

УДК 519.63, 532.5

**Maksym Sorokin**, Researcher

ORCID ID: <https://orcid.org/0000-0002-3165-6040> **e-mail:** maxim.sorockin@gmail.com

Institute of Mathematical Machines and Systems Problems of NAS of Ukraine, Kyiv, Ukraine

## **PARALLELIZATION OF NUMERICAL SOLUTIONS OF SHALLOW WATER EQUATIONS BY THE FINITE VOLUME METHOD FOR IMPLEMENTATION ON MULTIPROCESSOR SYSTEMS AND GRAPHICS PROCESSORS**

***Abstract.** An overview of approaches to parallelization of grid-based numerical methods for solving shallow water equations for multiprocessor systems and graphics processors is presented. A multithreaded approach for shared-memory computing systems implemented on the basis of the OpenMP programming interface and a geometric decomposition approach with message-passing using the MPI library for distributed-memory computers are described. Multithreading for programming GPUs based on the OpenACC software interface is considered. For the COASTOX-UN system of two-dimensional modeling of hydrodynamics, sediment and radionuclide transport in river systems and coastal areas of the seas, the parallelization of its hydrodynamic model COASTOX-HD was carried out. In the developed numerical model, the shallow water equations are solved by finite-volume numerical methods on unstructured computational grids with triangular cells of variable size. The parallelization is implemented using a hybrid MPI+OpenACC approach targeting multiprocessor systems and GPUs. For multiprocessor computers, geometric decomposition and MPI-based messaging are used, and for GPUs, multithreading is implemented using OpenACC directives. The performance of the developed parallel hydrodynamic model was evaluated during the calculation of typical problems of hydrodynamics of shallow water bodies, river flood, and tsunami wave run-up on the coast on a Dell Precision Workstation 7920 multi-core workstation with two 20-core Intel Xeon Gold 6230 processors and NVIDIA Quadro RTX 5000 and NVIDIA GeForce RTX 3080 graphics cards. It is shown that the developed model has significantly accelerated the simulation on the considered multiprocessor system and the considered GPUs. The acceleration on GPUs depends on the size of the computational grid, increasing to saturation with an increase in the number of grid cells. It is established that for the developed parallel model, whose numerical schemes are related to algorithms with low computational intensity, the memory bandwidth of the NVIDIA architecture GPUs is a more important limiting factor of acceleration than their performance.*

***Keywords:** river modeling; coastal modeling; shallow water equations; finite volume; unstructured grid; parallel computing; GPU computing; MPI; OpenACC.*

---

© М.В. Сорокін, 2023

М.В. Сорокін

Інститут проблем математичних машин і систем НАН України, м. Київ, Україна

## РОЗПАРАЛЕЛЮВАННЯ ЧИСЕЛЬНИХ РОЗВ'ЯЗКІВ РІВНЯНЬ МІЛКОЇ ВОДИ МЕТОДОМ СКІНЧЕННИХ ОБ'ЄМІВ ДЛЯ РЕАЛІЗАЦІЇ НА БАГАТОПРОЦЕСОРНИХ СИСТЕМАХ І ГРАФІЧНИХ ПРОЦЕСОРАХ

**Анотація.** Представлено огляд підходів до розпаралелювання сіткових чисельних методів розв'язування рівнянь мілкої води для багатопроцесорних систем та графічних процесорів. Описано багатопотоковий підхід для обчислювальних систем із загальною пам'яттю, реалізований на основі програмного інтерфейсу OpenMP, та підхід геометричної декомпозиції з передачею повідомлень з використанням бібліотеки MPI для комп'ютерів з розподіленою пам'яттю. Розглянуто багатопотоковість для програмування графічних процесорів на основі програмного інтерфейсу OpenACC. Для системи COASTOX-UN двовимірного моделювання гідродинаміки, переносу наносів і радіонуклідів у річкових системах і прибережних районах морів проведено розпаралелювання її гідродинамічної моделі COASTOX-HD. В розробленій чисельній моделі рівняння мілкої води розв'язуються скінченно-об'ємними чисельними методами на неструктурованих обчислювальних сітках з трикутними комірками змінного розміру. Розпаралелення реалізовано за допомогою гібридного підходу MPI+OpenACC, орієнтованого на багатопроцесорні системи і графічні процесори. Для багатопроцесорних комп'ютерів використовується геометрична декомпозиція і обмін повідомленнями на основі MPI, для графічних процесорів – багатопотоковість, реалізована за допомогою директив OpenACC. Представлено тестування продуктивності розробленої паралельної гідродинамічної моделі при обчисленні характерних задач гідродинаміки мілководних водойм, річкової повені і набігання хвилі цунамі на узбережжя, на багатоядерній робочій станції Dell Precision Workstation 7920 з двома 20-ядерними процесорами Intel Xeon Gold 6230 і графічних картах NVIDIA Quadro RTX 5000 і NVIDIA GeForce RTX 3080. Показано, що розроблена модель значно прискорює моделювання задач на розглянутих багатопроцесорній системі і графічних процесорах. Прискорення обчислення задач на графічних процесорах залежить від розміру розрахункової сітки, зростаючи до насичення при збільшенні кількості комірок сітки. Встановлено, що для розробленої паралельної моделі, чисельні схеми якої відносяться до алгоритмів з невисокою обчислювальною інтенсивністю, пропускна здатність пам'яті графічних процесорів архітектури NVIDIA є більш важливим обмежуючим фактором прискорення, чим їхня продуктивність. **Ключові слова:** моделювання річкових систем; моделювання прибережних зон моря; рівняння мілкої води; скінченні об'єми; неструктуровані сітки; паралельні обчислення; обчислення на GPU, MPI, OpenACC.

<https://doi.org/10.32347/2411-4049.2023.2.163-193>

### 1. Вступ

Чисельні моделі, засновані на двовимірних рівняннях мілкої води (РМВ), використовуються для моделювання широкого спектра задач гідродинаміки і переносу забруднень у річках, водосховищах, озерах і прибережних зонах морів, коли характерний горизонтальний розмір збурення набагато більше глибини водойми.

Після аварії на Чорнобильській АЕС, для обґрунтування водоохоронних заходів у Чорнобильській зоні відчуження з урахуванням переносу радіонуклідів ( $^{90}\text{Sr}$  та  $^{137}\text{Cs}$ ) у заплаві р. Прип'ять та водосховищах р. Дніпро в Інституті проблем математичних машин і систем НАН України (ПММС) почалась розробка системи моделей різної розмірності, що включала початкову версію двовимірної моделі COASTOX, заснованої на РМВ [1–3].

На теперішній час COASTOX-UN – це система моделювання гідродинаміки (COASTOX-HD), транспорту наносів (COASTOX-SED) і радіонуклідів (COASTOX-RN) у річкових системах і прибережних районах морів з урахуванням впливу річкового стоку, атмосферного тиску, вітру і вітрових хвиль. Рівняння моделі включають РМВ, двовимірні рівняння переносу наносів та радіонуклідів і рівняння обміну між водною товщею і верхнім шаром дна, що розв'язуються за допомогою чисельних методів скінченних об'ємів на неструктурованих розрахункових сітках з трикутними комірками змінного розміру [4].

COASTOX-HD є аналогом відомих і широко використовуваних систем моделювання, таких як програмний комплекс MIKE 21 розробки датської компанії DHI [5], системи моделювання TELEMAC від консорціуму TELEMAC-MASCARET [6], системи CMS, розробленої Корпусом інженерів Армії США [7]. COASTOX-HD використовувалася у великій кількості проєктів у США, Індонезії, Сінгапурі, Японії та Україні [8–13]. COASTOX-HD, разом COASTOX-SED і COASTOX-RN включено в модуль двовимірного моделювання дисперсії радіонуклідів у водному середовищі для системи підтримки прийняття рішень (СППР) – багатоплатформної системи ядерного аварійного реагування JRODOS [14]. На основі COASTOX-HD розроблено декілька СППР у випадку катастрофічних паводків: системи прогнозування затоплення берегів р. Дніпро у м. Київ [15], системи моделювання повеней на р. Дністер у м. Могилів-Подільський [16], системи передбачення паводків в Українській частині басейнів річок Прут та Сірет [17].

Використання подібних систем моделей для розв'язку комплексних гідродинамічних задач у водоймах великого розміру, з високою деталізацією, на тривалих часових періодах потребує значних обчислювальних ресурсів.

Основним напрямком прискорення розрахунків є використання паралельних методів розв'язку основних рівнянь моделей, орієнтованих на паралельні обчислювальні системи (ПОС): багатопроцесорні комп'ютерні системи із загальною або розподіленою пам'яттю, кластери, графічні процесори (GPU).

