2Y.\end{aligned}$$

Pri integriranju diferencialnih enačb sem vzel začetni koncentraciji $X(0) = Y(0) = 0$ ter parametra $A = 1$ in $B = 2$. Iz dobljenega obnašanja je LAGRANGE odkril naslednje enačbe:

$$\begin{aligned}(1)\quad \dot{Y} &= 1 - X - \dot{X}\\ (2)\quad X^2Y &= -1 + 3X + \dot{X}\\ (3)\quad X^2Y &= 2X - \dot{Y}\\ (4)\quad X^2Y &= 2 - 2\dot{X} - 3\dot{Y}.\end{aligned}$$

Druga in tretja enačba sta ekvivalentni enačbam modela (5.6 in 5.6), ostali dve dobimo z linearno kombinacijo druge in tretje enačbe:

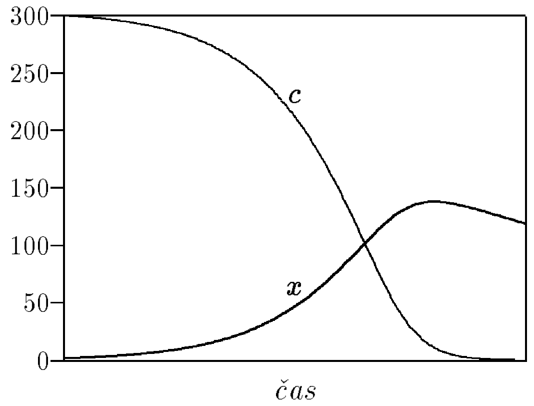
$$
\begin{aligned}
(1) &= (3) - (2) \\
(4) &= 3(3) - 2(2).
\end{aligned}
$$

5.3 Populacijska rast

5.3.1 Monod

V posodi imamo neko koncentracijo hranilne raztopine c in koncentracijo bio-mase x. Posoda je izolirana od okolja, tako da ni dotoka hrane in bio-mase od zunaj ter ima konstantno temperaturo. Velika začetna koncentracija hranilne raztopine povzroči razvoj bio-mase. Koncentracija bio-mase narašča, dokler ne zmanjka hrane, potem pa začne padati (Slika 5.4). Spreminjanje koncentracij bio-mase in hranilne raztopine opišemo z naslednjima diferencialnima enačbama [Jorgensen and Johnsen 1989] (str. 301):

$$
\begin{aligned}
\dot{c} &= -\frac{\mu_{max}}{y} \frac{c}{c + k_s} x && (5.6) \\
\dot{x} &= (\mu_{max} \frac{c}{c + k_s} - k_d)\, x. && (5.7)
\end{aligned}
$$



Slika 5.4: Časovni potek koncentracij bio-mase in hranilne raztopine

Zgornje enačbe sem integriral pri vrednostih parametrov $\mu_{max} = 0.1$, $k_d = 0.01$, $k_s = 100$ in $y = 0.6$, z začetnima koncentracijama $c(0) = 300$ in $x(0) = 2$ ter časovnim korakom $h = 0.1$. Časovni korak je večji kot v ostalih poskusih zaradi počasne dinamike spreminjanja koncentracij bio-mase in hranilne raztopine. LAGRANGE je odkril naslednje štiri enačbe:

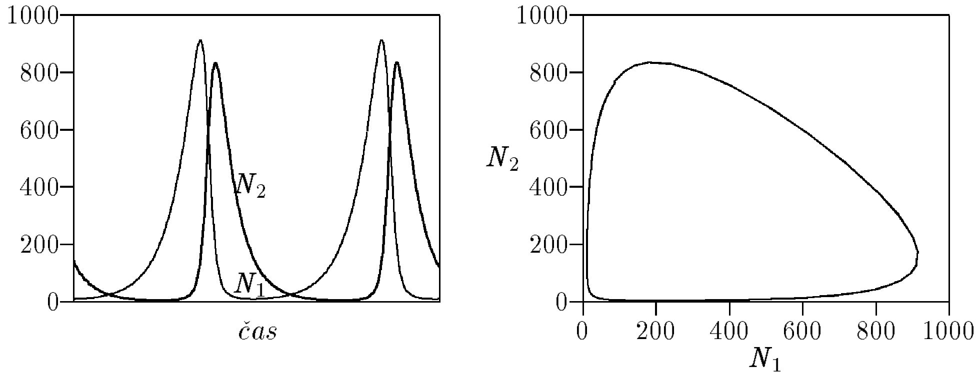
$$\begin{array}{lrcl} (1) & \dot{x} & = & -0.01x - 0.6\dot{c} \\ (2) & c\dot{c} & = & -100\dot{c} - 0.167cx \\ (3) & c\dot{x} & = & 60\dot{c} + 0.09cx \\ (4) & c\dot{x} & = & 6\dot{c} - 0.54c\dot{c}. \end{array}$$

