УДК 519.6

ВІД НАУКОВОГО РЕЗУЛЬТАТУ ДО КОМП'ЮТЕРНОЇ ТЕХНОЛОГІЇ

В.К. ЗАДІРАКА, І.В. СЕРГІЄНКО

Побудовано комп'ютерні технології розв'язування складних задач обчислювальної та прикладної математики із заданими характеристиками якості за точністю та швидкодією. Технології засновано на теорії похибок обчислень, загальній теорії оптимальних алгоритмів та використанні резервів оптимізації обчислень. Розглянуто приклади розв'язування складних задач із високою якістю.

Вычислительная математика — часть информатики, относящаяся к методологии применения ЭВМ для решения задач науки, техники, производства и практически всех областей человеческой деятельности

А.Н. Тихонов, Энциклопедия математики

На сьогодні «легкі» задачі обчислювальної на прикладної математики вже перерішані. Лишилися «складні», тобто такі, які неможливо розв'язати із заданою якістю за допомогою штатного математичного забезпечення ЕОМ. Саме ці задачі стимулюють створення нових поколінь ЕОМ, сучасних чисельних методів їх розв'язування та методів діагностики якості наближеного розв'язку задачі за точністю та швидкодією. Прикладами таких задач є високоточні задачі, задачі математичного моделювання, нелінійні, великої розмірності, задачі, близькі до NP-повних, інформаційної безпеки тощо.

Все це спонукає, не зважаючи на нові перспективи в розв'язанні складних задач шляхом використання Grid-систем, методів системного аналізу [1, 2] тощо, активізувати роботи в галузі теорії похибок, загальної теорії оптимальних алгоритмів, виявлення та уточнення апріорної інформації про задачу, виявлення та використання резервів оптимізації обчислень і на базі відповідних фундаментальних досліджень запропонувати комп'ютерні технології розв'язання задач обчислювальної та прикладної математики із заданими характеристиками якості за точністю та швидкодією.

Використання таких технологій дасть змогу або розв'язати задачу із заданою якістю, або дати поради замовнику (що треба додатково зробити для розв'язання задачі), або довести: при даній інформації про задачу її неможливо розв'язати із заданою якістю.

Оскільки аналіз Фур'є широко використовується в обчислювальній практиці при розв'язанні багатьох класів задач (наприклад, краєві задачі для рівнянь в частинних похідних, спектральний та кореляційний аналіз випадкових процесів, розпізнавання образів, цифрова голографія та томографія, медична електроніка, інформаційна безпека), то будемо ілюструвати загальнотеоретичні положення цієї роботи на прикладі задачі наближеного обчислення перетворення Фур'є.

1. АНАЛІЗ ЯКОСТІ НАБЛИЖЕНОГО РОЗВ'ЯЗКУ ЗАДАЧ

Головне, на що треба звернути увагу — це гарантовані оцінки якості характеристик обчислювального алгоритму.

Спершу розглянемо таку його характеристику, як точність. Гарантією якості похибки є комплексний підхід до оцінки точності, який враховує різні джерела її накопичення (реальний обчислювальний процес супроводжують такі види похибок: неусувна за рахунок неточності вхідної інформації про задачу, а також методу та заокруглення). І тільки врахування всіх трьох видів похибки (так званої повної похибки) дасть змогу гарантувати оцінку якості наближеного розв'язку задачі. На жаль, у більшості результатів з обґрунтування тих чи інших методів наближеного розв'язання задач, як правило, проведено аналіз лише одного виду похибки і не врахована реальна обчислювальна ситуація. Через це їх можна використовувати для з'ясування якості наближеного розв'язку задачі з відповідними застереженнями.

Обчислювальна складність часто визначається вимогами до точності наближеного розв'язку, співвідношеннями складових повної похибки, залежністю похибки від типу, структури, об'єму вхідних даних та їх точності, розрядної сітки комп'ютера, правила заокруглення, типу оцінки похибок.

Розглянемо вплив величини частки окремих складових повної похибки на можливість отримання ε-розв'язку задачі за заданий час. Інформація I_0, що визначає клас задач, і $I_n(f)$, що стосується конкретної задачі, як правило, задані наближено. Нехай I_0^T, $I_n^T(f)$ — точні значення цієї інформації, а інформація I_0, $I_n(f)$ — точна для деякої задачі f; R_f і R_φ — точні розв'язки задач відповідно до I_0^T, $I_n^T(f)$ і I_0, $I_n(f)$; R_f^α, R_φ^α — наближення відповідно до R_f, R_φ, які отримані деяким алгоритмом в припущенні, що обчислення виконуються точно (без заокруглень); R_f^τ, R_φ^τ — розв'язки, отримані тим же алгоритмом з заокругленнями.

Тоді для повної похибки наближеного розв'язку $E(I,X,Y)$ (X,Y — вектори параметрів, що характеризують відповідно алгоритм і ЕОМ) можна записати

$$E(I,X,Y) = R_f - R_f^\tau = (R_f - R_\varphi) + (R_\varphi - R_\varphi^\alpha) + (R_\varphi^\alpha - R_f^\tau) =$$

$$= E_H(\bullet) + E_\mu(\bullet) + E_\tau(\bullet),$$

$$\rho(E) \le \rho(E_H) + \rho(E_\mu) + \rho(E_\tau),$$

де $\rho(\bullet)$ — деяка міра похибки наближеного обчислення розв'язку задачі; $E_H(\bullet)$ — неусувна похибка за рахунок неточності вихідної інформації про задачу; E_μ — похибка методу; E_τ — похибка заокруглень.

Розглянемо можливий розподіл доданків в умові $\rho(E_H) + \rho(E_\mu) + \rho(E_\tau) \le \varepsilon$ з урахуванням обмежень на час розв'язку задачі. Замінемо цю умову на такі:

$$\rho(E_H) \le \delta_H,\ \rho(E_\mu) \le \delta_\mu,\ \rho(E_\tau) \le \delta_\tau,$$

де

$$\delta_i \le \alpha_i \varepsilon,\ \sum_i \alpha_i = 1,\ \alpha_i \ge 0,\ i = H, \mu, \tau.$$

Згаданий розподіл визначається вибором чисел $\{\alpha_i\}$. Невдалий розподіл може значно ускладнити нашу задачу. Наприклад, при $0 \le \varepsilon - \rho(E_H) << \varepsilon$ необхідно накласти жорсткі обмеження на обидві інші складові повної похибки

$$\rho(E_\mu) + \rho(E_\tau) << \varepsilon - \rho(E_H) << \varepsilon.$$

Як відомо, оцінки можуть бути апріорні та апостеріорні, мажорантні та асимптотичні, детерміновані та стохастичні. Кожний з цих типів оцінок має свої «плюси» і «мінуси». Так, апріорні мажорантні детерміновані оцінки, з одного боку, гарантовані, достатньо легко обчислюються, а з другого — оцінка може бути дуже завищеною і малозастосовною для подальшого аналізу. Ситуація не змінюється в класі задач, якщо навіть оцінка не покращується. Задача, на якій досягається ця оцінка, може бути нетиповою для класу, тому можливості обчислювального алгоритму на інших задачах класу залишаються не використаними.

Асимптотичні апостеріорні оцінки можуть бути достатньо близькими до похибки, але така близькість досягається при значеннях параметру з певної області асимптотики, що не завжди зручно для практичних обчислень. Крім того, у обчисленні таких оцінок використовується розв'язок задачі, а для його побудови потрібен значний об'єм обчислень.

Доцільність використання оцінок певного типу визначається обчислювальною ситуацією.

Особливою цінністю користуються непокращувальні оцінки. З теоретичної точки зору вони є оцінками найвищої якості, наприклад, оцінка константи Лебега [3].

Розглянемо ітераційний поліном вигляду

$$u_n^+(\varphi, \tau) = \sum_{k=0}^{n} \alpha_k \tau^k,\ \alpha_k = \frac{1}{n+1} \sum_{j=0}^{n} \varphi(\tau_j) \tau_j^{-k},\ \tau_j = e^{i\frac{2\pi}{n+1} j},\ u_n^+ = (\varphi, \tau_j) = \varphi(\tau_j).$$

Теорема 1. Має місце оцінка

$$\left|u_n^+(\varphi,\tau)\right| \le \begin{cases} \left[\frac{4}{\pi}+\frac{2}{\pi}\ln\left(\frac{2}{\pi}(n+1)\right)\right]\max\limits_j\left|\varphi(u_j)\right| & \text{для парного } n, \\ \left[2+C+\frac{1}{n+1}+\ln\frac{n-1}{2}\right]\max\limits_j\left|\varphi(u_j)\right| & \text{для непарного } n, \end{cases} \quad (1)$$

($C = 0{,}577215$ — стала Ейлера), причому її не можна покращити в тому сенсі, що існує функція $\psi(t)$, для якої

$$\max_u\left|u_n^+(\psi,u)\right| = \left[\frac{4}{\pi}+\frac{2}{\pi}\ln\left(\frac{2}{\pi}(n+1)\right)+O\left(\frac{1}{n}\right)\right]\max_j\left|\psi(u_j)\right|.$$

Тобто оцінка (1) є асимптотично досяжною.