Тому разом із розробкою системи моделей COASTOX в ПММС розвивалися методи паралелізації її алгоритмів. Спочатку була розроблена версія COASTOX-UN для багатопроцесорних систем: багатоядерних робочих станцій та кластерів, на основі геометричної декомпозиції розрахункової сітки з використанням технології MPI [18]. Пізніше – версія COASTOX-UN для GPU на основі програмного інтерфейсу OpenACC [19].

У цій статті викладається метод розпаралелювання чисельних схем розв'язку РМВ гідродинамічної моделі COASTOX-HD, з системи моделей COASTOX-UN, за допомогою гібридного підходу MPI+OpenACC. Представлені результати тестування її продуктивності та ефективності при розв'язанні різних задач гідродинаміки мілководних водойм на багатопроцесорних системах і графічних процесорах, використання яких

дозволило на два порядки збільшити швидкість обчислювань, що дає можливість використовувати розрахункові сітки з дуже великою кількістю вузлів на доступних широкому колу користувачів ПОС. Наведені приклади чисельного розв'язання задач, що вимагали великих розрахункових сіток, та проаналізовано вплив технічних характеристик ПОС на продуктивність паралельних обчислень.

## **2. Огляд підходів до розпаралелювання різницевих методів розв'язку рівнянь мілкої води**

Для чисельного розв'язку РМВ найбільш поширені методи скінченних різниць, скінченних об'ємів або скінченних елементів, реалізовані на розрахункових сітках різної структури. Розрахункові сітки можуть бути як простими прямокутними (MIKE 21), так і більш складними: телескопічними декартовими, з прямокутними комітками, що локально згущуються (CMS), або неструктурованими з трикутними елементами змінного розміру (TELEMAC, MIKE-21).

Основним напрямком прискорення розв'язку РМВ на сучасних обчислювальних системах є використання паралельних методів.

Різницеві методи розв'язку рівнянь динаміки мілководних водойм є перспективними для розпаралелювання через наявність у них внутрішнього паралелізму, при якому у початковому алгоритмі легко виділити незалежні один від одного обчислення.

Відповідно до цих методів, неперервні змінні дискретно апроксимуються у просторі (на розрахунковій сітці) та часі. Диференціальні рівняння при цьому представляються алгебраїчними різницевами, для розв'язку яких необхідна обробка дискретизованих змінних і коефіцієнтів рівняння ітераційно за часовими кроками. При цьому на кожному кроці для окремих елементів розрахункової сітки виконуються одні й ті самі операції. Якщо при цьому обчислення ще й локальні, тобто для розрахунку змінної в якомусь елементі необхідні дані тільки з сусідніх елементів, то розпаралелювання природним чином будується на розподілі розрахунку окремих елементів сітки за окремими процесорними пристроями паралельної комп'ютерної системи.

Конкретне формулювання алгоритму та вибір програмного інструменту для його реалізації залежить від архітектури обчислювальної системи. Можна виділити два основних класи ПОС, що відрізняються організацією пам'яті, та відповідні їм технології паралельного програмування різницевих методів.

Перший – це обчислювальні системи із загальною пам'яттю, у яких процесорні пристрої розділяють єдиний простір адресів пам'яті. Для їх програмування використовується парадигма потоків. Спрощено це можна представити так – майстер потік створює деяку кількість додаткових потоків, що відповідає числу задіяних процесорів ПОС, потоки обробляють окремі ітерації в циклі за елементами сітки і переходять до наступних ітерацій, доки не буде виконано розрахунок всіх елементів. Перед наступним циклом за елементами виконується синхронізація потоків.

Другий – системи з розподіленою пам'яттю, у яких кожен процесор разом із локальною пам'яттю утворює вузол ПОС, а окремі вузли об'єднані комунікаційним середовищем. Для програмування таких систем застосовується парадигма паралельних процесів, що обмінюються повідомленнями. У цьому випадку паралельні процеси ініціалізуються

відповідно до числа використовуваних процесорів, розрахункова сітка розбивається на підобласті (геометрична декомпозиція), процесу/процесору ставиться у відповідність підобласть для розрахунку елементів в ній, відсутні дані уздовж границі підобластей процеси отримують в результаті обміну повідомленнями.

Найбільш популярні технології паралельного програмування різницевих методів для ПОС із загальною та розподіленою пам'яттю – це, відповідно, програмний інтерфейс OpenMP та інтерфейс передачі повідомлень MPI.

Технологія OpenMP надає набір директив для виділення ділянок алгоритму, наприклад циклів за елементами, для яких компілятор повинен згенерувати паралельний багатопотоковий код. Вона орієнтована на SMP-системи (Symmetric Multiprocessors), клас комп'ютерів, у яких однорідні процесори мають однаковий доступ до загальної пам'яті. Наприклад, загальнодоступними SMP-системами є персональні комп'ютери, робочі станції, сервери з багатоядерними процесорами Intel та AMD, що можуть містити до кількох десятків ядер – симетричних процесорів. Наразі підходи до розпаралелювання різницевих методів з використанням OpenMP добре розвинені та розроблено вже досить багато OpenMP реалізацій моделей гідродинаміки мілкої води [20–24]. Згадані вище системи моделей CMS та MIKE 21 також розпаралелені за допомогою OpenMP.

Технологія MPI реалізована у вигляді бібліотеки функцій, для традиційних мов програмування. Отже, MPI функції служать для ініціалізації паралельних процесів, передачі повідомлень між ними, їх колективної взаємодії, а алгоритм для кожного процесу написаний традиційною мовою. При цьому для різницевих методів зазвичай використовується SPMD підхід (Single Program, Multiple Data), так що для основної частини обчислень, циклів за елементами своїх підобластей, процеси виконують однаковий алгоритм. Хоча технологія призначена в першу чергу для ПОС з розподіленою пам'яттю, MPI також підтримує деякі типи SMP-систем, наприклад сервери з багатоядерними процесорами. У цьому випадку взаємодія між процесами виконується не через комунікаційне середовище, а через загальну пам'ять. MPI також підтримує такий клас високопродуктивних ПОС, як кластери, у яких сервери з багатоядерними процесорами об'єднані високошвидкісною мережею. Завдяки цьому технологія MPI популярніша для розпаралелювання різницевих методів, ніж OpenMP, а MPI моделі динаміки мілководних водойм набули більш широкого розповсюдження. Це, наприклад, моделі гідродинаміки прибережної зони моря ADCIRC-2DDI [25] та SLOSH [26], прогнозу повеней та гідродинаміки річкових систем RMA2 [27] та TRENT [28]. Зазначимо, що вищеназвана система моделей TELEMAC розпаралелена за допомогою MPI, а система Mike 21 має, крім OpenMP, також MPI версії своїх обчислювальних модулів.

Відносно того, яка технологія розпаралелювання дозволяє швидше моделювати задачі на серверах з багатоядерними процесорами, у літературі є повідомлення про більшу ефективність як OpenMP [22], так і MPI [29] версій алгоритмів моделей. Це обумовлено тим, наскільки кожна технологія та відповідний паралельний алгоритм підходить до конкретного процесора. Тому, для більш гнучкої адаптації моделей динаміки мілкої води до ПОС, насамперед кластерного типу, перспективним є гібридний підхід до паралелізації – MPI+OpenMP, із багатопотоковістю всередині вузла кластера

та MPI комунікацією між вузлами. Так розпаралелена, наприклад, модель гідродинаміки мілкої води TELEMACH-2D [30].

Починаючи з 2000-х років для прискорення розрахунків, у тому числі різницевих методів, почав використовуватися ще один тип обчислювачів. Це графічні процесори (GPU – graphics processing unit): відеокарти та прискорювачі на основі GPU. Сучасні GPU містять тисячі ядер – скалярних процесорів і мають сумарну обчислювальну потужність більшу, ніж багатоядерні процесори. Групи ядер із кешем та загальною пам'яттю утворюють потокові мультипроцесори, що розділяють загальну глобальну пам'ять GPU. Її пропускна здатність вища, ніж шини обміну даними між центральним процесором та загальною пам'яттю. Це додатковий ресурс для прискорення різницевих методів, що відносяться до алгоритмів з невисокою обчислювальною інтенсивністю, для яких число арифметичних операцій на байт переданих даних менше одиниці [31].

