

ПАРАЛЕЛЬНИЙ ВИРІШУВАЧ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ НА БАЗІ БЛОКОВИХ ОБЧИСЛЮВАЛЬНИХ МЕТОДІВ

Мірошкін О.М., Маргієв Г.Е.

Більшість сучасних програмних пакетів для вирішення диференціальних рівнянь побудовані за послідовними алгоритмами, які не можуть забезпечити ефективного розпаралелювання при застосуванні на паралельних архітектурах. Одною з головних проблем при розробці розподіленого паралельного моделюючого середовища є пошук алгоритмів для побудови вирішувача, який зможе виконувати пошук рішення із заданою точністю у встановлений час, та використовуватиме архітектурно-релевантні принципи по відношенню наявних обчислювальних засобів. Робота присвячена практичному застосуванню чотириточкового блокового чисельного методу задля побудови програмного вирішувача диференціальних рівнянь з урахуванням особливостей наявних паралельних обчислювальних архітектур. Запропонований вирішувач був запрограмований мовою програмування C, формат виклику якого було обрано на основі аналізу інших вирішувачів диференціальних рівнянь та їх систем та адаптовано з урахуванням особливостей обраної мови програмування. Отриманий вирішувач було успішно перевірено на тестових задачах на основі систем рівнянь першого і другого ступенів. Тестування показало високу стійкість та точність при розв'язанні систем кількістю до 10,000 рівнянь. В статті наведено приклад формату використання розробленого вирішувача при чисельному моделюванні аеродинамічних процесів у фрагменті шахтної вентиляційної мережі.

Ключові слова: вирішувач, диференціальні рівняння, блоковий чисельний метод, розпаралелювання, архітектурно-релевантний підхід.

Вступ. Для вирішення задачі чисельного розв'язання диференціальних рівнянь зазвичай використовують програмні вирішувачі, які реалізують чисельні методи Ейлера [1], Сімпсона [2], Адамса [3], Рунге-Кути [4]. Основою цих методів є послідовний підхід до вирішення задачі, а саме пряма залежність наступного значення функції від результатів попереднього кроку алгоритму, що призводить до обмеження в можливостях використання паралельних засобів на рівні алгоритмів [5]. В протилежність до них, блокові обчислювальні методи (БОМ) були розроблені з урахуванням необхідності виконання обчислень на паралельних комп'ютерних системах. БОМ-розробки запропоновано в роботах Фельдмана Л. П., Дмитрієвої О. А., Назарової І. А. та інших [6–8]. Задача побудови розподіленого паралельного моделюючого середовища (РПМС) [9–11] для моделювання складних динамічних систем вимагає програмної реалізації чисельних методів, які б давали змогу вирішувати різноманітні диференціальні рівняння із оптимальним залученням усіх наявних паралельних

ресурсів, реалізуючи так званий архітектурно-релевантний підхід у використанні доступних обчислювальних ресурсів. Для прикладу були обрані системи рівнянь у частинних похідних, які описують аеродинамічні процеси.

Розробка вирішувача. Згідно з [12], рівняння m -крокового k -точкового блокового методу мають наступний вигляд:

$$u_{n,i} = u_{n,0} + i\tau \left(\sum_{j=1}^m b_{i,j} F_{n,j-m} + \sum_{j=1}^k a_{i,j} F_{n,j} \right) \quad (1)$$

де $u_{n,i}$ – наближені рішення в точці i блоку n в момент часу $i\tau$; $n=1, 2, \dots, N$; $i=1, 2, \dots, K$; $u_{n,0}$ – наближене рішення в нульовій точці блоку n . Для першого блоку це значення співпадає з початковою умовою задачі Коші, для всіх наступних – із значенням останньої точки попереднього блоку, тобто:

$$u_{n,0} = \begin{cases} u(0), & n=0; \\ u_{n-i,K}, & n>0; \end{cases} \quad (2)$$

$F_{n,j} = f(t_n + j\tau; u_{n,j})$ – значення правої частини в точці j блоку n , де $n=1, 2, \dots, N$;
 $F_{n-1,j} = f(t_{n-1} + j\tau; u_{n-1,j})$ – значення правої частини в точці j блоку $n-1$; $a_{i,j}$, $b_{i,j}$ – коефіцієнти Лагранжа.