Константа Лебега відіграє важливу роль у теорії наближень. Знаючи її оцінку, можна легко отримати оцінку похибки наближення функції $\varphi(\tau)\in C(\gamma)$, де γ — границя одиничного кола, інтерполяційним поліномом $u_n^+(\varphi,\tau)$ (наприклад, для парного n) — $\left|\varphi(\tau)-u_n^+(\varphi,\tau)\right| \le$ $\le\left[1+\frac{4}{\pi}+\frac{2}{\pi}\ln\left(\frac{2}{\pi}(n+1)\right)\right]E_n(\varphi,\gamma)$, де $E_n(\varphi,\gamma)$ — елемент найкращого наближення $\varphi(\tau)$ в $C(\gamma)$

$$E_n(\varphi,\gamma) \le \frac{\pi}{2}\frac{\left\|u_n^+(u)\right\|c}{(n+1)^r}, \quad r=0,1,2,\ldots, \quad n=0,1,2,\ldots$$

Треба зауважити, що з практичної точки зору непокращувальні оцінки не завжди нас задовольняють, бо є досяжними на екзотичних задачах, які на практиці майже не зустрічаються. Через це для практичних задач вони завищені. Має сенс використовувати поряд з детермінованими і ймовірнісні оцінки похибки.

Наведемо детерміновано-ймовірнісні оцінки точності обчислення оцінок математичного сподівання, перетворення Фур'є, згортки, кореляційної функції, які часто використовуються при розв'язанні задач цифрової обробки сигналів, алгоритмізації неперервних виробничих процесів, теорії оптимізації тощо.

Оскільки ці ймовірнісні характеристики зустрічаються досить часто, є сенс дослідити оцінки повної похибки і окремих її видів відповідних алгоритмів їх наближеного обчислення, які в подальшому будуть використані в згаданих вище комп'ютерних технологіях.

Почнемо з оцінки математичного сподівання випадкової величини x.

$$M^*(x) = \frac{1}{N}\sum_{\upsilon=1}^{N} x_\upsilon .$$

Припустимо, що замість x_υ ми маємо справу з $x_{\varepsilon\upsilon}$, причому дисперсії

$$D(x_\upsilon - x_{\varepsilon\upsilon}) \le \varepsilon^2 .$$

Тоді замість $M^*(x)$ отримаємо

$$M_{\varepsilon}^{*}(x)=\frac{1}{N}\sum_{\upsilon=1}^{N}x_{\varepsilon\upsilon}\ .$$

Крім того, припускаючи $x_{\upsilon}-x_{\varepsilon\upsilon}$ попарно незалежними, будемо мати

$$D\left[M^{*}(x)-M_{\varepsilon}^{*}(x)\right]=\frac{1}{N^{2}}\sum_{\upsilon=1}^{N}D\left(x_{\upsilon}-x_{\varepsilon\upsilon}\right)\le\frac{\varepsilon^{2}}{N}\ .$$

Звідси, в силу відомої нерівності Чебишева

$$P\left(\left|x-M\left(x\right)\right|\le K\sqrt{Dx}\right)>1-\frac{1}{K^{2}},$$

знаходимо, що

$$\left|M^{*}(x)-M_{\varepsilon}^{*}(x)\right|\le 4{,}5\frac{\varepsilon}{\sqrt{N}}$$

з ймовірністю 0,95.

Для оцінки похибки заокруглення зауважимо, що при додаванні двох нормалізованих чисел x та y на ЕОМ (з плаваючою комою), що мають τ двійкових розрядів для представлення мантиси числа, отримуємо число $z_{\tau}=x_{\tau}\oplus y_{\tau}$, де $\oplus$ означає додавання на ЕОМ з заокругленням, причому з точністю до перших степенів $2^{-\tau}$

$$z_{\tau}=x_{\tau}\oplus y_{\tau}=\left[x(1+\xi_{x})+y(1+\xi_{y})\right](1+\eta),\quad \left|\xi_{x}\right|,\left|\xi_{y}\right|,\left|\eta\right|\le 2^{-\tau}.$$

Аналогічно

$$z_{\tau}=x_{\tau}\otimes y_{\tau}=x\left(1+\xi_{x}\right)y\left(1+\xi_{y}\right)\left(1+\eta\right),\ \left|\xi_{x}\right|,\left|\xi_{y}\right|,\left|\eta\right|\le 2^{-\tau}\ .$$

Застосовуючи ці формули до рекурентного співвідношення

$$\frac{1}{N}\otimes\sum_{\upsilon=1}^{r}\oplus x_{\varepsilon\upsilon_{\tau}}=\frac{1}{N}\otimes\left[\sum_{\upsilon=1}^{r-1}\oplus x_{\varepsilon\upsilon_{\tau}}\oplus x_{\varepsilon r_{\tau}}\right],\quad r=\overline{2,N},$$

отримаємо

$$M_{\varepsilon}^{*}\left(x\right)_{\tau}=\frac{1}{N}\sum_{\upsilon=1}^{N}x_{\varepsilon\upsilon}\left(1+\xi_{x_{\varepsilon\upsilon}}\right)\left(1+\eta_{1}\right)\left(1+\eta_{2}\right)\times\ldots\times\left(1+\eta_{N-\upsilon}\right)\left(1+\eta\right),$$

$$\left|\xi_{x_{\varepsilon\upsilon}}\right|,\left|\eta_{i}\right|,\left|\eta\right|\le 2^{-\tau}.$$

Звідси

$$\left|M_{\varepsilon}^{*}\left(x\right)-M_{\varepsilon}^{*}\left(x\right)_{\tau}\right|\le\frac{1}{N}\sum_{\upsilon=1}^{N}\left|x_{\varepsilon\upsilon}\right|\left[\left(1+2^{-\tau}\right)^{N-\upsilon+2}-1\right].$$

Якщо $r2^{-\tau}<0{,}1$, то $(1+2^{-\tau})^{r}-1<1{,}06r2^{-\tau}$.

Тому при умові $\left(N+1\right)2^{-\tau}<0{,}1$

$$\left|M_{\varepsilon}^{*}\left(x\right)-M_{\varepsilon}^{*}\left(x\right)_{\tau}\right|\le\frac{1{,}06}{N}\sum_{\upsilon=1}^{N}\left|x_{\varepsilon\upsilon}\right|\left(N-\upsilon+2\right)2^{-\tau}\le 1{,}06\max_{\upsilon}\left|x_{\varepsilon\upsilon}\right|\frac{N\left(N+3\right)}{2}2^{-\tau}.$$

При великому N похибка заокруглення може бути значною. Для того щоб уникнути цього недоліку, необхідно здійснювати додавання на ЕОМ по можливості без заокруглення. Один із таких способів діє, якщо нормалізовані числа

$$x_{\varepsilon\upsilon} = 2^{P_\upsilon}\ 0, a_1^{(\upsilon)} a_2^{(\upsilon)} \ldots a_s^{(\upsilon)} \overbrace{0\ldots0}^{\tau-1}, \quad \upsilon = \overline{1, N}$$

мають не більше s значущих цифр, причому

$$s < \tau - r,\ \left|x_{\varepsilon\upsilon}\right| < 2^r .$$

Тоді очевидно

$$\sum_{\upsilon=1}^{2^{\tau-s-r}} \oplus\, x_{\varepsilon\upsilon} = \sum_{\upsilon=1}^{2^{\tau-s-r}} x_{\varepsilon\upsilon} .$$

Якщо $N >> 2^{\tau-s-r}$, то слід використати формулу

$$\frac{1}{N}\sum_{\upsilon=1}^{N} x_{\varepsilon\upsilon} = \sum_{K=0}^{m} \frac{1}{N} \sum_{\upsilon=k\cdot 2^{\tau-s-r}}^{(k+1)2^{\tau-s-r}} x_{\varepsilon\upsilon} + \frac{1}{N} \sum_{\upsilon=(m+1)\cdot 2^{\tau-s-r}}^{N} x_{\varepsilon\upsilon} ,$$

$$(m+1)2^{\tau-s-r} \le N \le (m+2)2^{\tau-s-r} .$$

Другий спосіб полягає у використанні комп'ютерної арифметики багаторозрядних чисел [4].

