

Trijų lygių lygiagretinimo įrankis - teorija, realizacija, galimybės, pavyzdžiai

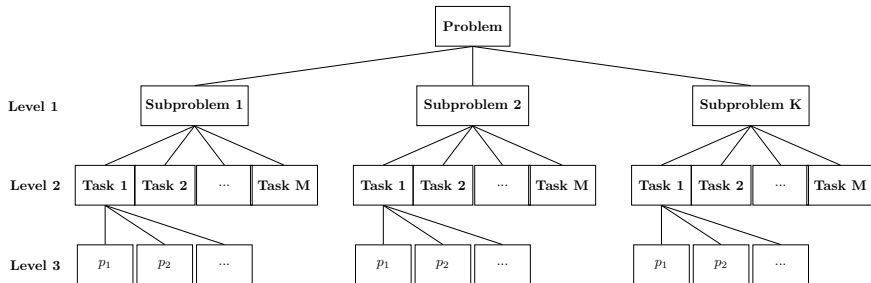
Rima Kriauzienė^{1,2}, Andrej Bugajev¹, Raimondas Čiegis¹

¹Vilnius Gediminas Technical University

²Institute of Data Science and Digital Technologies, Vilnius University

Liepos 1 d., 2019 m., Vilnius, Lietuva

Trisluoksnė lygiagretinimo schema

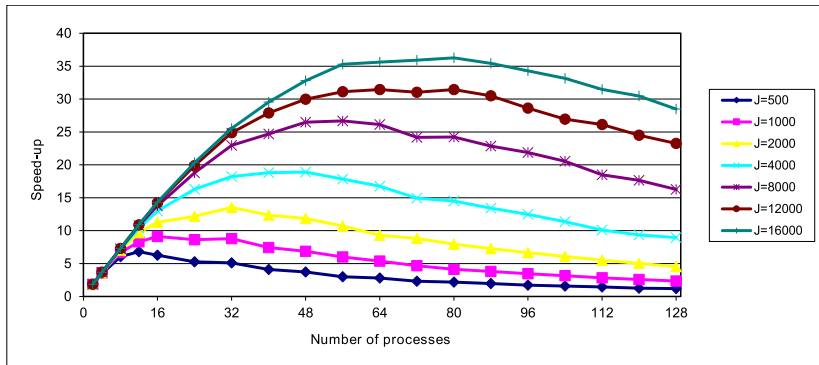


Pavyzdys, turime M diferencialinių lygčių dalinėmis išvestinėmis $L_j u = 0, 1 \leq j \leq M$, kurias aproksimuojame skaitiškai $L_j^h U = 0, 1 \leq j \leq M$, tuomet sprendinys yra u^h . Jis priklauso nuo parametrų $q_1, q_2, \dots, q_m$.

Optimizavimo uždavinys

$$\min_{q_1, q_2, \dots, q_m} F(u_1^h(q_1, q_2, \dots, q_m), \dots, u_M^h(q_1, q_2, \dots, q_m))$$

Lygiagretaus Vango ¹ algoritmo pagreitėjimo kreivės



¹H.H. Wang (1981). A parallel method for tridiagonal equations. ACM Transactions on Mathematical Software (TOMS), 7(2):170-183.

ALGORITHM 1: P procesų paskirstymas tarp M užduočių

Set $p[m] = 1$, for $m = 1, \dots, M$

$P = P - M$

Compute $t_m(p[m])$, for $m = 1, \dots, M$

stop = 0

while $P > 0$ & stop == 0 **do**

find j such that $t_j(p[j]) = \max_{1 \leq m \leq M} t_m(p[m])$

if $p[j] == P_j$ **then**
stop = 1

else

$p[j] = p[j] + 1$; $P = P - 1$;

end

end

Procesai P_j skirstomi atsižvelgiant į šiuos reikalavimus:

- Procesų skaičius užduočiai negali būti didinamas, jei pagreitis pradeda mažėti, didinant procesų skaičių atitinkamai užduočiai.
- Trečiojo lygmens efektyvumas yra ribojamas pasiūlytos euristikos E_{min} .

$$\begin{aligned} i \frac{\partial u}{\partial t} + \frac{\partial^2 u}{\partial x^2} &= 0, \quad x \in (a, b), t \in [0, T] \\ u(x, 0) &= u_0(x) \\ L_l u(a) &= 0, L_r u(b) = 0. \end{aligned} \quad (1)$$

Kraštinių sąlygų aproksimavimas²

$$\partial_n u = -e^{-i\frac{\pi}{4}} \left(\left(\sum_{k=1}^{\frac{m+1}{2}} q_k \right) u - \sum_{k=1}^{\frac{m-1}{2}} q_{k+1} q_{k+(m+3)/2} \varphi_k \right), \quad (2)$$

čia $\partial_n u$ yra normalinė išvestinė, parametrų skaičius $m \in \mathbb{N}$ ir φ_k gaunamas iš šios lygties

$$\frac{d\varphi_k(x, t)}{dt} + q_{k+(m+3)/2} \varphi_k(x, t) = u(x, t), \quad x = a, b, \quad k = 1, \dots, [m/2].$$

²A. Bugajev, R. Čiegis, R. Kriauzienė, T. Leonavičienė and J. Žilinskas (2017). On the Accuracy of Some Absorbing Boundary Conditions for the Schrödinger Equation. *Mathematical Modelling and Analysis*, 22(3):408-423.

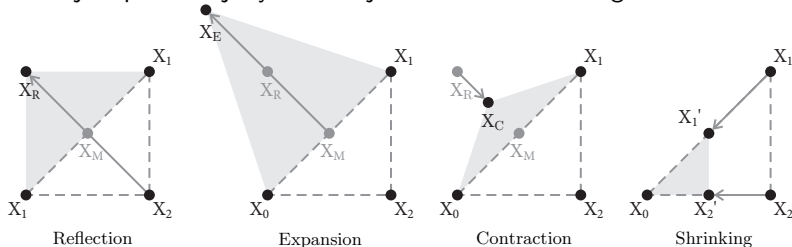
$$\min_{q_1, q_2, \dots, q_m} \max_{1 \leq j \leq M} \|u_j - u_j^h(q_1, q_2, \dots, q_m)\|_{\infty}$$

Pastebėjimai

- u_j yra žinomas, kiekviena funkcionalo reikšmė, kuri bus minimizuojama, reikalauja išspręsti M diferencialinių lygčių.
- Visos M lygčių gali būti sprendžiamos lygiagrečiai.

Pirmasis lygiagretinimo lygmuo

Lokaliajai optimizacijai yra naudojami Nelder-Mead algoritmas.



Vienos iteracijos Nelder-Mead algoritmo žingsnis $\mathbb{R}^2$ erdvėje (Weise, 2009³), čia X_i yra taškai, X_M – dviejų blogiausių taškų centroidas.

Kiekvienos iteracijos metu gali šie scenarijai

- Atspindys (vienas taškas: f_R)
- Išsiplėtimas (du taškai: f_R, f_e)
- Susitraukimas (du taškai: f_R, f_c)

Pastaba: mūsų atveju $K = 3$, $\gamma_1 = 1$, $\gamma_2 = 0.75$, $\gamma_3 = 2/3$.

³Weise, T. *Global optimization algorithms. Theory and application*. Self-Published Thomas Weise, 2009

Sprendžiamas funkcionalas

$$\max_{1 \leq j \leq M} \|u_j - u_j^h(q_1, q_2, \dots, q_m)\|_\infty = F(q_1, q_2, \dots, q_m), \quad (3)$$

čia u^h aproksimavimo paklaidos turi būti mažos palyginus su F .