При $k=4$ в роботі [13] отримано формули (3) для чотириточкового блокового методу.

Важливою особливістю цих формул є те, що обчислення точок блоку виконується незалежно, що дає змогу застосувати розпаралелювання. На основі формул (3) розроблено програмний вирішувач, загальний алгоритм якого наведено на рис. 1.

Виклик вирішувача. Формат виклику вирішувача для зручності зроблено у звичний для інших програмних вирішувачів спосіб. Далі наведені приклади викликів вирішувачів в різних математичних програмних пакетах. Система Matlab використовує наступний формат виклику функцій [14]:

`[t,y]=ode45(odefun, tspan, y0, options),`

система Mathcad поряд з іншими пропонує для використання наступну функцію [15]:

`rkfixed(init, x1, x2, intvls, D),`

а в системі Wolfram Mathematica можна вирішити диференціальне рівняння наступним чином [16]:

`DSolve[{y'[x] == z[x], z'[x] == -y[x], y[0] == 0, z[0] == 1}, {y[x], z[x]}, x].`

Виклик вирішувачів в інших математичних бібліотеках має свої особливості, але використовує ті ж основні параметри: межі діапазону пошуку рішення, початкові значення, спосіб обчислення значень правої частини.

Виходячи з аналізу вирішувачів у відомих математичних пакетах та алгоритму роботи вирішувача блокового чисельного методу, пропонується наступний формат виклику розробленого вирішувача:

`int solver(*fn, t0, tf, *y0, nf, **x, **y, *n),`

де `*fn` – посилання на функцію обчислення правих частин системи диференціальних рівнянь; `t0`, `tf` – межі діапазону пошуку