Розглянемо оцінку повної похибки Δ оцінки автокореляційної функції

$$R_{xx}^*(t, t+\theta) = \frac{1}{L-j}\sum_{\upsilon=1}^{L-j} \left(x_{\varepsilon,\tau}(\upsilon\Delta t) - M_{L,\varepsilon,\tau}^*\right)\left(x_{\varepsilon,\tau}((\upsilon+j)\Delta t) - M_{L,\varepsilon,\tau}^*\right),$$

$$\left|\theta - j\Delta t\right| \le \frac{\Delta t}{2},\ j = \overline{1, M};$$

$$R_{xx}^*(t, t+\theta) = 0,\ \ j = M+1, \ldots$$

для випадкової функції вигляду $x(t) = y(t) + n(t)$, $\max\limits_{t_1,t_2} \left|x(t_1) - x(t_2)\right| \le C$, де $y(t)$ — стаціонарна нормально розподілена функція, для якої $R_{yy}(\theta) = 0$, $\theta \ge \bar{\theta},\ \ R_{yy}(\theta) \le \psi(\theta) R_{yy}(\bar{\theta}),\ \ \theta \le \bar{\theta}$; $n(t)$ — довільна функція, що не залежить від $y(t)$ і $\left|n(t)\right| \le \delta,\ \delta > 0$ з ймовірністю 0,99; $x(t)$ у вузлах $\upsilon\Delta t$ задана з похибкою ε. Тоді з ймовірністю 0,95, з точністю до малих першого порядку відносно $\delta,\ \frac{1}{\sqrt{L}},\ \omega_y\left(\frac{\Delta t}{2}\right)$ маємо $(\psi(\theta) = 1)$ [5]

$$\left|\Delta\right| \le c\left[\omega_y\left(\frac{\Delta t}{2}\right) + 4\delta + 10\sqrt{\frac{M}{L}}\right] + 1{,}06\, c^2 (L+8) 2^{-\tau} ,$$

де $\omega_y\left(\frac{\Delta t}{2}\right)$ — модуль неперервності функції $y(t)$.

На практиці зазвичай відома додаткова інформація про мажорантну функцію $\psi(\theta)$, що дозволяє отримати більш точні оцінки.

Якщо $\psi(\theta)=1-\frac{\theta}{\overline{\theta}}$, то

$$|\Delta| \le c\left[\omega_y\left(\frac{\Delta t}{2}\right)+4\delta+5\sqrt{\frac{\frac{4}{3}M+2}{L}}\right]+1{,}06c^2(L+8)2^{-\tau}.$$

Якщо $\psi(\theta)=e^{-a\frac{\theta}{\overline{\theta}}}$, то

$$|\Delta| \le c\left[\omega_y\left(\frac{\Delta t}{2}\right)+4\delta+\frac{5}{\sqrt{L}}\left(1+e^{-2j\frac{a}{L}}+\frac{4e^{-2\frac{a}{L}}}{1-e^{-2\frac{a}{L}}}\right)^{\frac{1}{2}}\right]+1{,}06c^2(L+8)2^{-\tau}.$$

Дві останні оцінки мають місце з ймовірністю 0,95 та з точністю до малих першого порядку відносно δ, $\frac{1}{\sqrt{L}}$, $\omega_y\left(\frac{\Delta t}{2}\right)$.

При великих L оцінка похибки в наведених оцінках $|\Delta|$ може бути великою за рахунок похибки заокруглення, яку можна значно зменшити, використовуючи непрямий метод обчислення $R^*_{xx}(t,t+\theta)$ з використанням теореми про дискретну згортку функцій та алгоритму швидкого перетворення Фур'є (ШПФ) [3].

$$\left\| f\left(R^*_{xx}\right)-R^*_{xx}\right\|_E < 8\cdot 1{,}06\sqrt{N_1}\,\gamma\, 2^{-\tau}\|x\|^2_E,$$

де $N_1=M+L$, $N_1=2^\gamma$.

Для квадратурної формули типу Файлона наближеного обчислення перетворення Фур'є фінітної функції з носієм $[0,T]$

$$I(\omega)=\int_0^T f(t)e^{-i\omega t}dt$$

наведемо спочатку для порівняння детерміновану та ймовірнісну оцінки неусувної похибки Δ_{H} (щоб продемонструвати, наскільки ймовірнісна оцінка точніша за детерміновану).

Нехай δ — максимальна похибка, з якою задана $f(t)$ на $[0,T]$ (для детермінованої оцінки), або, припускаючи, що можливі похибки ξ_i задання $f(t)$ у вузлах $\tilde{f}(t_i)=f(t_i)+\xi_i$, $i=\overline{0,n-1}$ є взаємнонезалежними випадковими величинами, розподіленими рівномірно на $[0,\delta]$.

Оцінки мають вигляд:

$$\Delta_{\text{Н}} \le \delta \bullet T \ \text{— детермінована;}$$

$$\Delta_{\text{Н}} \le \frac{5T\delta}{\sqrt{3n}} \ \text{— з ймовірністю 0,96 (}n\text{ — кількість вузлів).} \qquad (2)$$

Для отримання гарантованої оцінки якості наближеного розв'язку задачі треба враховувати всі види похибок (неусувну $\Delta_{\text{Н}}$, методу $\Delta_{\text{М}}$, заокруглення $\Delta_{\text{С}}$), які реально супроводжують обчислювальний процес. У зв'язку з цим наведемо оцінку повної похибки квадратурної формули типу Файлона

$$I_n(\omega) = \frac{1 - e^{-i\omega\Delta t}}{i\omega} \sum_{j=0}^{n-1} f(t_j) e^{-i\omega t_j}$$

для класу $C[0,T]$. Має місце така теорема.

Теорема 2 [3]. Нехай $f(t) \in C[0,T]$, виконується оцінка (2). Тоді з ймовірністю 0,96 маємо таку оцінку повної абсолютної похибки Δ квадратурної формули $I_n(\omega)$:

$$\Delta \le \Delta_{\text{Н}} + \Delta_{\text{М}} + \Delta_{\text{З}} \le$$

$$\le \left[\omega_f(\Delta t) + \frac{5\delta}{\sqrt{3n}} \right] T + 2^{-\tau} \max_r \left[31 + 2^{-\tau} \left| \varepsilon_{\text{ШПФ}} \right| \right] \frac{\left| \operatorname{Re} I_n(\omega_r) \right| + \left| \operatorname{Im} I_n(\omega_r) \right|}{\left| \omega_r \right|},$$

де $\varepsilon_{\text{ШПФ}}$ — похибка заокруглення алгоритму ШПФ.

Ми приділили достатньо уваги якісним конструктивним оцінкам (детермінованим, ймовірнісним, асимптотичним) повної похибки алгоритмів розв'язування деяких часто використовуваних задач. На використанні такого типу оцінок ґрунтується комп'ютерна технологія розв'язування задач із заданими характеристиками якості (див. розд. 6).

2. ВИЯВЛЕННЯ ТА УТОЧНЕННЯ АПРІОРНОЇ ІНФОРМАЦІЇ

Відомості про вихідну інформацію задачі та її якість дуже важливі з багатьох причин. Відмітимо деякі з них.

- Чим якісніша інформація про задачу, тим якісніший наближений розв'язок, на який ми можемо розраховувати.
- Максимальне використання усієї наявної інформації про задачу дає змогу звузити клас задач, що розв'язуються, і тим самим підвищує потенційну спроможність чисельного методу.
- Чим точніша вихідна інформація, тим точніші оцінки похибки і менша область невизначеності наближеного розв'язку задачі.
- На аналізі оцінок похибки ґрунтується комп'ютерна технологія розв'язування задач із заданими характеристиками якості за точністю і швидкодією.

Зупинимося на деяких аспектах виявлення та уточнення апріорної інформації про задачу.

Для отримання якісних розв'язків задач необхідна відповідна апріорна інформація. Наприклад: порядок похідної, константи, які її обмежують, константа Гельдера і відповідний показник (для задач відновлення функцій і функціоналів). Корисною може бути також інформація про геометричні властивості (опуклість, монотонність, кількість екстремумів та інші властивості). Така інформація необхідна, щоб отримати оцінку похибки наближеного розв'язку. Якщо ця інформація з достатньо великою похибкою, то неточними будуть і висновки про якість розв'язку.