GPU можна, в деякому сенсі, вважати ПОС із загальною пам'яттю. Для їх програмування також використовується багатопотоковий підхід. Один із інструментів його реалізації – розширення стандартних мов CUDA C/C++ та CUDA Fortran. Вони є засобами низькорівневого програмування, що дозволяють маніпулювати поточковими мультипроцесорами та різними рівнями пам'яті GPU і досягати високої продуктивності GPU програм. Недоліки CUDA Fortran і C – це те, що вони потребують значних зусиль та часу на розробку, а отримана паралельна програма може бути використана лише на GPU від компанії NVIDIA. Тим не менш, на сьогоднішній день це найуживаніша технологія розпаралелювання алгоритмів для GPU. За її допомогою створені GPU версії моделей гідродинаміки мілкої води SCHE2D [32], TUFLOW [33], Iber [34], TRITON [35].

Останнім часом популярним засобом, що спрощує розробку для GPU, стала модель паралельного програмування OpenACC. Подібно до OpenMP, OpenACC є набором директив компілятора для мов C і Fortran. За допомогою них можна вказати ділянки алгоритму, наприклад, цикли, з незалежними ітераціями, які можуть бути оброблені багатоядерними процесорами в багатопотоковому режимі. OpenACC має низку переваг. Крім GPU від NVIDIA, OpenACC підтримує архітектури багатоядерних процесорів Intel, AMD, POWER та Arm. Дозволяє реалізувати паралелізм, що міститься в алгоритмі, без його істотного перероблення. Допускає поетапне розпаралелювання. Написаний на основі OpenACC програмний код значно коротший за код мовами CUDA. При цьому OpenACC забезпечує, хоча і меншу, ніж CUDA, але все ж таки порівнянну продуктивність різницевих методів на GPU (від 50% до 98% згідно з [36]). Тому все більше моделей динаміки мілководних водойм розпаралелюється за допомогою OpenACC [22, 37, 38].

Для адаптації моделей під декілька графічних процесорів, сервери з кількома GPU або кластери, вузли яких містять GPU, використовується аналогічний MPI+OpenMP гібридний підхід – MPI+CUDA [33, 35] або MPI+OpenACC [39, 40].

Саме останній використано для розпаралелювання чисельних схем розв'язку РМВ гідродинамічної моделі COASTOX-HD.

### 3. Основні рівняння моделі

Гідродинамічні рівняння моделі COASTOX-HD – це РМВ, що утворюються інтегруванням тривимірних рівнянь по глибині шару води з використанням кінематичних граничних умов, які описують адвекцію водних мас за заданим рельєфом під дією гравітації з урахуванням гідростатичного тиску, припливних сил, сили Коріоліса, атмосферного тиску, дії вітру, хвильових напружень, донного тертя і турбулентної в'язкості:

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \mathbf{F} = \mathbf{S} + \nabla \mathbf{F}^d, \quad (1)$$

де  $\mathbf{U}$  – вектор змінних,  $\mathbf{F} = (\mathbf{E}, \mathbf{G})$  – вектор конвективного потоку,  $\mathbf{F}^d = (\mathbf{E}^d, \mathbf{G}^d)$  – вектор дифузійного потоку ( $\mathbf{E}$  і  $\mathbf{G}$  – компоненти в  $x$  та  $y$  напрямках),  $\mathbf{S}$  – вектор вільних членів, що визначаються за формулами:

$$\mathbf{U} = \begin{bmatrix} h \\ q_x \\ q_y \end{bmatrix}, \mathbf{E} = \begin{bmatrix} uh \\ uq_x + \frac{1}{2}gh^2 \\ uq_y \end{bmatrix}, \mathbf{G} = \begin{bmatrix} vh \\ vq_x \\ vq_y + \frac{1}{2}gh^2 \end{bmatrix}, \mathbf{E}^d = \begin{bmatrix} 0 \\ D_x \frac{\partial q_x}{\partial x} \\ D_x \frac{\partial q_y}{\partial x} \end{bmatrix}, \mathbf{G}^d = \begin{bmatrix} 0 \\ D_y \frac{\partial q_x}{\partial y} \\ D_y \frac{\partial q_y}{\partial y} \end{bmatrix},$$

$$\mathbf{S} = \begin{bmatrix} 0 \\ -gh \frac{\partial \eta}{\partial x} + gh \frac{\partial \psi}{\partial x} + \frac{1}{\rho} (\tau_x^\xi + \tau_x^S - \tau_x^\eta) + fq_y - \frac{h}{\rho} \frac{\partial P_a}{\partial x} \\ -gh \frac{\partial \eta}{\partial y} + gh \frac{\partial \psi}{\partial y} + \frac{1}{\rho} (\tau_y^\xi + \tau_y^S - \tau_y^\eta) - fq_x - \frac{h}{\rho} \frac{\partial P_a}{\partial y} \end{bmatrix}. \quad (2)$$

Тут  $\eta(x, y, t)$  – рівень поверхні землі;  $\xi(x, y, t)$  – рівень вільної поверхні води;  $h$  – глибина води ( $= \xi - \eta$ );  $u, v$  –  $x$  та  $y$  компоненти осередненої по глибині швидкості потоку;  $q_x, q_y$  – компоненти витрати води через одиничну ширину;  $g$  – прискорення вільного падіння;  $\rho$  – густина води;  $\partial \eta / \partial x, \partial \eta / \partial y$  – похили річкового дна в  $x$  та  $y$  напрямках;  $\psi$  – приливоутворюючий потенціал;  $\tau_x^\eta, \tau_y^\eta$  – дотичні напруження на дні в  $x$  та  $y$  напрямках;  $\tau_x^\xi, \tau_y^\xi$  – вітрові дотичні напруження на поверхні води;  $\tau_x^S, \tau_y^S$  – хвильові дотичні напруження;  $f$  – параметр Коріоліса;  $P_a$  – атмосферний тиск на вільній поверхні;  $D_x, D_y$  – компоненти коефіцієнта турбулентної в'язкості.

Донне тертя апроксимоване квадратичною залежністю від швидкості течії:

$$\tau_x^\eta / \rho = c_b u U, \tau_y^\eta / \rho = c_b v U, U = \sqrt{u^2 + v^2}, \quad (3)$$

з коефіцієнтом тертя  $c_b$ , що задається формулами Маніга  $c_b = g n^2 / h^3$  ( $n$  – коефіцієнт шорсткості Манінга) або Шезі  $c_b = g / c^2$  ( $c$  – коефіцієнт шорсткості Шезі).

Вітрові дотичні напруження визначаються наступним чином:

$$\tau_x^\xi = C_d \rho_a W^2 \sin \theta, \tau_y^\xi = C_d \rho_a W^2 \cos \theta, \quad (4)$$

де  $C_d$  – коефіцієнт тертя вітру об вільну поверхню води,  $\rho_a$  – густина повітря,  $W$  – швидкість вітру,  $\theta$  – напрямок вітру в декартовій системі координат.

Хвильові дотичні напруження розраховуються за формулами:

$$\frac{\tau_x^\xi}{\rho} = -\frac{1}{\rho} \left( \frac{\partial S_{xx}}{\partial x} + \frac{\partial S_{xy}}{\partial y} \right), \quad \frac{\tau_y^\xi}{\rho} = -\frac{1}{\rho} \left( \frac{\partial S_{xy}}{\partial x} + \frac{\partial S_{yy}}{\partial y} \right), \quad (5)$$

через компоненти хвильових радіаційних напружень,  $S_{xx}$ ,  $S_{xy}$ ,  $S_{yy}$ , що обчислюються об'єднаною із COASTOX-HD хвильовою моделлю.

Параметр Кориоліса дається виразом:

$$f = 2\Omega \sin \varphi, \quad (6)$$

$\Omega$  – кутова частота обертання Землі,  $\varphi$  – широта.

Припливоутворюючий потенціал визначено, як у моделі ADCIRC [25]:

$$\psi(\lambda, \varphi, t) = \sum_{j,n} \varepsilon_{j,n} A_{j,n} f_{j,n}(t_0) L_j(\varphi) \cos[2\pi(t - t_0)/T_{j,n} + j\lambda + v_{j,n}(t_0)], \quad (7)$$

де  $\varepsilon_{j,n}$  – коефіцієнт зменшення амплітуди припливу завдяки пружній реакції Землі на припливи;  $A_{j,n}$  – амплітуда припливної компоненти  $n$ , припливу виду  $j$ ;  $f_{j,n}(t_0)$  – вузловий коефіцієнт, що враховує вплив 18.6-річного припливу на короткостроковий приплив;  $j = 0, 1, 2$  – вид припливу:  $j = 0$  – довгостроковий,  $j = 1$  – добовий,  $j = 2$  – напівдобовий;  $L_0(\varphi) = 3 \sin^2(\varphi) - 1$ ,  $L_1(\varphi) = \sin(2\varphi)$ ,  $L_2(\varphi) = \cos^2(\varphi)$ ;  $\lambda$  – довгота;  $T_{j,n}$  – період припливу;  $v_{j,n}(t_0)$  – астрономічний аргумент;  $t_0$  – час відліку. Необхідний набір припливних компонент залежить від розташування досліджуваної водойми та потрібної точності прогнозу.