Enačba (2) je ekvivalentna enačbi 5.6 za ustrezne vrednosti parametrov. Upoštevaje enačbi (1) in (2) dobimo enačbo, ki je ekvivalentna enačbi 5.7. Torej sta prvi dve enačbi, ki jih je odkril LAGRANGE, ekvivalentni modelu. Preostali dve sta redundantni, ker ju lahko izpeljemo z naslednjimi linearnimi kombinacijami:

$$\begin{array}{rcl} (3) & = & c(1) - 0.6(2) \\ (4) & = & c(1) - 0.06(2). \end{array}$$

5.3.2 Volkovi in zajci

Predstavljajmo si sožitje populacij volkov in zajcev v istem okolju. Predpostavimo še da so zajci edina hrana volkovom, zajci pa imajo neomejene zaloge korenčkov. Če je populacija zajcev velika, populacija volkov narašča, ker imajo veliko hrane. Naraščanje traja toliko časa, dokler ne začne zmanjkovati hrane, torej zajcev. Potem se število volkov začne zmanjševati. To pa takoj izkoristijo zajci, katerih število začne naraščati (Slika 5.5).



Slika 5.5: Nihanje velikosti populacij volkov in zajcev

Nihanje velikosti populacij volkov in zajcev lahko opišemo z modelom Volterra-Lotka [Babloyantz 1986] (str. 145):

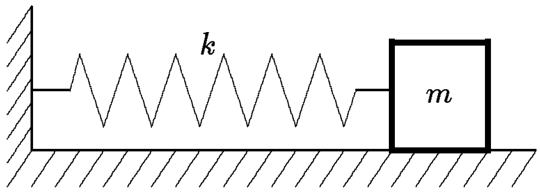
$$\begin{aligned} \dot{N}_1 &= k_1 N_1 - s N_2 N_1 \\ \dot{N}_2 &= s N_1 N_2 - k_2 N_2, \end{aligned}$$

pri čemer sta N_2 in N_1 ustrezni velikosti populacij volkov in zajcev. Pri integriranju enačb sem vzel začetne velikosti $N_1(0) = 10$ in $N_2(0) = 140$, ter parametre $k_1 = 1.6$, $s = 0.01$ in $k_2 = 2$. LAGRANGE je na osnovi časovnega poteka spremenljivk N_1 in N_2 (Slika 5.5) odkril dve enačbi, ki sta ekvivalentni enačbam modela:

$$\begin{aligned} N_1 N_2 &= 160 N_1 - 100 \dot{N}_1 \\ N_1 N_2 &= 200 N_2 - 100 \dot{N}_2. \end{aligned}$$

5.4 Mehanska dinamika

5.4.1 Vzmetno nihalo



Slika 5.6: Vzmetno nihalo

Telo z maso m je pritrjeno na prožno vzmet s prožnostno konstanto k (Slika 5.6). V ravnovesni legi vzmet ni ne raztegnjena ne skrčena. Če telo izmaknemo iz ravnovesne lege za x_0 in nato spustimo (ob tem zanemarimo trenje med telesom in podlago), začne le to nihati z amplitudo x_0. Spreminjanje odmika telesa od ravnovesne lege x lahko opišemo z naslednjo linearno diferencialno enačbo drugega reda [Kladnik 1989] (str. 44):