Отже, отримання якісної апріорної інформації має важливе значення при розв'язанні прикладних задач. Таку інформацію отримують у фахівців, які добре знають фізичне явище, що ми вивчаємо. Ця інформація може бути отримана і за допомогою алгоритмів виявлення та уточнення апріорної інформації.

Наприклад, якщо апроксимується функція з інтерполяційного класу Ліпшица $F \equiv C_{L,N,\varepsilon}$ [3], а відомі не самі L і ε, а лише наближення до них, то у таких випадках доцільно використовувати для апроксимації функції методи нев'язки та квазірозв'язків [6].

Для класу $F \equiv C_{L,N,\varepsilon}$ апроксимуюча функція є розв'язком задачі

$$\min_{f \in F} \max_{i} \varepsilon_i . \tag{3}$$

Іншими словами, метод квазірозв'язків полягає в знаходженні функції, що найменше відхиляється від заданого набору точок $\left(x_i, \tilde{f}_i\right)$, $i = \overline{0, N-1}$, $\tilde{f}_i = f_i + \varepsilon_i$.

Розв'язок задачі (3) є лінійним сплайном $S(x, L)$, у якого максимальне відхилення від заданих точок $\left(x_i, \tilde{f}_i\right)$, $i = \overline{0, N-1}$, мінімальне [6].

$$S(x, L) = \hat{f}_i + \frac{x - x_i}{x_{i+1} - x_i}\left(\hat{f}_{i+1} - \hat{f}_i\right), \quad x \in [x_i, x_{i+1}], \quad i = \overline{0, N-1},$$

$$\hat{f}_i = \frac{\tilde{f}_i^{+} - \tilde{f}_i^{-}}{2}, \quad \tilde{f}_i^{\pm} = \max_{1 \le j \le N}\left[\pm\left(\tilde{f}_j \mp L\left|x_j - x_i\right|\right)\right], \quad i = \overline{0, N-1}. \tag{4}$$

Часто кількісна апріорна інформація, задіяна у визначенні класу F, задається у вигляді обмежень на деякий функціонал. Для класів $C_{L,N}$ і $C_{L,N,\varepsilon}$ в якості такого функціоналу $\Phi(f)$ є рівномірна норма похідної. Будемо апроксимувати функцію $f(x)$ функцією, яка є розв'язком задачі

$$\min_{f \in F} \Phi(f). \tag{5}$$

Розв'язком задачі (5) є лінійний сплайн $S(x, M)$, що визначається співвідношеннями (4) із заміною константи Ліпшица на константу M, де

$$M = \max_{1 \le j \le N} \left(0;\ \max_{j>i} \frac{\left| f_j - f_i \right| - \varepsilon_j - \varepsilon_i}{x_j - x_i} \right).$$

Розглянуті алгоритми апроксимації є оптимальними за порядком точності з константою, яка не перевищує 2 (навіть якщо порівнювати з випадком точного задання L і ε) [3]. Однак наближення, які отримані за методом нев'язки або квазірозв'язків, можуть виявитися набагато точнішими за оптимальний за точністю алгоритм апроксимації, тому що пошук цих розв'язків направлений на уточнення апріорної інформації. Застосування методів нев'язки та квазірозв'язків є одним із способів використання резервів оптимізації за точністю.

3. РЕЗЕРВИ ОПТИМІЗАЦІЇ ОБЧИСЛЕНЬ ТА ЇХ ВИКОРИСТАННЯ

Складні задачі (які не розв'язуються «штатним» математичним забезпеченням і для розв'язання яких використовуються певні резерви оптимізації обчислень) завжди були каталізатором створення нових поколінь ЕОМ та розвитку чисельних методів.

Розвиток теорії обчислень показує, що використання оптимальних за точністю та швидкодією обчислювальних алгоритмів при розв'язуванні деяких важливих класів задач (оптимізації, цифрової обробки сигналів тощо) дає ефект, порівняний з використанням нової елементної бази та архітектури ЕОМ [3, 7].

Використання резервів оптимізації обчислень дозволяє підвищити потенційну спроможність чисельних методів і розв'язати задачу із заданою якістю та надати користувачеві певні рекомендації, щоб перевести задачу з розряду нерозв'язних у розв'язні, або довести, що при даній інформації про задачу її розв'язання із заданою якістю неможливе.

Наведемо резерви оптимізації обчислень.

- Виявлення та уточнення апріорної інформації про задачу (див. розд. 2).
- Звуження класу задач за рахунок максимального використання інформації про задачу.
- Використання оптимальних та близьких до них інформаційних операторів [8,9].
- Використання якісних оцінок обчислювальних алгоритмів (див. розд. 1).
- Використання оптимальних та оптимальних за порядком (точністю та швидкодією) алгоритмів в умовах найбільш повного використання апріорної інформації про задачу [3].
- Використання алгоритмів для високоточних обчислень [5].
- Використання схем обчислень, які мінімізують швидкість накопичення похибки заокруглень.
- Підвищення точності обчислень параметрів обчислювального процесу.
- Узгодження обчислювального алгоритму з архітектурою комп'ютера.

- Розпаралелювання обчислень.
- Використання результатів тестування алгоритмів-програм [10].
- Розробка спеціалізованих обчислювачів (вибір архітектури, який найкращим чином узгоджується з обчислювальним алгоритмом розв'язування задач даного класу).

Перераховані резерви оптимізації обчислень використовуються в комп'ютерних технологіях розв'язування задач прикладної та обчислювальної математики із заданими характеристиками якості [11].

4. ОПТИМАЛЬНІ ЗА ТОЧНІСТЮ ТА ШВИДКОДІЄЮ АЛГОРИТМИ

Не завжди можна отримати ε-розв'язки деяких задач (хоча, в принципі, вхідної інформації може бути для цього достатньо) або не можна переконатися, що ε-розв'язки досягнуті. В таких випадках важливо мати оптимальні за точністю алгоритми (які з метою покращення точності максимально повно використовують всю наявну інформацію про задачу), а також апостеріорні оцінки похибки (у порівнянні з апріорними вони більш точні). Для побудови оптимальних за точністю алгоритмів можна використати метод «капелюхів» М.С. Бахвалова або «метод граничних функцій» (МГФ), розроблений в Інституті кібернетики ім. В.М. Глушкова НАН України [12, 13]. МГФ, як правило, застосовується в умовах найбільш повного використання апріорної інформації про задачу і дає можливість, на відміну від методу «капелюхів», будувати оптимальні за точністю алгоритми. При цьому треба зважати на те, що максимальне врахування апріорної інформації про задачу покращує точність, але збільшує складність алгоритму.

Як приклад розглянемо задачу побудови оптимальних за точністю та близьких до них квадратурних формул наближеного обчислення інтегралу від швидкоосцилюючої функції вигляду

$$I_1(\omega)=\int_0^1 f(t)\sin\omega t\,dt\,,$$

де $f(t)\in F$, $|\omega|\ge 2\pi$ та інформація про $f(t)$ задана не більше ніж в N точках $\{f(t_j)\}_0^{N-1}$. Наведемо результати для таких класів підінтегральних функцій: Ліпшиця (C_L) та інтерполяційного класу Ліпшиця ($C_{L,N}$) [13]. Інтерполяційні класи максимально використовують всю інформацію про підінтегральну функцію і за рахунок цього звужують клас C_L до $C_{L,N}$, функції якого не розрізняються квадратурною формулою.

Теорема 3. Нехай $f(t)\in C_L$, виконується умова А: нулі $\sin\omega t$ співпадають з $\lfloor|\omega|/\pi\rfloor+1$ вузлами t_j квадратурної формули; сітка є рівномірною. Тоді квадратурна формула типу середньої точки для інтегралу $I_1(\omega)$

$$R_1=\sum_{j=0}^{N-1} f_j\int_{t_{j-1/2}}^{t_{j+1/2}}\sin\omega t\,dt$$

оптимальна за точністю при $N \ge |\omega|$ та $N = [|\omega|/\pi]+1$, причому для оптимальної оцінки похибки на класі C_L має місце оцінка знизу

$$V_N(C_L,\omega) \ge \begin{cases} \dfrac{L}{2\pi N}, & N \ge |\omega|, \\ \dfrac{2L}{\pi|\omega|}, & N = [|\omega|/\pi]+1. \end{cases}$$