#### 4. Чисельні алгоритми моделі

Чисельні схеми рівнянь гідродинамічної моделі COASTOX-HD, отримані на основі методу контрольних об'ємів, відповідно до якого, область моделювання дискретизується на неструктуровану розрахункову сітку із трикутними комірками – контрольними об'ємами (Рис. 1а). Кожен контрольний об'єм оточує точку, розташовану у центрі трикутника, в який визначаються всі гідродинамічні змінні рівнянь, рівень дна задано у вузлах сітки – вершинах трикутників, потоки через границі між комірками задано у серединах ребер трикутників (Рис. 1б). Вісь  $z$  вважається направленою протилежно вектору сили тяжіння.

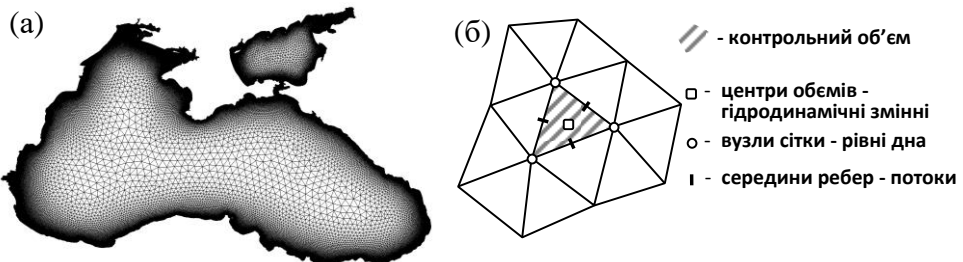


Рис. 1. а) Приклад неструктурованої сітки моделі COASTOX-HD для Чорного і Азовського морів. б) Збільшене зображення контрольного об'єму із зазначенням прив'язки змінних



Дискретизація рівнянь моделі здійснюється шляхом інтегрування їх по контрольних об'ємах. Інтегруючи РМВ (1, 2) по контрольному об'єму  $A$  із границею  $L$ , отримуємо рівняння:

$$\int_A \left( \frac{\partial U}{\partial t} + \nabla \cdot \mathbf{F} - \nabla \cdot \mathbf{F}^d - \mathbf{S} \right) dA = 0. \quad (8)$$

Використовуючи теорему Остроградського, інтеграл по контрольному об'єму від дивергенції конвекційного потоку перетворюємо на інтеграл по його границі:

$$\int_A \left( \frac{\partial U}{\partial t} - \mathbf{S} \right) dA + \oint_L (\mathbf{F} \cdot \mathbf{n} - \mathbf{F}^d \cdot \mathbf{n}) dl = 0. \quad (9)$$

В результаті дискретизації рівняння (9) інтеграли по контрольному об'єму апроксимуються середніми значеннями векторів в об'ємі, а інтеграл по границі апроксимується сумою по його ребрах:

$$\Delta A \frac{\Delta U}{\Delta t} = - \sum_{k=1}^3 \mathbf{F}_k \cdot \mathbf{n}_k \Delta l_k + \sum_{k=1}^3 \mathbf{F}_k^d \cdot \mathbf{n}_k \Delta l_k + \Delta A \mathbf{S}, \quad (10)$$

де  $\Delta t$  – часовий крок,  $\Delta A$  – площа контрольного об'єму,  $\mathbf{n}_k$  – одинична нормаль до ребра  $k$  ( $\mathbf{n} = (n_x, n_y)$ ),  $\Delta l_k$  – довжина ребра  $k$ .

Алгоритм розв'язання рівнянь (10) оснований на схемі SEA [41] та ряді модифікацій, описаних у [42]. Розв'язок відбувається за явним двоетапним методом Рунге-Кутта типу предиктор-коректор із використанням двох різних методів обрахунку потоків на етапах предиктора та коректора, завдяки чому досягається 2-й порядок у часі і просторі.

Для розрахунку потоків використовується підхід, заснований на властивості обертальної інваріантності двовимірних РМВ. Згідно з нею

$$\mathbf{F}(\mathbf{U}) \cdot \mathbf{n} = \mathbf{T}_n^{-1} \cdot \mathbf{E}(\mathbf{T}_n \cdot \mathbf{U}), \quad (11)$$

де  $\mathbf{T}_n$  – матриця повороту векторів до локальної системи координат із віссю  $x$  уздовж вектора нормалі  $\mathbf{n}$  до границі між комірками (Рис. 2),  $\mathbf{T}_n^{-1}$  – матриця повороту до початкової системи координат:

$$\mathbf{T}_n = \begin{pmatrix} 1 & 0 & 0 \\ 0 & n_x & n_y \\ 0 & -n_y & n_x \end{pmatrix}, \quad \mathbf{T}_n^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & n_x & -n_y \\ 0 & n_y & n_x \end{pmatrix}. \quad (12)$$

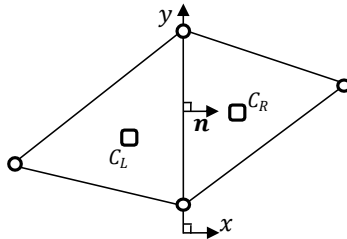


Рис. 2. Локальна система координат, заснована на границі між комірками

Таким чином, конвективний потік може бути обчислено в локальній системі координат, із використанням тільки компоненти  $\mathbf{E}$ , а потім відтворено в початковій системі координат за допомогою перетворення (12).

Для розрахунку компоненти  $\mathbf{E}^p$  на етапі предиктора використовується схема обчислення потоку Годуновського типу із наближеними методами розв'язання задачі Рімана: HLLC [43] чи Роу [44].

На етапі коректора потоки рахуються безпосередньо, вниз за потоком, використовуючи стани задачі Рімана:

$$\mathbf{E}^c(\mathbf{T}_n \cdot \tilde{\mathbf{U}}) = \begin{cases} \mathbf{E}(\tilde{\mathbf{U}}_R), & \text{if } \frac{1}{2}(\tilde{u}_L + \tilde{u}_R) \geq 0 \\ \mathbf{E}(\tilde{\mathbf{U}}_L), & \text{if } \frac{1}{2}(\tilde{u}_L + \tilde{u}_R) < 0 \end{cases} \quad (13)$$

де  $\tilde{\mathbf{U}}_L, \tilde{\mathbf{U}}_R$  – вектори змінних на лівій і правій сторонах границі, прогнозовані на етапі предиктора, перетворені до локальної системи координат;  $\tilde{u}_L, \tilde{u}_R$  – прогнозовані швидкості потоку нормальні до границі (позитивний напрямок від лівої комірки до правої).

Для моделювання течій із різкою зміною параметрів, загальний конвективний потік через ребро комірки на етапі коректора формулюється із використанням TVD обмежувача потоку:

$$\mathbf{F}^{TVD} = \mathbf{F}^p + \phi(r) \cdot \delta \mathbf{F}, \quad \delta \mathbf{F} = \frac{1}{2}(\mathbf{F}^c - \mathbf{F}^p), \quad (14)$$

де  $\mathbf{F}^p$  – потік предиктора,  $\mathbf{F}^c$  – потік коректора,  $\phi(r)$  – обмежувач потоку (minmod, superbee або Van-Leer),  $r$  – відношення варіації ввєрх за потоком до локальної варіації потоку, яке обчислюється окремо для кожної компоненти вектора потоку або, що теж саме, для кожного з трьох рівнянь РМВ.

Для трикутної комірки відношення  $r$  для кожної компоненти вектора потоку визначається наступним чином:

$$r = \begin{cases} -(\delta F \cdot \Delta l)^{-1} \cdot \sum_{i=1}^2 \delta F_i \cdot \Delta l_i, & \text{if } F_h^p \geq 0 \\ (\delta F \cdot \Delta l)^{-1} \cdot \sum_{j=1}^2 \delta F_j \cdot \Delta l_j, & \text{if } F_h^p < 0 \end{cases} \quad (15)$$

де сума по  $i$  – сума по двох інших ребрах лівої комірки, сума по  $j$  – сума по двох інших ребрах правої комірки. Потік позитивний, коли направлений зліва направо, згідно з Рис. 2.  $F_h^p$  – перша компонента прогнозованого потоку  $\mathbf{F}^p$ .