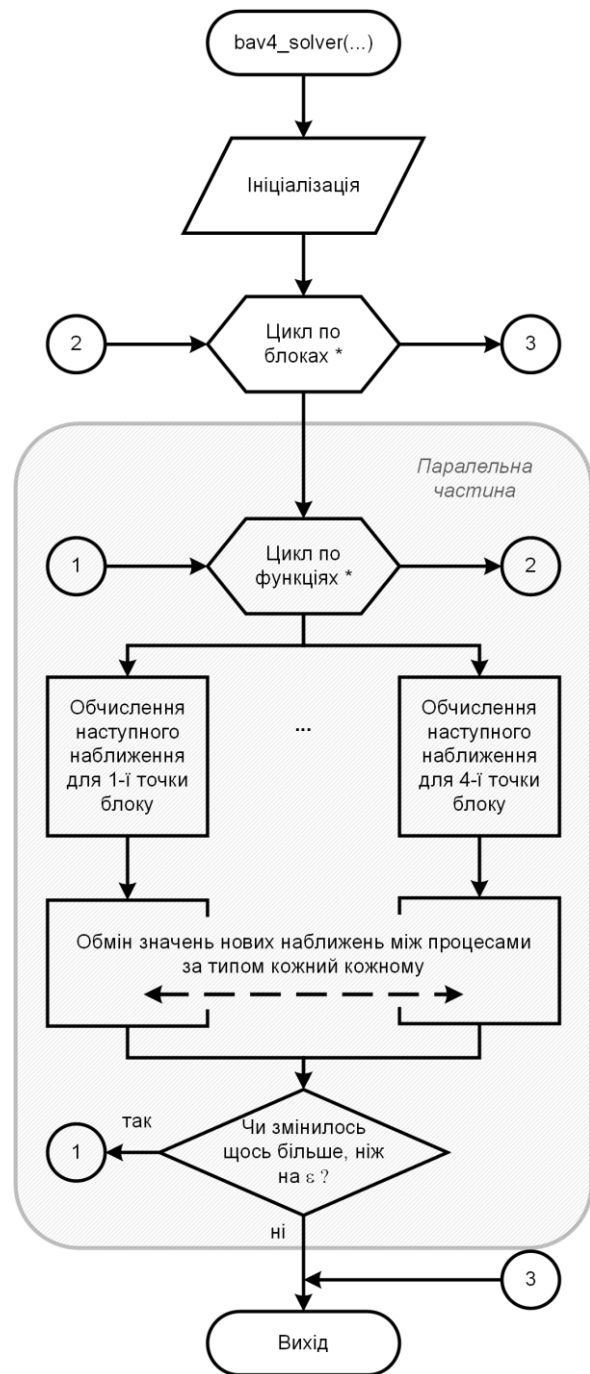


Рис. 1 – Узагальнений алгоритм роботи паралельного блокового вирішувача

рішення; `*y0` – масив початкових значень; `nf` – кількість функцій в системі; `**x`, `*n` – вихідний масив із значеннями абсцис точок, в яких знайдено рішення системи та їх кількість відповідно; `**y` – матриця знайдених значень.

$$\begin{aligned} u_{n,1} &= u_{n,0} + \frac{\tau}{720} (251F_{n,0} + 646F_{n,1} - 264F_{n,2} + 106F_{n,3} - 19F_{n,4}); \\ u_{n,2} &= u_{n,0} + \frac{\tau}{90} (29F_{n,0} + 124F_{n,1} + 24F_{n,2} + 4F_{n,3} - F_{n,4}); \\ u_{n,3} &= u_{n,0} + \frac{3\tau}{80} (9F_{n,0} + 34F_{n,1} + 24F_{n,2} + 14F_{n,3} - F_{n,4}); \\ u_{n,4} &= u_{n,0} + \frac{2\tau}{45} (7F_{n,0} + 32F_{n,1} + 12F_{n,2} + 32F_{n,3} + 7F_{n,4}); \end{aligned} \quad (3)$$

Збільшена кількість параметрів та певні особливості їх використання зумовлені особливостями мови програмування C, що була використана для реалізації вирішувача.

Розпаралелювання. Система формул (3) має чотири незалежних рівняння, обчислення яких доцільно виконати на чотирьох обчислювальних елементах. За умови використання процесору з кількістю обчислювальних ядер 4+, для обчислення точок блоку однієї функції доцільно використовувати переваги технології OpenMP [17], а при розв'язанні систем диференціальних рівнянь доцільне використання декількох процесорів та бібліотеки MPI [18] для обміну даними між ними.

Приклади використання.

Вихідні дані для моделювання. Однією з задач, які вирішуються за допомогою РПМС на кафедрі комп'ютерної інженерії ДонНТУ (м. Покровськ), є моделювання шахтних вентиляційних мереж (ШВМ) як складних динамічних об'єктів. Розглянемо одну з моделей ШВМ, яка задана за допомогою системи диференціальних рівнянь у частинних похідних:

$$\begin{cases} -\frac{\partial P_j}{\partial \xi} = r_j Q_j^2 + \frac{\rho}{F_j} \frac{\partial Q_j}{\partial t} + r_j(\xi_r, t) Q_j^2; \\ -\frac{\partial P_j}{\partial t} = \frac{\rho a^2}{F_j} \frac{\partial Q_j}{\partial \xi}. \end{cases} \quad (4)$$

Докладне пояснення параметрів моделі можна знайти в роботах [10, 19]. Із застосуванням розробленого вирішувача виконаємо моделювання гілки ШВМ довжиною $L = 1000$ м з кроком за просторовою координатою $\Delta \xi = 100$ м, площею поперечного перерізу $F = 5.98$ м² та питомим аеродинамічним опором $r = 1.9e-03$. Одним кінцем гілка, що моделюється, виходить в атмосферу, вважатимемо, що

відносний тиск на цьому кінці гілки дорівнює нулю ($P_{ATM} = 0$), а на іншому кінці гілки є вентилятор, який створює депресію $H = -1$ кПа відносно атмосферного тиску.

Функція обчислення правих частин системи. Після перетворення системи для придатного для обчислення вигляду функція обчислення правих частин системи з 19 рівнянь матиме наступний вигляд:

```
int RHS(double x, double *y, int n, double *rhs)
{
    /* Обчислення допоміжних коефіцієнтів *alpha,
    beta та gamma
    */
    double Patm = 0.0, Hvent = -1000.0;
    rhs[ 0] = alpha * (Patm-y[1]) - beta * y[0] *
    fabs(y[0]);
    rhs[ 1] = gamma * (y[0]-y[2]);
    rhs[ 2] = alpha * (y[1]-y[3]) - beta * y[2] * fabs(y[2]);
    ...
    rhs[17] = gamma * (y[16]-y[18]);
    rhs[18] = alpha * (y[17]-Hvent) - beta * y[18] *
    fabs(y[18]);
}
```

Парні елементи масиву rhs відповідають потокам повітря у фрагментах гілки ШВМ після апроксимації моделі за просторовою координатою. Непарні елементи масиву містять значення тиску на межах між фрагментами гілки ШВМ.