Теорема 4. Нехай $f(t) \in C_{L,N}$, виконується умова А. Тоді квадратурна формула

$$R_2 = \sum_{j=0}^{N-1} \int_{t_j}^{t_{j+1}} f^*(t) \sin \omega t \, dt,$$

де

$$f^*(t) = \begin{cases} f_j, & t_j \le t \le \bar{t}_j, \\ f_j + L(t - t_j)\sin \Delta f_j, & \bar{t}_j \le t \le \bar{\bar{t}}_j, \\ f_{j+1}, & \bar{\bar{t}}_j \le t \le t_{j+1}, \end{cases}$$

$$\bar{t}_j = \frac{t_j + t_{j+1}}{2} - \frac{|\Delta f_j|}{2L}, \quad \bar{\bar{t}}_j = \frac{t_j + t_{j+1}}{2} + \frac{|\Delta f_j|}{2L}, \quad \Delta f_j = f_{j+1} - f_j,$$

є оптимальною за точністю при $N \ge |\omega|$ та $N = [|\omega|/\pi]+1$, причому має місце оцінка знизу

$$V_N(C_{L,N},\omega) \ge \begin{cases} \dfrac{4L}{\omega^2} \displaystyle\sum_{j=0}^{N-2} \left| \sin \frac{\omega}{2}(t_j + t_{j+1}) \right| \left(\sin^2 \frac{\omega \Delta t_j}{2} - \sin^2 \frac{\omega |\Delta f_j|}{4L} \right), & N \ge |\omega|, \\ \dfrac{L}{\omega} \left[\dfrac{2}{\pi} + \dfrac{\pi}{\omega + \pi} - \dfrac{4}{\omega} \displaystyle\sum_{j=0}^{[|\omega|/\pi]} \sin^2 \frac{\omega |\Delta f_j|}{4L} \right], & N = [|\omega|/\pi]+1. \end{cases}$$

У роботах [3, 13] наведено отримані квадратурні та кубатурні формули ще для 20 класів підінтегральних функцій.

Скориставшись оптимальним за точністю алгоритмом розв'язування даної задачі і апостеріорною оцінкою похибки, нерідко можна отримати розв'язок, який задовольняє користувача, або зробити висновок, що такий розв'язок отримати неможливо.

5. ВИКОРИСТАННЯ РІЗНИХ МОДЕЛЕЙ ОБЧИСЛЕНЬ

Як було зазначено вище, вибір моделі обчислень для розв'язання задачі є одним із резервів оптимізації обчислень.

Конкретизуємо модель обчислень і розглянемо різні поняття складності. Вагомим є поняття складності задачі — це мінімальна вартість побудови ε-наближення.

Однією із частин моделі складності обчислень є набір найпростіших операцій. Алгоритми із скінченного числа найпростіших операцій називають доступними.

Звичайна модель для однопроцесорної ЕОМ. Нехай ρ — найпростіша операція. Найпростішими можна назвати, наприклад, арифметичні операції, операції порівняння, обчислення максимуму з n чисел, арифметичного кореня, інтегралу, лінійного або нелінійного функціоналу. Позначимо складність операції ρ через $\mathrm{comp}(\rho)$ і будемо вважати цю величину скінченною. Вибір множини P найпростіших операцій та визначення їх складності — це важлива проблема і її вирішення пов'язано з класом задач, що розв'язується. Питання в тому, яка повинна бути множина P, щоб задачу з певного класу можна було розв'язати з меншою складністю.

Нехай N_P — інформаційний оператор. Назвемо його допустимим відносно P, якщо існує програма обчислення $N_P(f)\ \forall f \in F$, яка складається із скінченного числа найпростіших операцій. Якщо це операції $\rho_1,...,\rho_N$, то

$$\mathrm{comp}(N_P(f)) = \sum_{i=1}^{N} \mathrm{comp}(\rho_i).$$

Назвемо число $\mathrm{comp}(N_P(f))$ інформаційною складністю оператора $N_P(f)$.

Нехай алгоритм φ використовує допустиму інформацію $N_P(f)$. Для того щоб знайти $\varphi(N_P(f))$, треба обчислити $y = N_P(f)$ та $\varphi(y)$. Назвемо алгоритм φ допустимим відносно P, якщо існує програма обчислення $\varphi(y)$ для $y = N_P(f)\ \forall f \in F$, яка складається із скінченного числа найпростіших операцій. Якщо це, скажімо, операції $q_1,...,q_j$, то

$$\mathrm{comp}(\varphi(y)) = \sum_{i=1}^{j} \mathrm{comp}(q_i).$$

Назвемо число $\mathrm{comp}(\varphi(y))$ номінальною складністю алгоритму $\varphi(y)$.

Як приклад розглянемо оцінку інформаційної та комбінаторної складності деяких алгоритмів відновлення функцій.

Нас цікавить відповідь на питання: яка мінімальна кількість інформаційних операторів про функцію, що відновлюється, і який мінімальний об'єм пам'яті ЕОМ потрібно використати для побудови ε-розв'язку задачі, тобто відновлення функції з деякого класу з точністю ε, $\varepsilon > 0$. Це, по суті, задача стиснення інформації про функцію, поставлена мовою теорії апроксимації. В якості класу функцій розглянемо клас Ліпшиця.

Нехай

$C^{\rho}_{1,C}$ — клас функцій, що задовольняє на відрізку $r[0,\rho]$ умову Ліпшиця з константою L і таких, що $|f(0)| \le C$;

Φ — метричне розширення класу $C^{\rho}_{1,C}$, яке зберігає матрицю;

$T_\varepsilon^\Phi(f)$ — таблиця $f \in C_{1,C}^\rho$, яка є впорядкованим набором $t = (t_1,...,t_p)$ елементів деякої множини ω та розшифровуючого алгоритму (р.а.) $A(t)$, який набору t ставить у відповідність деякий елемент $\varphi \in \Phi$ такий, що $\rho_\Phi(\varphi, f) \le \varepsilon$;

$P\left(T_\varepsilon^\Phi(f)\right)$ — об'єм таблиці (мінімальна кількість двійкових розрядів, необхідна для запису параметрів таблиці);

$H_\varepsilon\left(C_{1,C}^\rho\right) = \log_2 N_\varepsilon\left(C_{1,C}^\rho\right)$ — абсолютна ε-ентропія простору $C_{1,C}^\rho$, де $N_\varepsilon\left(C_{1,C}^\rho\right)$ — число елементів найбільш економного 2ε покриття множини $C_{1,C}^\rho$;

$\Phi_{1,\varepsilon}^\rho$ — сукупність функцій $\varphi(x)$, які можна представити на відрізку r у вигляді

$$\varphi(x) = L\int_0^x \varphi^*(t)dt,$$

де $\varphi^*(t)$ — функція, яка приймає значення ± 1 і є постійною на кожному з інтервалів виду

$$\delta_k : \frac{(k-1)\varepsilon}{L} < t < \frac{k\varepsilon}{L}, \quad k = \overline{1, \left[\frac{\rho L}{\varepsilon}\right]};$$

$\varepsilon_N\left(C_{1,C}^\rho\right)$ — найкраща гарантована точність на класі $C_{1,C}^\rho$ та множині пасивних алгоритмів.

Таблицею $T_\varepsilon^\Phi(f)$ може слугувати послідовність з $\alpha + \left[\log_2 \frac{C}{\varepsilon}\right] + 2$ двійкових розрядів $\alpha_1, \alpha_2, ..., \alpha_{K_1}, \beta_1, \beta_2, ..., \beta_{K_2}$, де $\alpha = \left[\frac{\rho L}{\varepsilon}\right]$, $K_1 = \alpha + 1$, $K_2 = \left[\log_2 \frac{C}{\varepsilon}\right] + 1$, які в залежності від $\varphi^*(t)$ обираються таким чином: $\alpha_P = 0$, якщо на елементарному відрізку δ_k, $k = \overline{1, \alpha}$ $\varphi^*(t) < 0$ і $\alpha_P = 1$ в іншому випадку, $p = \overline{1, K_1}$; $\beta_1, ..., \beta_{K_2}$ — коефіцієнти двійкового розкладу числа $f(0)$.

Вираз $f(x) \approx f(0) + \int_0^x \varphi^*(t)dt$ є р.а., що по t обчислює $f(x)$ в будь-якій точці $x \in r$ з точністю ε. При цьому [14]

$$P\left(T_\varepsilon^\Phi(f)\right) = \alpha + \left[\log_2 \frac{C}{\varepsilon}\right] + 2,$$

$$\frac{\rho L}{\varepsilon} + \log_2 \frac{C}{\varepsilon} - 2 \le H_\varepsilon\left(C_{1,C}^\rho\right) \le \frac{\rho L}{\varepsilon} + \log_2 \frac{C}{\varepsilon} + 2, \quad (\varepsilon \le C).$$

Із наведеного співвідношення видно, що для класу $C_{1,L}^{\rho}$ не існує способів побудови таблиць з суттєво меншим об'ємом, ніж наведений.