Внесок дифузії у зміну моментів враховується на етапі предиктора. Для дифузійних потоків через границі контрольного об'єму у (10) використовуються центральні різниці.

Вектор вільних членів  $\mathbf{S}$  в рівняннях (1, 2) враховується наступним чином. Більшість його складових додається до вектора змінних після предиктора-коректора потоків у вигляді добутку  $\Delta t \mathbf{S}$ , окрім двох доданків: частини, обумовленої похилом дна  $\mathbf{S}_b$ , і донного тертя  $\mathbf{S}_f$ .

Для частини, обумовленої похилом дна, для узгодженості, також використовується двоетапний метод Рунге-Кутта

$$\mathbf{S}_b = \frac{1}{2}(\mathbf{S}_b^p + \mathbf{S}_b^c) = \mathbf{S}_b^p + \frac{1}{2}(\mathbf{S}_b^c - \mathbf{S}_b^p), \quad (16)$$

де  $S_b^p$  обчислюється за значеннями змінних, отриманих після адвекції на етапі предиктора,  $S_b^c$  – за значеннями змінних, прогнозованими на етапі коректора.

Для врахування донного тертя в (1, 2), з урахуванням квадратичної апроксимації (3), використовується напівнеявна схема, за якою моменти  $q_x, q_y$  розраховуються з рівнянь:

$$\hat{q}_x - q_x = -c_b \frac{\Delta t}{h^2} \hat{q}_x \sqrt{q_x^2 + q_y^2}, \quad \hat{q}_y - q_y = -c_b \frac{\Delta t}{h^2} \hat{q}_y \sqrt{q_x^2 + q_y^2}, \quad (17)$$

де шляпка позначає величину на наступному часовому кроці.

На обох етапах чисельної схеми може виникнути ситуація перевитрати, коли з комірки витікає більше води, ніж в ній є. В схемі використовується двоетапна корекційна процедура, за якою, спочатку, для кожної комірки обчислюється корекційний коефіцієнт  $R_{out}$  на основі порівняння загального потоку води, що залишає комірку через усі сторони  $F_{out}$ , і максимально можливого потоку  $F_{out,max}$ :

$$R_{out} = \begin{cases} F_{out,max}/F_{out}, & \text{if } F_{out} > F_{out,max} \\ 1, & \text{if } F_{out} \leq F_{out,max} \end{cases}, \quad (18)$$

$$F_{out} = \sum_{k=1}^3 \max(0, F_{h,k} \cdot \Delta l_k), \quad F_{out,max} = h\Delta A/\Delta t.$$

На другому етапі для кожного ребра розраховуються корекційний коефіцієнт  $C$  із використанням корекційних коефіцієнтів сусідніх комірок ( $R_{out}^L, R_{out}^R$ ) і відкоригований потік маси  $F_h^*$ :

$$F_h^* = CF_h, \quad C = \begin{cases} R_{out}^L, & \text{if } F_h \geq 0 \\ R_{out}^R, & \text{if } F_h < 0 \end{cases}. \quad (19)$$

Оскільки чисельна схема РМВ явна, її стабільність обумовлюється вибором часового кроку так, щоб виконувалася умова Куранта-Фрідріхса-Леві (CFL):

$$\Delta t = Cr \cdot \min_i \left[ \frac{2\Delta A_i}{\Delta l_{max,i} \cdot \left( \sqrt{u_i^2 + v_i^2} + \sqrt{gh_i} \right)} \right], \quad (20)$$

де мінімум знаходиться по всіх комірках, а  $\Delta l_{max,i}$  – найдовше ребро окремої комірки. Для стійкості двовимірних схем другого порядку у часі, число Куранта  $Cr$  повинно лежати в інтервалі  $0 < Cr \leq 0.5$ . Для більшості практичних застосувань у моделі використовується значення  $Cr = 0.4$ .

Граничні умови застосовуються на обох кроках чисельної схеми розв'язання РМВ, як на етапі предиктора значень змінних, так і на етапі коректора. Гідродинамічні змінні на границях розрахункової сітки або задаються, використовуючи встановлені значення, або апроксимуються, чи зі значень змінних у прилеглих до границь комірках, чи зі значень потоків на ребрах граничних комірок, чи зі значень інваріантів Рімана на границях, що дозволяє хвилям, збудженим всередині, вільно залишати область моделювання.

Для границь із умовами, не визначеними у вхідних файлах гідродинамічної моделі COASTOX-HD, граничні умови задаються автоматично. Для РМВ задаються граничні умови із нульовою компонентою нормальної до границі швидкості.

Описані чисельні схеми реалізовано в програмному коді моделі за допомогою мови програмування FORTRAN.

## 5. Паралелізація алгоритмів моделі

### 5.1. Паралельний алгоритм для багатопроцесорних комп'ютерів на основі MPI

Як показано у попередньому розділі, чисельні схеми розв'язання РМВ гідродинамічної моделі COASTOX-HD явні, скінченно-об'ємні, що мають властивість локальності. Тобто, значення змінної на новому часовому кроці в кожному елементі розрахункової сітки обчислюється на основі значень змінних на попередньому часовому кроці з сусідніх до даного елементів.

Тому для паралелізації чисельних алгоритмів моделі для багатопроцесорних комп'ютерних систем добре підходить метод на основі геометричної декомпозиції розрахункової області та обміну повідомленнями на основі технології MPI.

Після декомпозиції початкова розрахункова область розділяється на підобласті відповідно до числа процесорів ПОС. Далі кожен процесор обчислює гідродинамічні змінні тільки в елементах своєї підобласті. Для розрахунку значень в елементах, що лежать уздовж границь підобласті, необхідні значення, що обчислюються в граничних елементах інших підобластей іншими процесорами. Для цього уздовж границь вводяться halo-структури – фіктивні елементи сітки, комірки/вузли/ребра, що відповідають дійсним граничним елементам інших процесорів. Перед обчисленнями на новому часовому кроці значення змінних в halo-елементах оновлюються значеннями, розрахованими у сусідніх підобластях. Обмін значеннями реалізується за допомогою бібліотеки передачі повідомлень MPI, яка дозволяє обмінюватися даними між процесорами або через мережеве з'єднання на ПОС із розподіленою пам'яттю, або через пам'ять на ПОС із загальною пам'яттю.

Геометрична декомпозиція розрахункової сітки у COASTOX-HD реалізована на основі бібліотеки розділу графів METIS [45]. Вона просто інтегрується в FORTRAN код моделі. На основі масиву, що задає для кожної комірки сітки індекси її сусідів, генерується дуальний граф, який розділяється за допомогою методу багаторівневого k-полосного розбиття, реалізованого у функції METIS\_PartGraphKway(...). На Рис. 3 продемонстровано приклад декомпозиції розрахункової сітки моделі Чорного і Азовського морів, що була показана на Рис. 1. METIS також забезпечує балансування навантаження між процесорами ПОС, за рахунок розділення сітки на частини, що складаються з приблизно однакової кількості комірок.

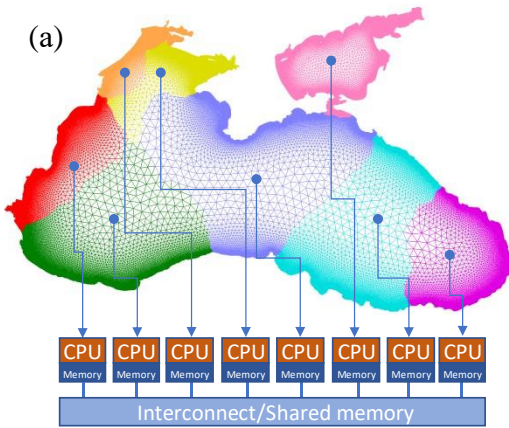


Рис. 3. Приклад розділення за допомогою бібліотеки METIS розрахункової сітки моделі Чорного і Азовського морів на 8 підобластей

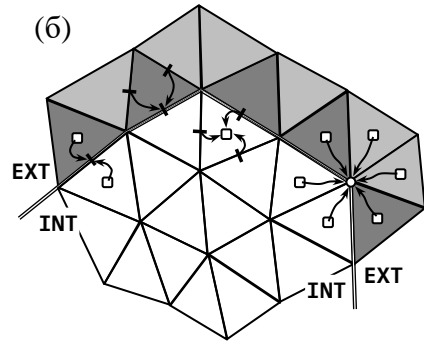


Рис. 4. Розрахункова сітка на границі між підобластями. Стрілками позначено інформаційні залежності для основних шаблонів чисельної схеми. Темно-сірим та світло-сірим кольором виділені фіктивні елементи для нижньої підобласті

Структура halo-елементів визначається шаблонами чисельних схем розрахунку гідродинамічних змінних. Інформаційні залежності шаблонів представлено на Рис. 4. Там же показані відповідні цим залежностям halo-структури.