Užduotys

Sprendžiamos M diferencialinių lygčių. Šių uždavinių skaičiavimo dydžiai yra nevienodi, nes kiekvienos lygties diskretizacijai turi būti naudojamas skirtingas taškų skaičius pagal laiką ir erdvę, kad būtų pasiektas toks pat tikslumas. Dėl to lygiagretinimo efektyvumas mažėja.

Trečiasis lygiagretinimo lygmuo

Tiesinių lygčių sistema

$$\begin{cases} b_0x_0 + c_0x_1 = d_0, \\ a_ix_{i-1} + b_ix_i + c_ix_{i+1} = d_i, & i = 1, \dots, N-2 \\ a_{N-1}x_{N-2} + b_{N-1}x_{N-1} = d_{N-1}, \end{cases} \quad (4)$$

čia a_i, b_i, c_i, d_i yra kompleksiniai skaičiai.

- Trijstrižinėms lygčių sistemos sprendžiamos Vango algoritmu.

Šis algoritmas leidžia skirtingiems uždaviniams priskirti skirtingą procesų skaičių.

Etaloninio uždavinio rezultatai

<i>Lygtis</i>	1	2	3	4
$J \times N$	8000 × 40000	4000 × 20000	2000 × 20000	2000 × 10000

p	16	32	64	96	128	128
	$k = 1$			$k = 2$	$k = 3$	$K = 1$
Lygtis 1	10	22	50	34	29	56
Lygtis 2	3	5	8	8	7	8
Lygtis 3	2	3	4	4	4	4
Lygtis 4	1	2	2	2	2	2
Procesų skaičius	16	32	64	96	126	70
Model. laikas	11.145	5.784	3.614	2.742	2.272	3.605
Eksp. laikas	11.003	5.394	3.608	2.719	2.308	3.600
Pagreitėjimas	12.68	25.86	38.66	51.31	60.44	38.75

Etaloninio uždavinio rezultatai naudojant efektyvumo parametrą $E_{min} > 0$

p	128		
	E_{min}		
	0.75 $k = 3$	0.8 $k = 3$	0.6 $K = 1$
Lygtis 1	26	19	42
Lygtis 2	7	5	8
Lygtis 3	4	3	4
Lygtis 4	2	1	2
Procesų skaičius p	117	84	56
Model. laikas	2.45	3.17	3.84
Eksp. laikas	2.49	3.08	3.76
Pagreitėjimas	56.03	45.37	37.11

$$u_1(t, x) = \frac{\exp(-i\pi/4)}{\sqrt{4t-i}} \exp\left(\frac{ix^2 - 6x - 36t}{4t-i}\right). \quad (5)$$

$x \in [-5, 5], t \in [0, 0.8]. J \times N = 8000 \times 4000.$

$$u_2(t, x) = \frac{1}{\sqrt{1+it/\alpha}} \exp\left(ik(x - x^{(0)} - kt) - \frac{(x - x^{(0)} - 2kt)^2}{4(\alpha + it)}\right), \quad (6)$$

čia $k = 100, \alpha = 1/120, x^{(0)} = 0.8. x \in [0, 1.5], t \in [0, 0.04],$
 $J \times N = 12000 \times 4000$

Sprendinys aprašomas (5) formule, $x \in [-10, 10], t \in [0, 2].$ Naudojamas tolygusis tinklas $J \times N = 16000 \times 10000.$

Sprendinys aprašomas (6) formule, $k = 100, \alpha = 1/120, x^{(0)} = 0.8.$
 $x \in [0, 2], t \in [0, 0.08].$ Naudojamas tolygusis tinklas
 $J \times N = 16000 \times 8000.$

Pavyzdžių sprendiniai

Dviejų lygiagrečiųjų Nelder-Mead metodų palyginimas

Palyginome mūsų metodo rezultatus su kitų autorių lygiagrečiuoju Nelder-Mead algoritmu⁴, kaip pavyzdį, pasirinkome Rozenbroko funkciją

$$f(x_1, \dots, x_d) = \sum_{i=1}^{d-1} 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2. \quad (7)$$

Lygiagretusis algoritmas⁴ gali pasiekti lygiagretumo laipsnį K , kuris yra lygus optimizavimo uždavinio dimensijai d . Tai reiškia, kad šis algoritmas potencialiai taikytinas su dideliu procesų skaičiumi.