Оцінки комбінаторної складності (затрати ресурсів ЕОМ на «внутрішні потреби» алгоритмів) та порівняльний аналіз різних алгоритмів побудови ε-розв'язків задачі наведено в роботі [15].

Постійно зростає кількість задач, для розв'язування яких недостатня продуктивність сучасних однопроцесорних комп'ютерів. Джерелами таких задач є дослідження з ядерної фізики, екології, інформаційної безпеки, космосу, моделювання клімату, довгострокового прогнозу погоди тощо.

У зв'язку з цим розглянемо можливості використання принципів паралельної обробки даних, квантової механіки та оптичних перетворювачів для організації високопродуктивних обчислень.

Модель паралельних обчислень. Важливими задачами організації швидких паралельних обчислень є розробка паралельних алгоритмів, які відображають паралелізм задачі, узгодження таких алгоритмів з можливостями паралельних комп'ютерів та організація обчислювальної системи таким чином, щоб на ній найкраще реалізовувалися алгоритми даного класу.

Для аналізу обчислювальної складності паралельних алгоритмів може бути використана ідеалізована модель паралельних обчислень, в якій k ідентичних процесорів мають необмежену пам'ять, що доступна кожному з них без конфліктів. Один процесор за одиничний інтервал часу (крок обчислень) може точно виконати одну з бінарних арифметичних операцій. Часом реалізації інших операцій можна знехтувати. k операцій, виконаних паралельно, здійснюються за один крок обчислень. Час розв'язання задачі дорівнює числу паралельних кроків обчислень.

Паралельний алгоритм, реалізований на k процесорах, будемо називати k-алгоритмом. Як міра паралелізму k-алгоритму використовується коефіцієнт прискорення паралельного алгоритму $S_k = T_1 / T_k$, де T_k — час обчислень при розв'язанні задачі k-алгоритмом, T_1 — мінімальний час розв'язання задачі на одному процесорі. У більшості випадків відомі лише оцінки T_1. Ця величина може відноситись до послідовного алгоритму, що розпаралелюється.

Враховуючи те, що в паралельних алгоритмах обсяг обчислень може бути більший, ніж у кращого (послідовного), тобто не усі k процесорів використовуються в k-алгоритмі на кожному кроці обчислень, використаємо співвідношення $T_k = V_k T_1 \varphi(k)$, де $V_k \geq 1$ — величина, яка характеризує збільшення обсягу обчислень при переході від послідовного до паралельного алгоритму, $\varphi(k) = \sum_{i=1}^{k} \frac{a_i}{i}$, $\sum_{i=1}^{k} a_i = 1$, $a_i \geq 0$, a_i — частина обсягу обчислень, яка виконується синхронно i процесорами при реалізації k-алгоритму.

Нехай $T_{k_1} = \upsilon_k \dfrac{T_1}{k}$ — час обчислень при рівномірному розподілі обсягу обчислень між процесорами. Тоді $T_k = T_{k_1} + T_{k_2}$, де $T_{k_2} = \upsilon_k T_1 \left(\varphi(k) - \dfrac{1}{k} \right)$ — витрати часу за рахунок простоїв деяких процесорів. Формула для прискорення може бути записана у вигляді

$$S_k = \frac{k}{\upsilon_k (1 + \delta_k)}, \ \delta_k = \frac{T_{k_2}}{T_{k_1}}.$$

Тобто максимальне прискорення $(S_k = k)$ досягається тоді, коли перехід від послідовного до паралельного алгоритму відбувається без збільшення обсягу обчислень $(\upsilon_k = 1)$ при рівномірному розподілі обчислень між процесорами $(\alpha_k = 1)$.

Чи завжди паралельні алгоритми можна застосовувати до реальних обчислень? Не завжди. Паралельні алгоритми можуть бути чисельно нестійкими. Це, наприклад, алгоритми обчислення арифметичних виразів, подвоєння. При розв'язанні задач лінійної алгебри з розпаралелюванням обчислень чисельна стійкість може істотно погіршитися при значному збільшенні числа процесорів [25]. Разом з тим, паралельний алгоритм методу «пристрілки» розв'язання крайових задач для звичайних диференційних рівнянь має кращу чисельну стійкість, ніж послідовний.

Витрати на обмін даними можуть бути обмежуючим фактором в ефективному використанні паралельного алгоритму. У конкретних паралельних комп'ютерах реалізовані певні форми паралельної обробки даних, тому для ефективних паралельних обчислень функціональні зв'язки алгоритму повинні бути узгоджені з комутаційними схемами комп'ютера.

Для отримання прискорення, близького до максимального, повинен бути достатньо великий обсяг обчислень на одиницю витрат при підготовці даних, достатньо малі простої та надлишок обчислень.

Модель обчислень для квантової ЕОМ. Як відомо, при будь-яких здобутках у розвитку ЕОМ завжди існують задачі, що їм «не під силу». Скоріш за все суперкомп'ютери майбутнього не зможуть розв'язувати задачі, які мають експоненціальну і більші складності (PSPACE, EXPTIME тощо).

Однією з таких задач є задача практичної криптостійкості алгоритму RSA двоключової криптографії. Криптостійкість алгоритму RSA залежить від обчислювальної складності розв'язання задачі факторизації — розкладання двоскладового модуля (добутку двох простих чисел) на множники. Факторизація модуля дає змогу розкрити секретний ключ і, як наслідок, дешифрувати будь-яке повідомлення, підробити цифровий підпис. Чим більше число, тим більша обчислювальна трудомісткість факторизації. Основне завдання при виборі модуля RSA — одночасне забезпечення криптостійкості та обчислювальної ефективності процедури шифрування/дешифрування. Таким чином, при виборі модуля необхідно виходити з реальних оцінок зростання потужності комп'ютерів та досягнень в галузі теорії чисел.

Час роботи найкращих (на сьогодні) алгоритмів факторизації числа x, яке записується за допомогою n двійкових розрядів, оцінюється виразом $O\left(2{,}08\, n^{1/3}(\log n)^{2/3}\right)$, тому рекомендована мінімальна довжина RSA — 768 бітів (або≈ 230 десяткових знаків). Продуктивність комп'ютера для ефективної факторизації такого числа повинна бути не меншою, ніж 10^8 МУ (1 МУ — рік роботи комп'ютера з продуктивністю 1 мільйон цілочисельних інструкцій в секунду).

Нещодавно розроблено алгоритм факторизації числа на квантовому комп'ютері, час роботи якого оцінюється виразом $O\left(n^{2+\varepsilon}\right)$, де ε — деяке мале число [16]. Це означає, що криптосистеми з відкритими ключами, криптостійкість яких ґрунтується на складності задачі факторизації та дискретного логарифму, можуть бути зламані.

Таким чином, існують моделі обчислень, які дозволяють прискорити процес обчислень для деяких спеціальних класів задач.

Наведемо прикладні задачі, на яких використання квантових моделей обчислення [17] дає виграш (за часом розв'язання) порівняно з класичними. По-перше, на квантовому комп'ютері можна моделювати довільну квантову систему за поліноміальне число кроків. Це дає змогу (за наявності квантового комп'ютера) передбачити властивості молекул і кристалів, проектувати мікроскопічні електронні пристрої. Інший приклад ґрунтується на роботі П. Шора [16], який показав розв'язність на квантових машинах задачі знаходження дискретного логарифму числа в мультиплікативних цілочисельних групах і задачі розкладання цілого числа на множники за поліноміальний (відносно довжини числа) час.

Прослуховування інформації, що передається квантовим каналом зв'язку, неможливе без внесення в повідомлення спотворень, які можуть бути виявлені користувачами каналу. Цей ефект дає можливість двом абонентам таємно обмінюватися інформацією «під носом» у противника, який хоче її перехопити.

Наскільки складно буде побудувати квантовий комп'ютер? Квантові обчислення ґрунтуються на побудові траєкторії від стандартного початкового стану до складного кінцевого. Найбільша проблема — це надчутливість до збурень, які порушують траєкторію випадковим чином. Такі збурення відбуваються завдяки неконтрольованим зв'язкам із зовнішніми шумами. Здається, що немає фундаментальних обмежень на те, як добре ми можемо ізолювати систему. На сьогоднішній день теоретики і експериментатори різних країн досліджують деякі реалізації квантових комп'ютерів.

З'явилось повідомлення [26], що 13 лютого 2007 р. відбулася демонстрація квантового комп'ютера Orion. Його створила компанія D-Wave.