1) Для розрахунку потоку крізь границю між комірками використовуються значення гідродинамічних змінних, заданих у центрах цих комірок. Отже, для розрахунку потоку у ребрах комірок, що лежать на міжпроцесорних границях, вводяться фіктивні комірки, які безпосередньо до них прилягають. Вони позначені темно-сірим кольором на Рис. 4.

2) Для корекції потоку крізь будь-яке ребро використовуються значення потоків з двох інших ребер комірки, що примикає до цього ребра (зліва або справа, в залежності від напрямку потоку). Відтак, для корекції потоку у ребрах комірок, що лежать на міжпроцесорних границях, необхідні фіктивні ребра – ребра фіктивних комірок, позначених темно-сірим кольором на Рис. 4.

3) Для обчислення гідродинамічних змінних у центрах комірок необхідні значення потоків крізь ребра комірки та значення цієї змінної із попереднього етапу. Тому, для цього шаблону введення фіктивних елементів сітки непотрібно.

4) Значення змінних у вузлах сітки отримуються осередненням по всіх комірках, що оточують даний вузол. Для осереднення у вузлах, що лежать на міжпроцесорних границях, необхідні фіктивні комірки, що оточують ці вузли з зовнішньої сторони. Вони повинні відповідати дійсним коміркам інших підобластей. Оскільки вже введені фіктивні комірки для розрахунку потоків, додатково вводяться фіктивні комірки, що їх доповнюють для осереднення у граничних вузлах. Вони позначені світло-сірим кольором на Рис. 4.

Завдяки локальному характеру чисельної схеми моделі необхідний тільки один ряд фіктивних елементів уздовж міжпроцесорних границь. Відповідно об'єм даних, яким обмінюються процесори, дорівнює, із точністю до множника, кількості елементів у одному halo-ряду підобласті.

Програмно обміни даними виконуються за допомогою викликів у FORTRAN коді моделі неблокованих функцій відправки та отримання повідомлень `MPI_ISEND(...)`, `MPI_IRECV(...)` сумісно із функцією завершення викликів `MPI_WAITALL(...)`. Після розділення сітки для кожного процесора визначаються сусідні процесори, а також – дійсні елементи сітки, значення з яких будуть до них відправлятися, і фіктивні елементи сітки, значення в яких будуть заповнюватися отриманими від них даними. Відповідність відправлення і отримання значень встановлюється за допомогою похідних MPI типів даних для комірок/вузлів/ребер, на основі масивів їх індексів, за допомогою функцій `MPI_TYPE_CREATE_INDEXED_BLOCK(...)` і `MPI_TYPE_COMMIT(...)`.

Для операцій на всій розрахунковій сітці, як-то пошук максимуму або мінімуму по всіх елементах розрахункової сітки, знаходження суми, використовується глобальна редукція. Кожен процесор виконує відповідну операцію на своїй підобласті, а потім значення для всієї сітки отримується зі значень для підобластей за допомогою функції глобальної редукції `MPI_ALLREDUCE(...)`.

Вивід даних здійснюється одним master-процесором. Для цього всі інші процесори відсилають йому значення у всіх елементах своєї підобласті, що реалізовано за допомогою функцій `MPI_ISEND(...)`, `MPI_IRECV(...)`, сумісно із `MPI_WAITALL(...)`, і похідних типів даних.

Загалом схема роботи паралельного алгоритму COASTOX-HD для багатопроцесорних комп'ютерів із розподіленою пам'яттю наступна:

- На початку ініціалізується MPI і створюються паралельні процеси.
- Далі всі створені паралельні процеси читають вхідні параметри, граничні умови, початкові умови та рельєф з вхідних файлів.
- Для всієї розрахункової сітки ініціалізуються глобальні змінні та структури, що задають сусідів для кожного елемента сітки.
- Відбувається декомпозиція сітки на підобласті, визначаються сусідні процесори, задаються halo-структури за допомогою похідних типів даних MPI.
- Ініціалізуються гідродинамічні змінні по підобластях.
- Стартує ітераційний процес за часом:
  - Розраховується часовий крок за умовою CFL. Для пошуку мінімального кроку по всіх підобластях використовується глобальна операція редукції `MPI_MIN`.
  - Розв'язуються РМВ:
    - Етап предиктора:
      - Розраховуються граничні гідродинамічні величини.
      - Розраховуються адвективні потоки.
      - Виконується корекція перевитрати.
      - Розраховується зміна величин за рахунок адвекції.
      - Враховуються вільні члени обумовлені похилом дна.
    - Оновлюються гідродинамічні змінні в halo-комірках.

- Етап коректора:
  - Розраховуються граничні гідродинамічні величини.
  - Розраховуються адвективні потоки.
  - Оновлюються адвективні потоки у halo-ребрах.
  - Виконується TVD корекція потоків.
  - Виконується корекція перевитрати.
  - Розраховується зміна величин за рахунок адвекції.
  - Враховуються вільні члени, обумовлені похилом дна.
- Враховується донне тертя.
- Оновлюються гідродинамічні змінні в halo-комірках.
- Якщо час дорівнює часу виведення даних:
  - Виконується збір даних зі всіх підобластей на master-процесі.
  - Master-процес записує дані у вихідні файли.
- Перевіряється чи поточний час менше часу моделювання, якщо менше – відбувається перехід до наступної часової ітерації, якщо більше – вихід з ітераційного процесу.
- Фіналізується MPI, знищуються паралельні процеси, завершується робота програми.

## ***5.2. Паралельний алгоритм для графічних процесорів на основі OpenACC***

Чисельні схеми моделі COASTOX-HD реалізовано в програмному FORTRAN коді моделі у формі циклів по елементах сітки – вузлах, комірках, ребрах комірок. Ітерації більшості циклів незалежні завдяки тому, що схеми явні, а обчислення локальні. Розрахунок значень у будь-якому елементі сітки не залежить від значень у інших елементах на поточному часовому кроці. Він залежить від значень з попереднього часового кроку у елементах сітки, що розташовані в деякому околі даного.

Відповідно, в алгоритмі COASTOX-HD багатопотокове розпаралелювання для GPU може бути реалізовано без суттєвих змін його структури. Кожному елементу сітки, чи групі елементів, ставиться у відповідність паралельний потік, що може оброблятися незалежно від інших потоків. Потоки обробляються ядрами GPU паралельно, доки, порція за порцією, не будуть опрацьовані всі елементи сітки.

Багатопотоковість реалізується в COASTOX-HD засобами OpenACC. Директиви OpenACC вставляються у FORTRAN код моделі, показуючи компілятору цикли по елементах сітки, для яких необхідно генерувати багатопотоковий код, так щоб ітерації циклів оброблялися паралельно ядрами GPU. Всередині циклів синхронізувати потоки не потрібно. Потоки синхронізуються по завершенню циклів.

Окрім цього, необхідно явно контролювати завантаження даних від центрального процесора на GPU та вивантаження даних з GPU до центрального процесора для їх подальшої обробки.

Розпаралелювання чисельних схем розв'язку РМВ реалізовано в COASTOX-HD за допомогою OpenACC наступним чином. Псевдокод, що відображає структуру FORTRAN коду моделі COASTOX-HD із директивами OpenACC, наведено на Рис. 5.

```

Program COASTOX-UN
!$ACC DATA COPYIN(V,FV,...)
Do while(Time<TimeMax)
  call ComputeTimeStepCFL(DT)
  Time=Time+DT
  call ComputeFluxFV()
  call ComputeVariableV()
  If(Time==OutputTime) Then
!$ACC UPDATE HOST(V,...)
    call Output(V,...)
  Endif
Enddo !Time
!$ACC END DATA
End program

Subroutine ComputeTimeStepCFL(DT)
!$ACC KERNELS PRESENT(...)
!$ACC LOOP REDUCTION(min:DT)
  do ic=1,NC
    ...
    DT = min(DT,...)
  enddo
!$ACC END KERNELS
End subroutine

Subroutine ComputeFluxFV()
!$ACC DATA PRESENT(EC,V,FV,...)
!$ACC KERNELS
!$ACC LOOP INDEPENDENT PRIVATE(C,icL,icR,...)
  do ie=1,NE
    icL = EC(1,ie)
    icR = EC(2,ie)
    call ComputeCoefficientC(...)
    FV(ie) = C*(V(icL) - V(icR)) ...
  enddo
!$ACC END KERNELS
!$ACC END DATA
End subroutine

Subroutine ComputeVariableV()
!$ACC DATA PRESENT(CE,A,L,V,FV,...)
!$ACC KERNELS
!$ACC LOOP INDEPENDENT PRIVATE(k,ie,tV,...)
  do ic=1,NC
!$ACC LOOP SEQ
    do k=1,3
      ie = CE(k,ic)
      tV = tV + FV(ie)*L(ie) ...
    enddo
    V(ic) = V(ic) - DT*tV/A(ic) ...
  enddo
!$ACC END KERNELS
!$ACC END DATA
End subroutine

Subroutine ComputeCoefficientC(...)
!$ACC ROUTINE SEQ
C = ...
End subroutine

```

Рис. 5. Псевдокод, що відображає структуру коду моделі COASTOX-HD із директивами OpenACC. NE – кількість ребер (edge) у розрахунковій сітці; NC – кількість комірок (cell); EC(2,NE) – масив, що задає для кожного ребра індекси сусідніх комірок; CE(3,NC) – масив із індексами ребер для кожної комірки; L(NE) – масив із довжинами ребер; A(NC) – масив із площами комірок

На самому початку центральний процесор ініціалізує масиви гідродинамічних змінних, потоків та допоміжних величин на розрахунковій сітці.