⁴Lee, D. and Wiswall, M. (2007). A parallel implementation of the simplex function minimization routine. *Computational Economics*, 30(2): 171187.

Dviejų lygiagrečiųjų Nelder-Mead metodų palyginimas (1)

- Nagrinėjami trys atvejai $d = 3; 6; 7$: $d = 3$ - mažiausia reikšmė reikalinga lygiagretinimui, $d = 7$ - reikšmė, kurią naudojome taikymo pavyzdyje, $d = 6$ - reikšmė dėsningumui nustatyti.

γ_k apskaičiuojama tokiu būdu

$$\gamma_k = G_1 / G_k,$$

čia G_k taškų skaičius reikalingas pilnai Nelder-Mead algoritmo iteracijai, kai lygiagretumo laipsnis lygus k (pirmame lygmenyje).

Table: Tiesioginio lygiagretaus Nelder-Mead metodo γ_k reikšmės

k	$d = 3$	$d = 6$	$d = 7$
2	0.603	0.604	0.606
3	0.584	0.517	0.502

- Efektyvumo koeficientų reikšmės parodo, kad algoritmas yra stabilus ir tinkamas trijų lygių lygiagrečiame sprendiklyje, kai naudojame mažos dimensijos tikslo funkciją.

Dviejų lygiagrečiųjų Nelder-Mead metodų palyginimas (2)

Table: Lygiagretaus Nelder-Mead algoritmo (Lee and Wiswall, 2007) γ_k reikšmės

k	$d = 3$	$d = 6$	$d = 7$
2	0.668	0.685	0.714
3	–	0.436	0.454
4	–	0.023	0.104
5	–	0.001	0.002
6	–	–	0.001

- Lygiagretumo laipsnis yra ribotas, nes konvergavimo greitis mažėja, kai lygiagretumo laipsnį didiname.
- Šis metodas orientuotas į uždavinius su didele tikslo funkcijos dimensija (pavyzdžiui, finansų matematikos uždaviniai, kai $d \approx 100$).

- Pasiūlyta trijų lygmenų lygiagretinimo schema padidina lygiagretumo laipsnį, leidžiantį pasiekti papildomą pagreitėjimą. Ši metodika, lyginant su klasikiniu dviejų lygmenų lygiagretinimo algoritmu, leidžia panaudoti didesnę skaičiavimo išteklių kiekį.
- Pasiūlyta darbo krūvio paskirstymo euristika, kuri atsižvelgia į skaičiavimo laikus, gautus sprendžiant uždavinius su skirtingais jų dydžiais ir procesorių skaičiais, yra pakankamai gera – modelio prognozavimo laikas yra artimas tikram skaičiavimo laikui.
- Pasiūlyta trijų lygmenų lygiagretinimo schema, pritaikyta optimizavimo uždaviniui spręsti. Siūloma darbų paskirstymo euristika pasirenka optimalią lygiagretaus simplekso versiją pirmajame lygmenyje, t.y. dinamiškai didina lygiagretumo laipsnį pirmajame lygmenyje, kai antrojo ir trečiojo lygmenų pagreitėjimas prisisotina.
- Pasiūlyta darbų paskirstymo euristika riboja skaičiavimo išteklių kiekį, kad išsaugotų pasirinktą efektyvumą E_{min} . Skaičiavimo eksperimentų rezultatai parodė, kad šios euristikos pakanka efektyvumui valdyti trečiajame lygmenyje.

Ačiū už dėmesį.