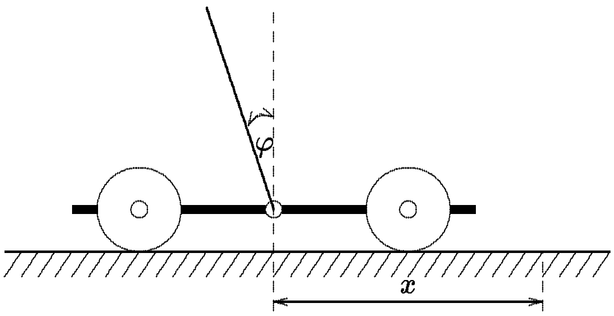
$$\ddot{x} = -\frac{k}{m} x \tag{5.8}$$

Začetna pogoja diferencialne enačbe sta $x(0) = x_0$ in $\dot{x}(0) = 0$. LAGRANGE je iz podatkov, dobljenih z integriranjem diferencialne enačbe 5.8 pri $x(0) = 10$, $k = 2$ in $m = 1$, odkril naslednje diferencialne enačbe:

$$
\begin{aligned}
\ddot{x} &= -2x \\
\ddot{x}^2 &= 4x^2 \\
\ddot{x}^2 &= 400 - 2\dot{x}^2 \\
\dot{x}^2 &= 200 - 2x^2 \\
\dot{x}^2 &= 200 + x\ddot{x}
\end{aligned}
$$

Prva enačba je enaka enačbi 5.8, drugo enačbo dobimo s kvadriranjem obeh strani prve. Ostale tri enačbe lahko izpeljemo, če upoštevamo analitično rešitev enačbe 5.8 $x = x_0 \cos\sqrt{\frac{k}{m}}t$, ki pripelje te enačbe do oblike $\sin^2\sqrt{\frac{k}{m}}t + \cos^2\sqrt{\frac{k}{m}}t = 1$. Zadnjo enačbo lahko interpretiramo kot zakon o ohranitvi energije.

5.4.2 Invertirano nihalo



Slika 5.7: Invertirano nihalo

Vodenje invertiranega nihala je standardni problem za preizkus različnih metod vodenja dinamičnih sistemov. Invertirano nihalo je sestavljeno iz vozička, na katerem je vrtljivo vpeta palica, ki je gibljiva v vertikalni ravnini, ki gre skozi tirnice (Slika 5.7). Če sta M in m masi vozička in palice, l dolžina palice, F sila s katero delujemo na voziček, x pozicija vozička glede na izhodišče in φ kot nagiba palice, potem dinamiko sistema palice na vozičku opišemo z naslednjima diferencialnima enačbama [Džeroski 1989]:

$$
\begin{aligned}
(M+m)\ddot{x} + \frac{1}{2}ml(\ddot{\varphi}\cos\varphi - \dot{\varphi}^2\sin\varphi) &= F \qquad (5.9) \\
\ddot{x}\cos\varphi + \frac{2}{3}l\ddot{\varphi} &= g\sin\varphi. \qquad (5.10)
\end{aligned}
$$

Pri integriranju enačb sem vzel za $M = 1[kg]$, $m = 0.1[kg]$, $l = 1[m]$ ter $F = 7.5[N]$. Integraciski korak je bil $h = 0.01$. Iz začetnega stanja $x(0) = \dot{x}(0) = \dot{\varphi}(0) = 0$ ter $\varphi(0) = 3\pi/16$ je po $N = 133$ časovnih korakih palica padla. Iz dobljenih časovnih potekov spremenljivk x, φ, $\dot{x}$ in $\dot{\varphi}$ je LAGRANGE odkril naslednjih pet enačb:

$$
\begin{aligned}
\cos^2\varphi &= 1 - \sin^2\varphi \\
\ddot{x}\cos\varphi &= -0.67\ddot{\varphi} + 9.81\sin\varphi \\
\dot{\varphi}^2\sin\varphi &= -150 + 22\ddot{x} + \ddot{\varphi}\cos\varphi \\
\ddot{x}\cos\varphi\sin\varphi &= 9.81 - 0.67\ddot{\varphi}\sin\varphi - 9.81\cos^2\varphi \\
\ddot{x}^3 &= 394.34 - 272.73\sin^2\varphi.
\end{aligned}
$$

Prva enačba izraža znano zvezo med sinusom in kosinusom kota ($\sin^2\varphi + \cos^2\varphi = 1$). Druga in tretja enačba sta ekvivalentni enačbama 5.9 in 5.10. Četrto enačbo dobimo, če zmnožimo drugo s $\sin\varphi$ in upoštevamo zvezo $\sin^2\varphi + \cos^2\varphi = 1$. Zadnja enačba, ki jo je LAGRANGE odkril, ni ne redundantna, ne ekvivalentna kateri izmed enačb modela. Od ostalih odkritih enačb jo loči večja vrednost kriterija S.