Перед стартом ітераційного циклу за часом ініціалізовані масиви завантажуються в пам'ять графічного процесора за допомогою директив !\$ACC DATA COPYIN(...).

Багатопотокова обробка циклів по комірках/вузлах/ребрах сітки визначається директивами:

```

!$ACC DATA PRESENT(...),
!$ACC KERNELS,
!$ACC LOOP INDEPENT PRIVATE(...),
!$ACC END KERNELS,
!$ACC END DATA .

```

Директива !\$ACC KERNELS визначає початок паралельної частини коду.

Директива !\$ACC LOOP INDEPENT PRIVATE(...) вказує цикл, що буде оброблятися у багатопотоковому режимі. При цьому, уточнення PRIVATE визначає локальні змінні для всіх потоків.

Директива !\$ACC END KERNELS визначає завершення паралельної частини коду та синхронізацію потоків.



Для вказівки масивів, вже завантажених в пам'ять графічного процесора, перед циклом вставляється директива `!$ACC DATA PRESENT(...)`. Директива `!$ACC END DATA` визначає завершення частини коду, для якої актуальні ці масиви.

Якщо в окремій ітерації паралельного циклу є короткий цикл, наприклад в циклі по комітках – внутрішній цикл по трьох граничних ребрах, він не розпаралелюється по потоках, а виконується окремим потоком. Це уточнюється за допомогою директиви `!$ACC LOOP SEQ` перед коротким циклом.

Для багатопотокової операції пошуку мінімального часового кроку за умовою Куранта-Фрідріхса-Леві використовується директива паралельного циклу із редукцією – `!$ACC LOOP REDUCTION(min:...)`.

Якщо у циклі присутній виклик функції, ця функція повинна бути оголошена такою, що виконується у паралельному режимі окремим потоком, за допомогою директиви `!$ACC ROUTINE SEQ`.

Перш ніж виводити змінні у зовнішні файли, масиви змінних копіюються з GPU до центрального процесора за допомогою директив `!$ACC UPDATE HOST(...)`.

## 6. Результати тестування

В ідеальних умовах, за відсутності накладних витрат на доставку операндів до процесорних пристроїв, розроблений паралельний алгоритм повинен прискорюватися лінійно, пропорційно кількості використовуваних процесорів. Точніше, прискорення повинне зростати за законом Амдала [46], тому що деяка кількість арифметичних операцій алгоритму залишається нерозпаралеленою.

В реальних ПОС прискорення залежить від характеру взаємодії між пам'яттю системи і процесорами. В багатоядерних комп'ютерах час витрачається на завантаження даних з загальної пам'яті до обчислювальних ядер. В ПОС із розподіленою пам'яттю об'єм даних, завантажених з пам'яті до процесора, на вузлі зменшується (завдяки чому можливе суперлінійне прискорення), але додаються витрати на обмін даними між вузлами. В GPU час витрачається на загрузку операндів з пам'яті до мультипроцесорів, але пропускна здатність пам'яті вища, ніж у багатоядерних комп'ютерів, що є додатковим фактором прискорення.

Для тестування прискорення моделі COASTOX-HD на ПОС різної архітектури її точності і ефективності було обрано декілька типових задач динаміки мілководних водойм.

Тестування проводилось на наступному обладнанні:

- Двопроцесорна робоча станція Dell Precision Workstation 7920 з 20-ядерними процесорами Intel Xeon Gold 6230, із тактовою частотою 2.1 ГГц, і оперативною пам'яттю 96 ГБ. Операційна система станції – Windows 10 Pro for Workstations. Для компіляції та збірки виконуваного файлу COASTOX-HD використовувався компілятор Intel Fortran Compiler 19.0.4.245 разом із бібліотекою Intel MPI Library 4.0.

- Графічна плата NVIDIA Quadro RTX 5000, архітектури Turing, що має 3072 ядра із частотою 1.815 GHz та 16 GB оперативної пам'яті. Пікова продуктивність – 11.15 TFLOPS на одинарній точності і 348.5 GFLOPS на подвійній точності, пропускна здатність пам'яті – 448.0 GB/s. Плата

встановлена на робочій станції Dell. Для збірки виконуваного файлу для GPU використовувався компілятор PGI Fortran Compiler 19.10 Community Edition, що підтримує OpenACC директиви під Windows.

- Графічна карта NVIDIA GeForce RTX 3080, архітектури Ampere, із 8704 ядрами частотою 1.71 GHz та 10 GB оперативної пам'яті. Пікова продуктивність – 29.77 TFLOPS на одинарній точності і 465.1 GFLOPS на подвійній точності, пропускна здатність пам'яті – 760.3 GB/s. Карта встановлена на ПК із операційною системою Linux Ubuntu 20.04 LTS. Для збірки виконуваного файлу для GPU під Linux використовувався підтримуючий OpenACC компілятор NVIDIA Fortran 22.1-0.

### 6.1. Цунамі

*Лабораторний експеримент з нахату цунамі на складний пляж. Долина Монай.*

Точність моделі COASTOX-HD у прогнозуванні цунамі перевірялась на лабораторному тесті [47], що відтворює умови цунамі 1993 року, яке вразило острів Окусірі і супроводжувалось екстремальною висотою заплеску 30 м у бухті долини Монай.

Лабораторне налаштування відповідає реальній батиметрії у прямокутній області розміром 5.448 м на 3.402 м (Рис. 6б). Вхідна хвиля з глибини 13.5 см являє собою N-подібну хвилю із ведучим поглибленням -2.5 мм і гребнем 1.6 см позаду нього (Рис. 6а). Висота хвилі вимірювалась протягом 200 с у трьох точках з координатами  $(x, y) = (4.521, 1.196)$ ,  $(4.521, 1.696)$ , і  $(4.521, 2.196)$  м (Рис. 6б).

Для моделювання була побудована розрахункова сітка, що складається з 12 989 вузлів і 25 553 комірок розміром від 20 см на глибині до 2.5-3 см на узбережжі (Рис. 6б). Донне тертя задавалось за формулою Шезі із коефіцієнтом  $180 \text{ м}^{1/2}/\text{с}$ .

Моделювались перші 60 с лабораторного експерименту. Порівняння розрахованої висоти хвилі цунамі і вимірів у точках 1-3 наведено на Рис. 7.

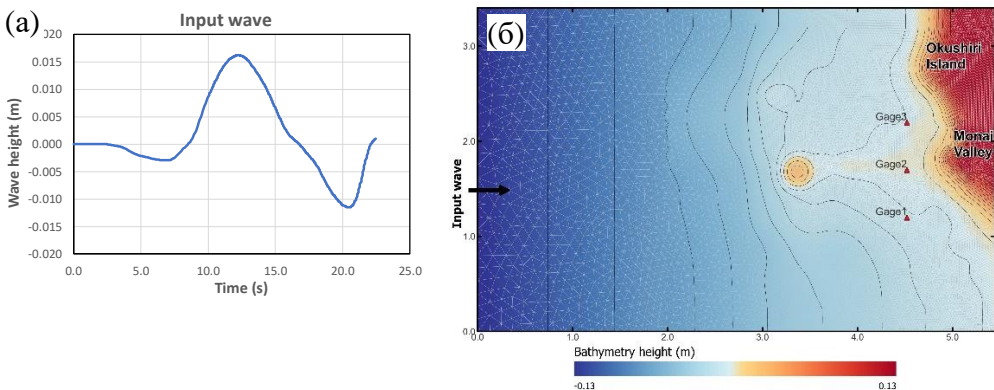


Рис. 6. а) Профіль вхідної хвилі експерименту з цунамі в долині Монай. б) Розрахункова область експерименту

Видно, що модель трохи завищує висоту і швидкість відбитої хвилі, але добре відтворює хвилю, що підходить до берега. Тому COASTOX-HD може використовуватися в задачах нахату цунамі на берег.