Синтаксис виклику вирішувача

```
int main(int argc, char** argv)
{
    int NumSections = 10;
    int i, nx, nf = 2 * NumSections - 1;
    double *x, *y, y0[nf];
    for (i = 0; i < nf; i++)
        y0[i] = 0.0;
    solver_ex(RHS, 0.0, 100.0, &y0[0], nf, &x, &y, &nx,
    0.1, 0);
    printXY_Excel(x, y, nx, nf, 1);
    return 0; }
```

Результати обчислень. На рис. 2 та рис. 3 наведені результати моделювання фрагменту шахтної вентиляційної мережі за допомогою розробленого вирішувача на базі

чотириточкового блокового чисельного методу з кроком за часом $t = 0.01$.

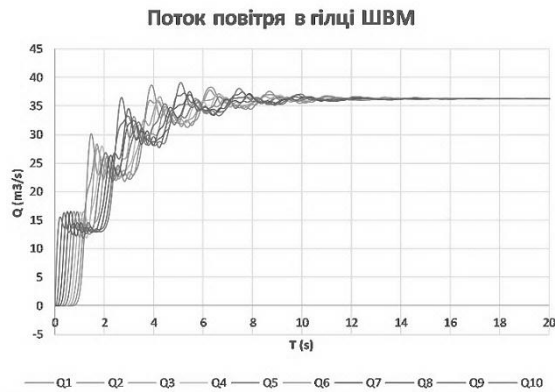


Рис. 2 – Результати моделювання. Потіки повітря в фрагментах гілки ШВМ

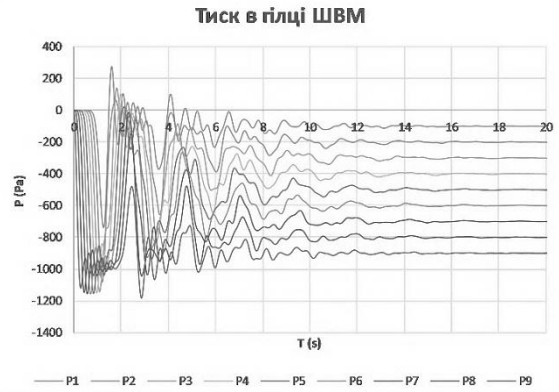


Рис. 3 – Результати моделювання. Тиск в фрагментах гілки ШВМ

Подальші дослідження. Подальша робота над вирішувачем планується в двох основних напрямках – дослідження використання різних підходів до розпаралелювання вирішувача з метою зменшення часу виконання алгоритму на паралельних архітектурах та тестування вирішувача на різних системах диференціальних рівнянь. Крім того, обов'язковим є порівняння розробленого вирішувача із засобами розв'язання диференціальних рівнянь та їх систем, що існують, з метою виявлення класів задач та особливостей апаратних обчислювальних платформ, при яких застосування розробленого вирішувача може зменшити часові показники або збільшити точність рішення.

Висновки. У результаті дослідження було розроблено паралельний вирішувач, який реалізує алгоритм чотириточкового

блокового чисельного методу з урахуванням особливостей архітектури паралельної обчислювальної системи. Вирішувач дозволяє знаходити рішення диференціальних рівнянь та/або їх систем з заданою точністю. Подальшому дослідженню підлягають використання елементів технології OpenMP, щодо оптимізації використання обчислювальних ресурсів у обчислювальних багатоядерних вузлах зі спільною пам'яттю, а також використання розробленого вирішувача на різних рівнях розпаралелювання моделей складних динамічних систем при їх моделюванні в розподіленому паралельному моделюючому середовищі, розробкою якого займаються в лабораторії надпродуктивних обчислень Донецького національного технічного університету в м. Покровськ.