Za ocenjevanje signifikantnosti enačb pri poskusu z domeno `Cart` sem uporabil samo kriterij S s pragom $t_S = 10^{-4}$. S privzetima vrednostima pragov t_R in t_S je LAGRANGE odkril preveliko število enačb. Vrednost praga t_S sem določil tako, da sem program večkrat zapored pognal s padajočimi vrednostmi praga, dokler nisem dobil obvladljivo število enačb.

6.

Zaključki in nadaljnje delo

V diplomskem delu sem razvil program LAGRANGE, ki avtomatsko odkriva množico diferencialnih in algebraičnih enačb iz izmerjenega obnašanja dinamičnega sistema v nekem časovnem intervalu.

Bistveni prispevek programa LAGRANGE je razširitev področja uporabe programov za avtomatsko odkrivanje zakonitosti v smeri modeliranja dinamičnih sistemov. Metode, uporabljene v drugih programih sem prilagodil posebnostim, ki se pojavljajo pri odkrivanju zakonitosti v dinamični domeni:

- ni eksperimentalne kontrole nad posameznimi spremenljivkami
- ni eksplicitnega razlikovanja med odvisnimi in neodvisnimi spremenljivkami

LAGRANGE sem preizkusil na problemih rekonstrukcije modelov dinamičnih sistemov iz dinamike tekočin, kemičnih reakcij, populacijske ter mehanske dinamike. Med njimi so tudi nelinearni modeli oscilatornih procesov ter dinamični model inverznega nihala, ki je standardni problem za testiranje različnih metod vodenja dinamičnih sistemov. V vseh domenah je LAGRANGE uspešno rekonstruiral enačbe modela, poleg njih pa še enačbe, ki jih lahko razvrstimo v štiri skupine:

- zakon o ohranjanju volumna oz. mase (`Utube` in `Chem`) ter energije (`Spring`)
- trigonometrijska identiteta $\sin^2\varphi + \cos^2\varphi = 1$ (`Cart`)
- enačbe, ki izvirajo iz analitične rešitve (`Spring`)
- redundantne enačbe, ki se lahko izpeljejo iz enačb modela (vse domene razen `Pop`)

Prioritetna naloga nadaljnjega dela bi bila vgraditev mehanizmov, podobnih tistim iz modula EQUATION FINDER, za eksplicitno obravnavanje meritvene napake v vhodnih podatkih. To bi omogočilo uporabo programa LAGRANGE za odkrivanje zakonitosti v realnih dinamičnih domenah. Pri domenah z nenatančnimi podatki lahko ustrezno povečamo vrednosti pragov t_S in t_R ter dobimo tudi enačbe, ki so manj signifikantne. Za spreminjanje vrednosti pragov je potrebna podrobna analiza vpliva teh vrednosti na rezultate, ki jih da program.

Druga naloga bi bila nadaljnji razvoj programa LAGRANGE v interaktivno orodje za modeliranje dinamičnih sistemov, ki bi zajemalo:

- vizualizacijo vhodnih podatkov v časovnem, frekvenčnem, faznem prostoru ter prostoru stanj
- predprocesiranje vhodnih podatkov s ciljem odstranjevanja šuma in meritvenih napak
- primerjavo vhodnih podatkov s podatki, dobljenimi s simuliranjem enačb, ki jih je LAGRANGE odkril.

Nadaljnje delo bi lahko zajelo tudi izločanje redundantnih enačb, izbiranje spremenljivk za linearno regresijo na bolj inteligenten način ter odkrivanje parcialnih diferencialnih enačb. Za zahtevnejšo praktično uporabo na določenih področjih, kot je na primer ekološko modeliranje, bi naj bile generirane enačbe prilagojene ustaljenemu slogu pisanja enačb na konkretnem področju uporabe.

7.

Zahvala

Končne besede bi rad namenil za iskreno zahvalo vsem, ki so mi bili v pomoč pri izdelavi diplomske naloge.

Najprej se želim zahvaliti svojemu mentorju prof. dr. Ivanu Bratku, za strokovno pomoč in napotke pri delu.

Za vzpodbudo ter številne nasvete vsebinske in metodološke narave se najtopleje zahvaljujem mag. Sašu Džeroskemu.

Hvala mag. Borisu Kompareu, ki mi je svetoval testno domeno populacijske rasti (`Monod`).