Модель обчислень для оптичних ЕОМ. Динамічна голографія [18] є перспективним засобом реалізації різних оптичних перетворювачів. Зокрема, ефект перекачки енергії пучка світла в когерентний йому пучок, що скерований в іншому напрямку (завдяки їх перетину в динамічному середовищі), фактично є оптичним аналогом транзистора. Змінюючи інтенсивність підсилюючого пучка, можна керувати часовими змінами його інтенсивності.

Існує варіант оптичного аналогу електронного транзистора, коли таке керування досягається шляхом зміни не інтенсивності, а фази підсилюючого пучка. Ще одним прикладом може бути оптичний перемикаючий пристрій, який діє аналогічно швидкодіючому електронному перемикачу, що є невід'ємною частиною найважливіших приладів обчислювальної техніки. Перевагою голографії є також можливість перетворення найбільш складних зображень, а не тільки найпростіших плоских або сферичних хвиль.

На сьогодні вже виконані експерименти зі створення оптичних бістабільних пристроїв, що перемикаються за 10^{-12} с, та елементів волоконно-оптчиних ліній зв'язку, інформація з яких переноситься за допомогою оптичних солітонів з довготриваліcтю, що досягає 10^{-13} с. З перемиканням за такий час продуктивність оптичного процесора, який має $10^5 - 10^6$ паралельних каналів, склала б до 10^{18} операцій за секунду, тобто на 6 порядків вища за потенційну продуктивність електронних схем. Прикладами елементарних операцій для оптичного комп'ютера є додавання і віднімання зображень, обчислення перетворення Фур'є, розпізнавання образів тощо.

6. НАУКОЄМНІ КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ РОЗВ'ЯЗАННЯ ЗАДАЧ ОБЧИСЛЮВАЛЬНОЇ ТА ПРИКЛАДНОЇ МАТЕМАТИКИ

Під комп'ютерною технологією (КТ) розв'язання задач прикладної та обчислювальної математики із заданими значеннями характеристик якості будемо розуміти вибір і побудову обчислювальних ресурсів і способів ефективного їх використання при обчисленні наближеного розв'язку задачі із заданою точністю за обмежений процесорний час.

Загальна схема розв'язування задач прикладної та обчислювальної математики з використанням КТ складається з таких етапів:

1. Постановка прикладної задачі в термінах предметної області.
2. Вибір математичної моделі прикладної задачі (ММ).
3. Вибір комп'ютерної моделі обчислень (КМО), яка має складові:

- вхідні дані про задачу;
- клас задач обчислювальної математики на основі вхідних даних;
- клас обчислювальних алгоритмів (о.а.) обчислення розв'язку, побудова оцінок характеристик якості та параметрів обчислювального процесу (ОП);
- архітектура комп'ютера;
- програмне забезпечення ОП;
- обмеження на значення характеристик якості.

4. Можливі корегування ММ, складових комп'ютерної моделі обчислень та повторний розгляд етапів цієї схеми.
5. Побудова ОП і здійснення обчислень.
6. Інтерпретація результатів обчислень.

Загальною схемою побудови розв'язків породжується множина КТ у залежності від глибини розробки і конкретного використання наведених етапів. До факторів, які породжують множину КТ, належать: тип задачі і

ММ, доступні вхідні дані про задачу (вид, об'єм, точність), вимоги до наближеного розв'язку задачі та обмеження на обчислювальні ресурси (процесорний час, пам'ять комп'ютера), можливості обчислювальної техніки, алгоритмічне та доступне програмне забезпечення, кваліфікація розробників та користувачів.

Постановка задачі: обчислити наближений розв'язок задачі за умов

$$\rho\big(E(I,X,Y)\big)\le\varepsilon, \tag{6}$$

$$T\big(\varepsilon,I,X,Y)\big)\le T_0(\varepsilon), \tag{7}$$

$$M\big(\varepsilon,I,X,Y)\big)\le M_0(\varepsilon), \tag{8}$$

де $\rho(\bullet)$ — деяка міра похибки наближеного розв'язку задачі; $E(I,X,Y)$, як правило, — повна похибка наближеного розв'язку, яка є сумою трьох складових: $E_{\text{Н}}(\bullet)$ — неусувної похибки, $E_{\mu}(\bullet)$ — похибки методу, $E_{\tau}(\bullet)$ — похибки заокруглення; X, Y — вектори параметрів, які характеризують відповідно алгоритми і комп'ютери; $T(I,X,Y)$, $M(I,X,Y)$ — процесорний час та пам'ять комп'ютера, необхідні для обчислення наближеного розв'язку; ε, $T_0(\varepsilon)$, $\mu_0(\varepsilon)$ — обмеження, задані на основі вимог до математичного моделювання і властивостей вхідної інформації (об'єму, точності, структури).

Наближений розв'язок, для якого виконана умова (6), називається ε-розв'язком, $A(\varepsilon,X,Y)$ — множина о.а. побудови ε-розв'язку в даній КМО.

Обчислювальний алгоритм, який задовольняє умовам (6), (7), називається T-ефективним, $A(\varepsilon,T_0,X,Y)$ — множина T-ефективних о.а. в даній КМО.

Опишемо коротко послідовності кроків КТ.

Крок 1. ММ задачі, яку ми розв'язуємо на основі її вхідних даних, відноситься до відповідного класу обчислювальної математики. Вивчається вхідна інформація.

Крок 2. Встановлюються вимоги (6)–(8) до значень характеристик якості розв'язку задачі до точності ($\varsigma > 0$), до комп'ютерного часу $T_0(\varepsilon)$ побудови ε -розв'язку, до пам'яті комп'ютера M_0, які повинні бути забезпечені відповідною о.а.-програмою.

Крок 3. Із використанням вхідної інформації про задачу і оцінок точності їх задання (обчислення) знаходиться оцінка $E_{\text{Н}}$ неусувної похибки розв'язку задачі.

Крок 4. Для знайденої оцінки $E_{\text{Н}}$ перевіряється виконання умови

$$\rho\big(E_{\text{Н}}\big)\le\alpha_1\varepsilon,\ \ 0<\alpha_1<1,\ \text{наприклад}\ \ \alpha_1=\frac{1}{3}. \tag{9}$$

Якщо умова (9) виконується, то перейти на крок 6, інакше — на крок 5.

Крок 5. Для забезпечення розв'язку задачі з точністю (6) треба змінити (якщо це можливо) вимогу до точності (збільшити ε) або зменшити (по-

кращити) оцінку неусувної похибки $E_{\text{Н}}$, використовуючи резерви її покращення. Перейти відповідно на крок 2 або 3. Якщо $\varepsilon_{\text{Н}}$ значно перевищує задане ε і не може бути зменшена, треба змінити саму постановку задачі і перейти на крок 1.

Крок 6. На основі апріорної інформації про задачу та вимог до значень характеристик якості розв'язку (6)–(8) звужується клас задач, що розв'язується. Це дає змогу розраховувати на більші можливості по забезпеченню точності та швидкодії побудови розв'язку задачі.

Крок 7. Наслідком кроку 6 є такі ситуації:

а) для розв'язання підкласу задач метод не відомий; перейти на крок 8;

б) відомі методи розв'язання підкласу задач, але не розроблені на їх основі о.а. і програми та теоретично не вивчені оцінки їх характеристик (E,M,T); перейти на крок 9;

в) маємо програму підкласу задач, але вона не має оцінок характеристик (E,M,T), за допомогою яких можна було б говорити про забезпечення розв'язку задачі з заданими характеристиками якості; перейти на крок 15;

г) для підкласу задач є програма, яка має оцінки характеристик (E,M,T) і забезпечує побудову ε-розв'язку задачі з підкласу із заданою точністю; перейти на крок 16.

Крок 8. Для підкласу, якому належить задача, не відомий метод. Здійснюється розробка методу.

Крок 9. Із множини методів розв'язання задачі обирається кращий за точністю та швидкодією.

Крок 10. Обираються параметри Y комп'ютера, на якому задача буде розв'язуватися.

Крок 11. На основі обраного або розробленого методу будується T-ефективний о.а. обчислення ε-розв'язку і знаходяться його характеристики (E,M,T), згідно з описаною в роботі [19] технологією.

Крок 12. Розроблений T-ефективний о.а. реалізується в програмі на відповідній алгоритмічній мові.

Крок 13. На визначеному наборі тестових задач здійснюється тестування програми у відповідності до технології тестування характеристик якості програм [10]. Якщо результати тестування підтверджують виконання умов (6)–(8), то перейти на крок 17, інакше — на крок 14.