Література

- [1] Метод Ейлера : [Електронний ресурс] // Вікіпедія – вільна енциклопедія. – Режим доступу : https://uk.wikipedia.org/wiki/Метод_Ейлера
- [2] Метод Сімпсона : [Електронний ресурс] // Вікіпедія – вільна енциклопедія. – Режим доступу : https://uk.wikipedia.org/wiki/Метод_Сімпсона
- [3] Метод Сімпсона : [Електронний ресурс] // Вікіпедія – вільна енциклопедія. – Режим доступу : https://uk.wikipedia.org/wiki/Метод_Адамса
- [4] Метод Рунге — Кутти : [Електронний ресурс] // Вікіпедія – вільна енциклопедія. – Режим доступу : https://uk.wikipedia.org/wiki/Метод_Рунге_—_Кутти

- [5] OpenMP topic: Loop parallelism : [Електронний ресурс] // Parallel Programming in MPI and OpenMP by Victor Eijkhout. – Режим доступу : <http://pages.tacc.utexas.edu/~eijkhout/pcse/html/omp-loop.html>
- [6] Дмитриева О. А. Разработка многошаговых параллельных коллокационных блочных методов с использованием интерполяционных полиномов Эрмита / О. А. Дмитриева // Радиоелектронні і комп'ютерні системи. – 2013. – № 5. – С. 243–249. – Режим доступу: http://nbuv.gov.ua/UJRN/recs_2013_5_42
- [7] Устойчивые одношаговые блочные методы численного решения жестких обыкновенных дифференциальных уравнений / Л.П. Фельдман // Штучний інтелект. – 2009. – № 1. – С. 213–217. – Бібліогр.: 9 назв. – рос. – Режим доступу: <http://dspace.nbuv.gov.ua/handle/123456789/7841>
- [8] Назарова И. А. Экстраполяционные блочные одношаговые численные методы решения жестких задач Коши // Научно-теоретический журнал ИПИИ НАН Украины «Искусственный интеллект», №3, 2010. – Донецк: ИПИИ, 2010. – С. 116–126. – Режим доступу: <http://ea.donntu.edu.ua/handle/123456789/6227>
- [9] Святний В.А. Паралельне моделювання складних динамічних систем // Моделирование–2006: Международная конференция. Киев, 2006 г. – Киев, 2006. – С. 83–90.
- [10] Kushnarenko Volodymyr, Miroshkin Oleksandr, Svyatnyy Volodymyr, "Towards the simulation of dynamic network objects with distributed parameters based on block difference numerical methods". The 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications. 21-23 September, 2017, Bucharest, Romania. P. 1073-1076.
- [11] Мірошкін О.М. Архітектура розподіленого паралельного моделюючого середовища для складних динамічних об'єктів з мережною топологією / О. М. Мірошкін, В. А. Святний, А. С. Любимов // Праці VII-ї Міжнародної науково-практичної конференції «Проблеми інформатики та комп'ютерної техніки» (ПІКТ – 2018), м. Чернівці: Видавничий дім «Родовід», 2018. – С. 59-61.
- [12] Фельдман Л.П., Назарова И.А. Параллельные алгоритмы численного решения задачи Коши для систем обыкновенных дифференциальных уравнений / Л.П. Фельдман, И.А. Назарова // Математическое моделирование. – 2006. – Т. 18, № 6. – С. 17–31.
- [13] Гусева А.Б. Дискретная Simulation-модель сетевого динамического объекта с распределенными параметрами на основе блочного численного метода / А. Б. Гусева, В. Г. Кушнarenко // Проблемы моделирования и автоматизации проектирования: Сб. научн. тр. ДонНТУ, вып. 9 (179), Донецк, 2011, С. 274–283.
- [14] ode45 : [Електронний ресурс] // MathWorks Matlab R2019a Online Documentation. – Режим доступу : <https://www.mathworks.com/help/matlab/ref/ode45.html>
- [15] Решатели ОДУ : [Електронний ресурс] // Справочный центр PTC Mathcad Prime 5.0.0.0. – Режим доступу : http://support.ptc.com/help/mathcad/ru/index.html#page/PTC_Mathcad_Help/ode_solvers.html
- [16] How to | Solve a Differential Equation : [Електронний ресурс] // Wolfram Language & System Documentation Centre. – Режим доступу : <https://reference.wolfram.com/language/howto/SolveADifferentialEquation.html>
- [17] OpenMP : [Електронний ресурс] // Режим доступу : <https://www.openmp.org>
- [18] MPICH | High-Performance Portable MPI : [Електронний ресурс] // Режим доступу : <https://www.mpich.org>
- [19] Svyatnyy V. A. Parallele Modellierung und Simulation der luftdynamischen Prozesse in den Grubenbewetterungsnetzen / V. A. Svyatnyy, V. G. Kushnarenko, O. M. Miroshkin / Вісн. Донецького гірничого інституту, № 1 від 2018 р., С. 116–129. - Режим доступу: http://science.donntu.edu.ua/wp-content/uploads/2018/08/Svyatnyy_JDMI_1_18.pdf