Kolegicam in kolegu Martini Grlj, Borislavi Injac, Sandi Martinčić in Vitu Muranoviču gre posebna zahvala za skrbno prebiranje teksta in številne pripombe.

Literatura

[Babloyantz 1986]
Babloyantz, A. (1986). *Molecules, Dynamics, and Life.* John Wiley & Sons, Inc., New York.

[Bielecki 1992]
Bielecki, M. W. (1992). Machine discovery approach to dynamic systems in the real laboratory. In Żytkow, J. M., editor, *Proc. of the ML92 Workshop on Machine Discovery*, pages 58–62.

[Bohte 1991]
Bohte, Z. (1991). *Numerične metode.* Društvo matematikov, fizikov in astronomov Slovenije, Ljubljana.

[Bratko 1990]
Bratko, I. (1990). *Prolog Programming for Artificial Intelligence.* Addison-Wesley, Wokingham, 2nd edition.

[Carbonell and Langley 1987]
Carbonell, J. and Langley, P. (1987). Learning, machine. In Shapiro, S. C. and Eckroth, D., editors, *Encyclopedia of Artifical Intelligence*, volume 1, pages 464–488. John Wiley & Sons, Inc., New York.

[Džeroski 1989]
Džeroski, S. (1989). Upravljanje inverznega nihala. Diplomska naloga, Fakulteta za elektrotehniko in računalništvo, Univerza v Ljubljani.

[Falkenheiner and Michalski 1990]
Falkenheiner, B. and Michalski, R. (1990). Integrating quantitative and qualitative discovery in the ABACUS system. In Kodratoff, Y. and Michalski, R., editors, *Machine Learning: An Artificial Intelligence Approach*, pages 153–190. Morgan Kaufmann, San Mateo, CA.

[Jorgensen and Johnsen 1989]
Jorgensen, S. and Johnsen, E. (1989). *Principles of Environmental Science and Technology.* Elsevier, Amsterdam.

[Karba 1993]
Karba, R. (1993). Matematično modeliranje dinamičnih sistemov. In *Active Learning in Automatic Control, II. Modeliranje in simulacija sistemov vodenja*, pages 3–34. Fakulteta za elektrotehniko in računalništvo, Univerza v Ljublani. TEMPUS projekt.

[Kladnik 1989]
Kladnik, R. (1989). *Visokošolska fizika.* Državna založba Slovenije. Del 1: Mehanski in toplotni pojavi.

[Kuipers 1986]
Kuipers, B. (1986). Qualitative simulation. *Artificial Intelligence*, 29(3):289–338.

[Langley et al. 1987]
Langley, P., Simon, H., and Bradshaw, G. (1987). Heuristics for empirical discovery. In Bolc, L., editor, *Computational Models of Learning.* Springer, Berlin.

[Langley and Żytkow 1989]
Langley, P. and Żytkow, J. (1989). Data-driven approaches to empirical discovery. *Artificial Intelligence*, 40:283–312.

[Press et al. 1986]
Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. (1986). *Numerical Recipes.* Cambridge University Press, Cambridge, MA.

[Shearer and Kulakowski 1990]
Shearer, J. L. and Kulakowski, B. T. (1990). *Dynamic Modeling and Control of Engineering Systems.* Macmillan Publishing Company, Inc., New York.

[Volk 1958]
Volk, W. (1958). *Applied Statistics for Engineers.* McGraw-Hill, New York.

[Zembowitz and Żytkow 1992]
Zembowitz, R. and Żytkow, J. (1992). Discovery of equations: experimental evaluation of convergence. In *Proc. Tenth National Conference on Artificial Intelligence.* Morgan Kaufmann, San Mateo, CA.

[Żytkow and Zhu 1991]
Żytkow, J. and Zhu, J. (1991). Application of empirical discovery in knowledge acquisition. In *Proc. Fifth European Working Session on Learning*, pages 101–117. Springer, Berlin.