Крок 14. Проводиться аналіз програми для виявлення резервів її покращення за тими характеристиками, які не задовольняють вимогам (6)–(8). Здійснюється оптимізація програми, і для отриманої її модифікації виконується робота, згідно з технологією [19]. Якщо вичерпані всі резерви оптимізації для вибраного о.а. і відповідної програми, але їх характеристики не задовольняють обмеженням (6)–(8), то необхідно продовжити пошук о.а., згідно з попередніми кроками (тобто перейти на крок 10 та 11).

Крок 15. Здійснюється тестування програми розв'язання підкласу задач і отримання її оцінок характеристик (E,M,T) у відповідності до технології [10]. Якщо результати тестування підтверджують, що обрана програма може забезпечити наближений розв'язок з потрібними значеннями характеристик

якості за точністю та швидкодією, то перейти на крок 17, інакше — на крок 7.

Крок 16. Використовуючи оцінки характеристик (E,M,T) програми побудови розв'язків підкласу задач, перевіряється, чи може вона забезпечити ε-розв'язок задачі з заданою точністю ε (6) і при обмеженні (7) на комп'ютерний час. Якщо ця програма здійснює побудову ε-розв'язку задачі з заданими характеристиками якості (6)–(8), то перейти на крок 17, інакше — 7.

Крок 17. Будується розв'язок підзадачі з заданими характеристиками якості (6)−(8).

7. ПРИКЛАДИ РОЗВ'ЯЗУВАННЯ ЗАДАЧ З ВИСОКОЮ ТОЧНІСТЮ

Перелічимо задачі, при розв'язанні яких було важливим використання оптимальних і близьких до них алгоритмів та розробка відповідних програмних засобів для отримання наближеного розв'язку з високою якістю.

1. Для побудови математичної моделі технологічної установки АВТ (атмосферно-вакуумної трубчатки) нафтопереробного заводу використано ефективні за точністю та швидкодією алгоритми визначення оцінок динамічних та ймовірнісних характеристик об'єктів керування [5].

2. Для експрес-обробки даних льотного експерименту розроблено системи «Випробувач», «Темп», «Темп-ЕК», в яких були використані оригінальні модифікації алгоритму швидкого перетворення Фур'є і на їх основі створено ефективні за швидкодією алгоритми розв'язання задач спектрального та кореляційного аналізу випадкових процесів [3, 20, 21]. Ці системи впроваджено в Льотно-дослідному інституті (м. Жуковський, Московської обл.).

3. Розроблено комплекс пакетів програм «ПОМ-1» наближеного розв'язання типових задач обчислювальної математики з діагностикою якості [22]. «ПОМ-1» випробувано у Фізико-енергетичному інституті (м. Обнінськ), Обчислювальному центрі (м. Єреван), Обчислювальному центрі Київського національного університету ім. Тараса Шевченка та інших установах.

4. Розроблено комп'ютерну технологію тестування якості прикладного програмного забезпечення за точністю та швидкодією [10].

5. Розроблено апаратно-програмний комплекс арифметики багаторозрядних чисел для систем захисту інформації [23]. Ця робота була виконана в рамках Державного замовлення 8/35 у 2004 р., і результати її впровадження дозволяють підняти продуктивність систем двоключової криптографії.

6. На базі спектрального аналізу випадкових процесів та теорії похибки заокруглення розроблено нові комп'ютерні технології проектування стійких стеганоконтейнерів для розв'язання задач приховування інформації [24].

7. Розроблено комп'ютерну технологію розв'язування задач прикладної та обчислювальної математики із заданими характеристиками якості (див. розд. 6 та роботу [11]).

ЛІТЕРАТУРА

1. *Згуровський М.З., Панкратова Н.Д.* Системний аналіз: проблеми, методологія, застосування. — Київ: Наук. думка, 2005. — 744 с.
2. *Сергієнко І.В.* Інформатика та комп'ютерні технології. — Київ: Наук. думка, 2004. — 432 с.
3. *Задирака В.К., Мельникова С.С.* Цифровая обработка сигналов. — Киев: Наук. думка, 1993. — 294 с.
4. *Задірака В.К., Олексюк О.С.* Комп'ютерна арифметика багаторозрядних чисел: Наук. видання. — Київ, 2003. — 264 с.
5. *Методы* алгоритмизации непрерывных производственных процессов / В.В. Иванов, А.И. Березовский, В.К. Задирака и др. — М.: Наука, 1975. — 400 с.
6. *Морозов В.А.* Регулярные методы решения некорректно поставленных задач. — М.: Изд-во Моск. ун-та, 1974. — 359 с.
7. *Сергієнко І.В.* Інформатика в Україні: становлення, розвиток, проблеми / Відп. ред. Ю.В. Капітонова, Т.Т. Лебедєва. — Київ: Наук. думка, 1999. — 354 с.
8. *Переверзев С.В.* Оптимизация методов приближенного решения операторных уравнений. — Киев: Ин-т математики НАН Украины. — 1966. — 251 с.
9. *Литвин О.М.* Методи обчислень. Додаткові розділи: Навч. посіб. — Київ: Наук. думка, 2005. — 344 с.
10. *Бабич М.Д., Задирака В.К., Сергиенко И.В.* Вычислительный эксперимент в проблеме оптимизации вычислений // Кибернетика и системный анализ. — Ч.1. — 1999. — № 1. — С. 51–63; Ч.2. — 1999. — № 2. — С. 59–79.
11. *Компьютерные* технологии решения задач прикладной и вычислительной математики с заданными значениями характеристик качества / И.В. Сергиенко, В.К. Задирака, М.Д. Бабич и др. // Кибернетика и системный анализ. — 2006. — №5. — С. 33–41.
12. *Иванов В.В.* Об оптимальных алгоритмах минимизации функций некоторых классов // Кибернетика. — 1972. — № 4. — С. 81–94.
13. *Задирака В.К.* Теория вычисления преобразования Фурье. — Киев: Наук. думка, 1983. — 216 с.
14. *Витушкин А.Г.* Оценка сложности задач табулирования. — М.: Физматгиз, 1953. — 250 с.
15. *Задирака В.К.* Об оценках информационной и комбинаторной сложности некоторых алгоритмов восстановления функций // Информатика, вычислительная и прикладная математика. Теория, приложения, перспективы: Тр. междунар. конф. INAMATAP'96. — Киев: Наук.думка, 1998. — С. 94–98.
16. *Shor P.* Algorithms for Quantum Computation: Discrete Logarithms and Foundations of Computer Science // IEEE Computer Society Press. — 1999. — **26**. — P. 124–134.
17. *Китаев А., Шень А., Вялый М.* Классические и квантовые вычисления. — М.: МЦНМО ЧеРо, 1999. — 192 с.
18. *Гиббс Х.* Оптическая бистабильность. Управление светом с помощью света. — М.: Мир, 1988. — 518 с.
19. *Т-ефективні* алгоритми наближеного розв'язання задач обчислювальної та прикладної математики: Наук. видання // Задірака В.К., Бабич М.Д., Березовський А.І. та ін. — Київ: Ін-т кібернетики ім. В.М. Глушкова НАН України, 2003. — 216 с.
20. *Дехтярюк Е.С., Задирака В.К., Ярошенко Э.В.* Пакет программ статистической обработки данных // Опыт создания и внедрения автоматизированных систем обработки данных комплексных испытаний сложных объектов: Тез. докл. научн.-техн. конф. — Киев, апрель 1978. — Киев: ИК АН УССР, 1978. — С. 73–75.
21. *Задирака В.К., Абдикаликов К.А.* Быстрые ортогональные преобразования: Теория и приложения. — Алматы: Научн.-изд. центр «Ғылым», 2003. — 220 с.

22. *Комплекс* программ по расчету и оценке основных вероятностных характеристик, аппроксимации функций, решению ряда классов особых уравнений, минимизации функций и математическому программированию / В.В. Иванов, В.К. Задирака и др. — Ин-т кибернетики им. В.М. Глушкова АН УССР. — Киев, 1985. — 1235 с. — Деп. в Гос ФАП 14.02.1986. — №50860000156.
23. *Задирака В.К., Кудин А.М.* Построение программно-аппаратных комплексов арифметики сверхбольших чисел // Компьютерная математика. — Киев: Ин-т кибернетики им. В.М. Глушкова НАН Украины. — 2001. — № 1. — С. 158–165.
24. *Задірака В.К.* Теорія обчислень та сучасні комп'ютерні технології розв'язання задач інформаційної безпеки // Искусственный интеллект. — 2006. — № 3. — С. 727–736.
25. *Воеводин В.В.* Математические модели и методы в параллельных процессах. — М.: Мир, 1991. — 365 с.
26. *Шевченко В.* Квантовый компьютер // Компьютерное обозрение. — 29.03.2007. — URL: http://itc.ua/27644.

Надійшла 27.06.2007