References

- [1] “Euler method” 5 Dec. 2019. Accessed on: Dec. 6, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Euler_method
- [2] “Simpson's rule” 29 Nov. 2019. Accessed on: Dec. 6, 2019. [Online]. Available:

- https://en.wikipedia.org/wiki/Euler_method
- [3] "Linear multistep method" 27 Sep. 2019. Accessed on: Dec. 6, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Linear_multistep_method
- [4] "Runge–Kutta methods" 22 Nov. 2019. Accessed on: Dec. 6, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Runge%E2%80%93Kutta_methods
- [5] V. Eijkhout Parallel Programming in MPI and OpenMP. Accessed on: Dec. 6, 2019. [Online]. Available: <http://pages.tacc.utexas.edu/~eijkhout/pcse/html/omp-loop.html>
- [6] Dmitrieva O. A. "Razrabotka mnogoshagovykh parallel'nykh kollokatsionnykh blochnykh metodov s ispol'zovaniem interpolatsionnykh polinomov Ehrmita", Radioelektronni i komp'uterni sistemi, vol.64, no. 5, pp. 243–249, 2013. [Online]. Available: http://nbuv.gov.ua/UJRN/recs_2013_5_42
- [7] L. P. Fel'dman, "Stable One-step Block Methods of Numeral Decision of Stiff Ordinary Differential Equations", Iskustvennyy intellekt, no. 1, pp. 213–217, 2009. [Online]. Available: <http://dspace.nbuv.gov.ua/handle/123456789/7841>
- [8] Nazarova I. A. "Ehkstrapolyatsionnye blochnye odnoshagovye chislennyye metody resheniya zhestkikh zadach Koshi", Nauchno-teoreticheskiy zhurnal IPII NAN Ukrainy «Iskustvennyy intellekt» Donetsk, no. 3, pp. 116–126, 2010. [Online]. Available: <http://ea.donntu.edu.ua/handle/123456789/6227>
- [9] Sviatnyi V.A., "Paralelne modeliuvannia skladnykh dynamicnykh system" in Mezhdunarodnaya konferentsiya: Modelirovanie–2006. Kiev, 2006, Ukraine, pp. 83–90.
- [10] Kushnarenko Volodymyr, Miroshkin Oleksandr, Svyatnyj Volodymyr, "Towards the simulation of dynamic network objects with distributed parameters based on block difference numerical methods". The 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications. 21-23 September, 2017, Bucharest, Romania. P. 1073-1076.
- [11] O.M. Miroshkin, V. A. Sviatnyi, A. S. Liubymov, "Arkhitektura rozpodilenoho paralelnoho modeliuvannia seredovyscha dlia skladnykh dynamicnykh ob'ektiv z merezhnoi topolohiieiu", Pratsi VII-yi Mizhnarodnoi naukovo-praktychnoi konferentsii «Problemy informatyky ta kompiuternoi tekhniky» (PIKT – 2018), Vydavnychiy dim «Rodovid», Chernivtsi, 2018, Ukraine, pp. 59-61.
- [12] L.P. Fel'dman, I.A. Nazarova, "Parallel'nye algoritmy chislennogo resheniya zadachi Koshi dlya sistem obyknovennykh differentsial'nykh uravneniy", Matematicheskoe modelirovanie, vol. 18, no. 6 pp. 17–31, 2006.
- [13] A.B. Guseva, V. G. Kushnarenko "Diskretnaya Simulation-model' setevogo dinamicheskogo ob'ekta s raspredeleennymi parametrami na osnove blochnogo chislennogo metoda", Problemy modelirovaniya i avtomatizatsii proektirovaniya: Sb. nauchn. tr. DonNTU, vol. 179, no. 9, Donetsk, 2011, Ukraine, pp. 274–283.
- [14] "ode45 MathWorks Matlab R2019a Online Documentation". Accessed on: Dec. 6, 2019. [Online]. Available: <https://www.mathworks.com/help/matlab/ref/ode45.html>
- [15] "Решатели ОДУ" Accessed on: Dec. 6, 2019. [Online]. Available: http://support.ptc.com/help/mathcad/ru/index.html#page/PTC_Mathcad_Help/ode_solvers.html
- [16] "How to | Solve a Differential Equation" 2019. Accessed on: Dec. 6, 2019. [Online]. Available: <https://reference.wolfram.com/language/howto/SolveADifferentialEquation.html>
- [17] "OpenMP" Accessed on: Dec. 6, 2019. [Online]. Available: <https://www.openmp.org>
- [18] "MPICH | High-Performance Portable MPI" Accessed on: Dec. 6, 2019. [Online]. Available: <https://www.mpich.org>
- [19] V. A. Svyatnyy, V. G. Kushnarenko, O. M. Miroshkin, "Parallele Modellierung und Simulation der luftdynamischen Prozesse in den Grubenbewetterungsnetzen", Visti Donetskoho hirnychoho instytutu, no. 1, pp. 116–129, 2018. [Online]. Available: http://science.donntu.edu.ua/wp-content/uploads/2018/08/Svyatnyy_JDMI_1_18.pdf