A. Dodatek

Formalni opis vhodnih datotek programa Lagrange

$$
\begin{array}{lcl}
\mathit{InFile} & \longrightarrow & \mathit{TimeStep}\ \mathit{VarDecls}\ \mathit{Measures} \\[1ex]
\mathit{TimeStep} & \longrightarrow & \mathbf{number} \\[1ex]
\mathit{VarDecls} & \longrightarrow & \mathit{VarDecls}\ \mathit{VarDecl} \\
 & | & \mathit{VarDecl} \\[1ex]
\mathit{VarDecl} & \longrightarrow & \mathit{VarId}\ \mathit{VarType}\ \text{“;“} \\[1ex]
\mathit{VarId} & \longrightarrow & \mathbf{identifier} \\
 & | & \mathbf{identifier}\ \text{“'“} \\[1ex]
\mathit{VarType} & \longrightarrow & \text{“:“}\ \mathit{VarDim} \\
 & | & \epsilon \\[1ex]
\mathit{VarDim} & \longrightarrow & \text{“[“}\ \mathit{Product}\ \text{“/“}\ \mathit{Product}\ \text{“]“} \\
 & | & \text{“[“}\ \mathit{Product}\ \text{“]“} \\
 & | & \text{“[“}\ \text{“]“} \\[1ex]
\mathit{Product} & \longrightarrow & \mathit{Product}\ \text{“}*\text{“}\ \mathbf{identifier} \\
 & | & \mathbf{identifier} \\[1ex]
\mathit{Measures} & \longrightarrow & \mathit{Measure}\ \text{“;“}\ \mathit{Measures} \\
 & | & \mathit{Measure}\ \text{“;“} \\[1ex]
\mathit{Measure} & \longrightarrow & \mathit{Measure}\ \mathbf{number} \\
 & | & \mathbf{number}
\end{array}
$$

$$
\begin{array}{lcl}
\textbf{number} & \longrightarrow & \textbf{sign digits} \\
 & | & \textbf{sign digits}\ '.'\ \textbf{digits} \\
 & | & \textbf{sign digits}\ '.'\ \textbf{digits}\ '\mathbf{E}'\ \textbf{sign digits} \\
 & & \\
\textbf{digits} & \longrightarrow & \textbf{digits digit} \\
 & | & \textbf{digit} \\
 & & \\
\textbf{identifier} & \longrightarrow & \textbf{letter string} \\
 & & \\
\textbf{string} & \longrightarrow & \textbf{string alphanum} \\
 & | & \epsilon \\
 & & \\
\textbf{alphanum} & \longrightarrow & \textbf{letter} \mid \textbf{digit} \mid '\_' \\
 & & \\
\textbf{letter} & \longrightarrow & '\mathbf{a}' \mid '\mathbf{A}' \mid '\mathbf{b}' \mid '\mathbf{B}' \mid \ldots \mid '\mathbf{z}' \mid '\mathbf{Z}' \\
 & & \\
\textbf{digit} & \longrightarrow & '\mathbf{0}' \mid '\mathbf{1}' \mid \ldots \mid '\mathbf{9}' \\
 & & \\
\textbf{sign} & \longrightarrow & '+' \mid '-' \mid \epsilon
\end{array}
$$

B. Dodatek

Vhodne in izhodne datoteke za poskusne domene

B.1 Utube

1. Vhodna datoteka:

```
0.01

lA;
lB;


10.0 200.0;
13.7250032 196.2749968;
17.30394693852621 192.6960530614738;
...
104.9999999999998 105.0000000000002;
104.9999999999998 105.0000000000002;
```

2. Izhodna datoteka:

```
(*
Order of the system (o):        1
Maximal depth of new terms (d): 1
Maximal num. of regr. vars (r): 2
Tolerance (R):                  0.01
Tolerance (S):                  0.01
*)

eqn1 = lB == 210.00000 - 1.00000*lA
(* R = 1, sigma = 4.52358e-08 *)
```

```
eqn2 = lA' == 419.99995 - 4.00000*lA
(* R = 1, sigma = 2.14127e-06 *)

eqn3 = lB' == -419.99995 + 4.00000*lA
(* R = 1, sigma = 2.17125e-06 *)

eqn4 = lA' == -419.99995 + 4.00000*lB
(* R = 1, sigma = 1.71508e-06 *)

eqn5 = lB' == 419.99995 - 4.00000*lB
(* R = 1, sigma = 1.7187e-06 *)

eqn6 = lB' == -1.00000*lA'
(* R = 1, sigma = 7.04837e-08 *)

neqns = 6
(*
Time elapsed [s]: 1.820
Memory used  [K]: 420
*)
```

B.2 Chem

1. Vhodna datoteka:

```
0.01

a;
b;
c;


100.0 0 0;
98.0198673333333 1.950627916666667 0.02950475;
96.0789439204427 3.804981045820865 0.1160750337364645;
...
0.0 0.0 99.9999993816724;
```

2. Izhodna datoteka:

```
(*
Order of the system (o):        1
Maximal depth of new terms (d): 1
Maximal num. of regr. vars (r): 2
Tolerance (R):                  0.01
Tolerance (S):                  0.01
*)

eqn1 = a' == -2.00000*a
(* R = 1, sigma = 0.000447577 *)

eqn2 = c' == 3.00000*b
(* R = 1, sigma = 0.000544663 *)

eqn3 = c == 100.00000 - 1.00000*a - 1.00000*b
(* R = 1, sigma = 3.47684e-08 *)

eqn4 = b' == 2.00000*a - 3.00000*b
(* R = 1, sigma = 0.00324875 *)

eqn5 = b' == -300.00000 + 5.00000*a + 3.00000*c
(* R = 1, sigma = 3.42638e-05 *)

eqn6 = c' == 299.99999 - 3.00000*a - 3.00000*c
(* R = 1, sigma = 2.09675e-06 *)

eqn7 = c' == 2.00000*a - 1.00000*b'
(* R = 1, sigma = 0.000435006 *)
```

```
eqn8 = a' == -200.00000 + 2.00000*b + 2.00000*c
(* R = 1, sigma = 2.94267e-06 *)

eqn9 = b' == 199.99998 - 5.00000*b - 2.00000*c
(* R = 1, sigma = 3.04881e-05 *)

eqn10 = b' == -3.00000*b - 1.00000*a'
(* R = 1, sigma = 0.00553216 *)

eqn11 = b' == -299.99999 + 3.00000*c - 2.50000*a'
(* R = 1, sigma = 7.31779e-05 *)

eqn12 = c' == 299.99999 - 3.00000*c + 1.50000*a'
(* R = 1, sigma = 4.38116e-06 *)

eqn13 = c' == 120.00000 - 1.20000*c - 0.60000*b'
(* R = 1, sigma = 1.74301e-06 *)

eqn14 = c' == -1.00000*a' - 1.00000*b'
(* R = 1, sigma = 0.000110804 *)

neqns = 14
(*
Time elapsed [s]: 2.890
Memory used  [K]: 452
*)
```

B.3 Brus

1. Vhodna datoteka:

```
0.01

x;
y;


0 0;
0.0098514887699171 0.0000990074800626859;
0.01941182260224264 0.0003921180652128841;
...
0.7737243691800048 1.898851588795946;
0.7718929359083285 1.902954956909524;
```

2. Izhodna datoteka:

```
(*
Order of the system (o):        1
Maximal depth of new terms (d): 3
Maximal num. of regr. vars (r): 2
Tolerance (R):                  0.01
Tolerance (S):                  0.01
*)

eqn1 = y' == 1.00000 - 1.00000*x - 1.00000*x'
(* R = 1, sigma = 5.3879e-08 *)

eqn2 = x*x*y == -1.00000 + 3.00000*x + 1.00000*x'
(* R = 1, sigma = 8.37168e-08 *)

eqn3 = x*x*y == 2.00000*x - 1.00000*y'
(* R = 1, sigma = 3.93099e-05 *)

eqn4 = x*x*y == 2.00000 - 2.00000*x' - 3.00000*y'
(* R = 1, sigma = 4.47485e-08 *)

neqns = 4
(*
Time elapsed [s]: 102.910
Memory used  [K]: 812
*)
```

B.4 Pop

1. Vhodna datoteka:

```
0.01

n1;
n2;


10.0 140.0;
10.02134431545907 137.3652542437609;
10.04535540691442 134.7803990685424;
...
10.22324706078136 121.4492712266414;
10.26390971704884 119.1664186080841;
```

2. Izhodna datoteka:

```
(*
Order of the system (o):        1
Maximal depth of new terms (d): 2
Maximal num. of regr. vars (r): 2
Tolerance (R):                  0.01
Tolerance (S):                  0.01
*)

eqn1 = n1*n2 == 160.00001*n1 - 100.00007*n1'
(* R = 1, sigma = 9.6079e-05 *)

eqn2 = n1*n2 == 200.00001*n2 + 100.00008*n2'
(* R = 1, sigma = 0.000101256 *)

neqns = 2
(*
Time elapsed [s]: 8.960
Memory used  [K]: 540
*)
```