Надійшла до редакції 07.12.2019 р.

Мірошкін Олександр Миколайович – кандидат технічних наук, доцент, Університет Ульма (алея Альберта Ейнштейна, 11, Ульм, 89081, Німеччина).

E-mail: miroshkinan@gmail.com

Маргієв Георгій Едуардович – аспірант, кафедра комп'ютерної інженерії, Державний вищий навчальний заклад «Донецький національний технічний університет» (пл. Шибанкова, 2, м. Покровськ, 85300, Україна).

E-mail: margievge@gmail.com

BLOCK NUMERICAL METHOD BASED PARALLEL SOLVER FOR DIFFERENTIAL EQUATIONS

Most modern software packages for solving differential equations are based on sequential algorithms that cannot provide efficient parallelization when applied to parallel architectures. One of the main problems in distributed parallel simulation environment development is search for algorithms for a solver that can give a mathematical problem solution with a given accuracy within a specified time, and will use architecturally-relevant principles with respect to existing computing resources. A separate class of problems when using such principles is in load balancing between individual computing nodes while reducing the number of exchange operations between them.

The paper is dedicated to a practical application of a four-point block numerical method for the development of the software differential equations solver, taking into account the features of parallel computing architectures to be used. The proposed solver was programmed in C programming language, the call format of which has been chosen based on analysis of other solvers for differential equation and their systems and adapted to the specific features of the selected programming language. The solver is built using two approaches for computing processes parallelizing in modern computer architectures: multi-platform shared memory multiprocessing programming in C (OpenMP) and parallel library for distributed memory systems (MPI). The resulting solver was successfully tested on different test problems based on differential equation systems of the first and second orders. The testing showed high stability and accuracy during solving systems with up to 10,000 equations. In the article an example of developed solver usage for numerical simulation of mine air network aerodynamical processes, represented by the system of differential equations in partial derivatives, is shown.

Keywords: solver, differential equations, block numerical method, parallelization, architecturally-relevant approach.

Miroshkin Oleksandr Mykolaiovych – Candidate of Technical Sciences, Professor, Associated Professor, Ulm University (Albert Einstein Alley, Ulm, 89081, Germany).

E-mail: miroshkinan@gmail.com

Marhiiev Heorhii Eduardovych – PhD student, Computer Engineering Department, State Higher Education Establishment "Donetsk National Technical University", (Shybankova square, 2, Pokrovsk, Donetsk region, 85300, Ukraine).