B.5 Monod

1. Vhodna datoteka:

```
0.1

c;
x;


300.0 2.0;
299.9749188364525 2.013042184074707;
299.9496746477404 2.026169098744204;
...
0.1658897331790164 118.9091328294572;
0.1626413301530975 118.7922311956354;
```

2. Izhodna datoteka:

```
(*
Order of the system (o):        1
Maximal depth of new terms (d): 2
Maximal num. of regr. vars (r): 2
Tolerance (R):                  0.01
Tolerance (S):                  0.01
*)

eqn1 = x' == -0.01000*x - 0.60000*c'
(* R = 1, sigma = 2.54692e-06 *)

eqn2 = c*c' == -100.00000*c' - 0.16667*c*x
(* R = 1, sigma = 5.5155e-06 *)

eqn3 = c*x' == 60.00000*c' + 0.09000*c*x
(* R = 1, sigma = 4.87091e-06 *)

eqn4 = c*x' == 6.00000*c' - 0.54000*c*c'
(* R = 1, sigma = 3.02063e-06 *)

neqns = 4
(*
Time elapsed [s]: 9.010
Memory used  [K]: 536
*)
```

B.6 Spring

1. Vhodna datoteka:

```
0.01

x;


10.0;
9.99900001666667;
9.99600026666111;
...
0.0917332909457463;
-0.04968657418666778;
```

2. Izhodna datoteka:

```
(*
Order of the system (o):        2
Maximal depth of new terms (d): 2
Maximal num. of regr. vars (r): 2
Tolerance (R):                  0.01
Tolerance (S):                  0.01
*)

eqn1 = x'' == -2.00000*x
(* R = 1, sigma = 0.000214594 *)

eqn2 = x'*x' == 200.00000 - 2.00000*x*x
(* R = 1, sigma = 7.59995e-08 *)

eqn3 = x''*x'' == 4.00000*x*x
(* R = 1, sigma = 2.59519e-06 *)

eqn4 = x'*x' == 200.00000 + 1.00000*x*x''
(* R = 1, sigma = 8.63296e-08 *)

eqn5 = x''*x'' == 400.00000 - 2.00000*x'*x'
(* R = 1, sigma = 1.00376e-07 *)

neqns = 5
(*
Time elapsed [s]: 2.870
Memory used  [K]: 472
*)
```

B.7 Cart

1. Vhodna datoteka:

```
0.01

x;
x';
fi:[rad];
fi';


0 0
0.5890486225480862 0;

0.0003415605750759988 0.06831213622905757
0.5890313849137417 -0.003448068304054987;

0.001366243573485249 0.1366245271530282
0.588979639516068 -0.006902636906070287;

...

5.310309162121512 8.55017575269466
-1.257203392436399 -6.820144004618409;

5.396059516024213 8.59957645724162
-1.326216542754854 -6.982001444772862;
```

2. Izhodna datoteka:

```
(*
Order of the system (o):        2
Maximal depth of new terms (d): 3
Maximal num. of regr. vars (r): 2
Tolerance (S):                  0.0001
*)

eqn1 = Cos[fi]*Cos[fi] == 1.00000 - Sin[fi]*Sin[fi]
(* R = 1, sigma = 2.5109e-09 *)

eqn2 = Cos[fi]*x'' == -0.66667*fi'' + 9.81000*Sin[fi]
(* R = 0.999997, sigma = 0.00042398 *)

eqn3 = Sin[fi]*fi'*fi' == -149.99856 + 21.99979*x''
        + 0.99999*Cos[fi]*fi''
(* R = 1, sigma = 3.25714e-05 *)
```

```
eqn4 = Cos[fi]*Sin[fi]*x'' == 9.81001 - 0.66667*Sin[fi]*fi''
        - 9.81002*Cos[fi]*Cos[fi]
(* R = 1, sigma = 5.97045e-06 *)

eqn5 = x''*x''*x'' == 394.34014 - 272.73247*Sin[fi]*Sin[fi]
(* R = 0.984799, sigma = 0.0267243 *)

neqns = 5
(*
Time elapsed [s]: 1494.910
Memory used  [K]: 672
*)
```

Izjava

Izjavljam, da sem diplomsko delo samostojno izdelal pod vodstvom mentorja prof. dr. Ivana Bratka, dipl. ing. Izkazano pomoč drugih sodelavcev sem v celoti navedel v zahvali.

Ljupčo Todorovski