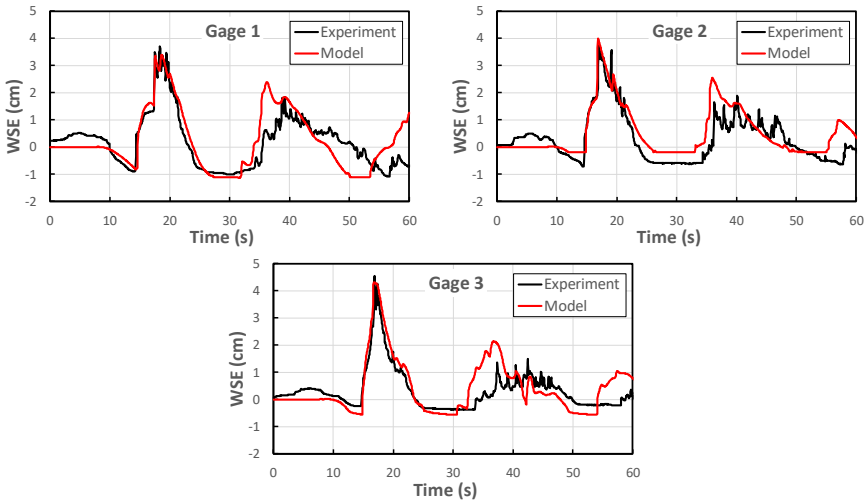


Рис. 7. Порівняння висоти хвилі, виміряної хвилемірами 1-3 під час експерименту з цунамі в долині Монай, і розрахованої моделлю в точках розташування хвилемірів

*Тохокське цунамі 2011 року.*

Для тестування паралельної продуктивності COASTOX-HD була обрана задача накату цунамі, викликаного Великим східно-японським землетрусом 11 березня 2011 року, на узбережжя регіону Тохоку.

Досліджувана область обмежена затокою Сендай, узбережжям префектур Міягі і Фукусіма (Рис. 8а). Для глибин затоки використовувались вільні батиметричні дані ETOPO1 роздільною здатністю 1 кутова хвилина [48], для відтворення рельєфу узбережжя – глобальні топографічні дані SRTM роздільною здатністю 3 кутових секунди (приблизно 90 м) [49].

На основі об'єднаного рельєфу було побудовано розрахункову сітку, що складається з 980 190 вузлів та 1 960 133 комірок. Розмір комірок змінюється від 90 м в прибережній зоні до 1850 м на відкритій морській границі. Донне тертя задавалось однорідним по всій області за формулою Манінга з коефіцієнтом  $0.025 \text{ c/m}^{1/3}$ .

У якості джерела цунамі розглядалась модель Мухарі та ін. [50]. Розподіл поля деформації донної поверхні, отриманий в [50], використовувався для миттєвої передачі деформації на поверхню води для її збурення у початковій фазі розповсюдження цунамі. Збурення трансформувалось у хвильову умову на морській границі побудованої розрахункової сітки.

Моделювались 100 хвилин динаміки хвилі цунамі від часу підходу хвилі до границі області досліджування. Результати моделювання представлені на Рис. 8б, 8в.

З порівняння висоти хвилі, виміряної на станції спостереження за припливами Сома Метеорологічного управління Японії (37°50' широти, 140°58' довготи), під час Тохокського цунамі [51], і зміни вільної поверхні води, розрахованої моделлю в місці розташування станції (Рис. 8б), випливає, що модель вірно прогнозує амплітуду хвилі цунамі і час її приходу. Порівняння промодельованої зони затоплення і зони затоплення, побудованої по результатах польових досліджень Товариством геоінформатики Японії [52] (Рис. 8в), показує, що модель також адекватно відтворює накат цунамі на узбережжя префектур Міягі і Фукусіма.

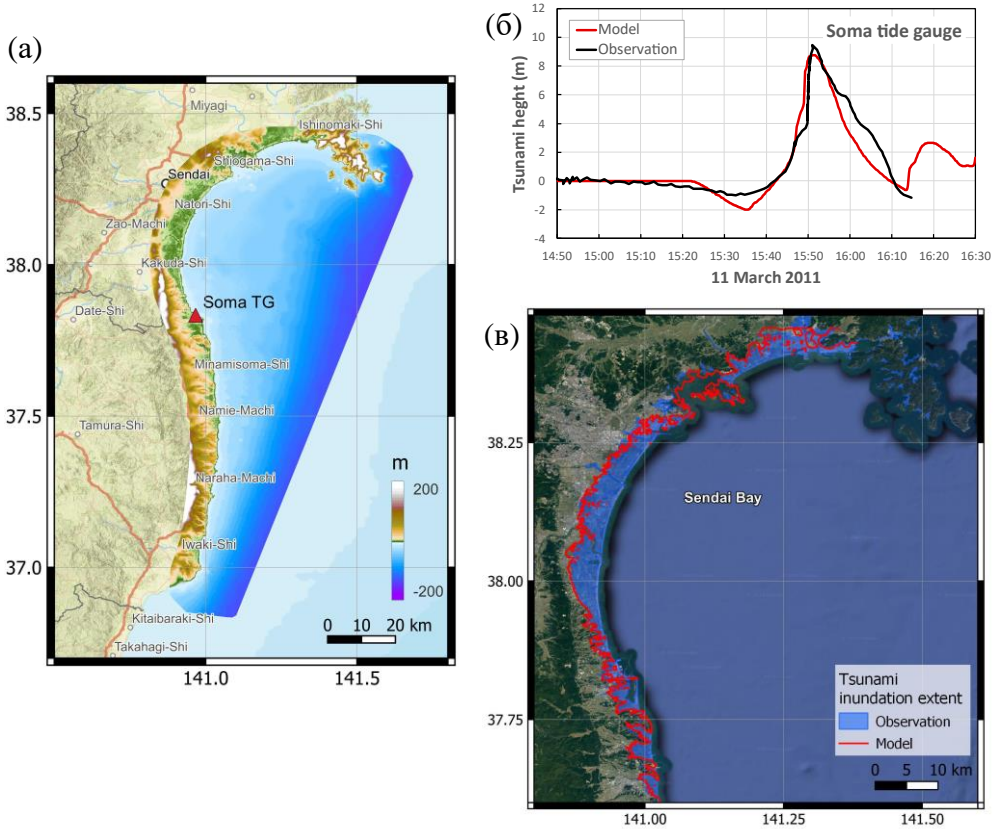


Рис. 8. а) Рельєф розрахункової області моделювання нахату Тохокського цунамі 2011 р. на узбережжя префектур Міягі і Фукусіма із зазначенням станції спостереження за припливами Сомы. б) Порівняння висоти хвилі, виміряної на станції Сомы і розрахованої моделлю в місці її розташування. в) Порівняння промодельованої зони затоплення і зони затоплення, побудованої по результатах польових досліджень

Для тестування прискорення паралельної моделі розрахунки проводились на багатоядерних процесорах робочої станції Dell і графічній карті Quadro RTX 5000. Час моделювання цунамі на робочій станції Dell багатопроекторною MPI версією COASTOX-HD дорівнює 6.54 хвилини, що у 20.44 разів швидше, ніж однопроцесорна версія із часом моделювання 133.6 хвилини. Розрахунок на Quadro RTX 5000 OpenACC версією займає лише 2.91 хвилини у подвійній точності і 1.47 хвилини у одинарній точності, що швидше однопроцесорної версії у 45.86/90.79 разів та швидше багатопроекторної у 2.24/4.44 разів, відповідно (Табл. 1, Рис. 9).

Таблиця 1 – Час розрахунку та прискорення різних версій COASTOX-HD при моделюванні 100 хвилини нахату Тохокського цунамі 2011 р. на узбережжя префектур Міягі і Фукусіма

Model	Facilities	# of processors	Time, min	Speedup vs Serial	Speedup vs MC
Serial	Dell Workstation	1	133.6	1	-
Multi-core MPI	7920	40	6.54	20.44	1
GPU OpenACC DP	Quadro RTX	1x3072	2.91	45.86	2.24
GPU OpenACC SP	5000		1.47	90.79	4.44

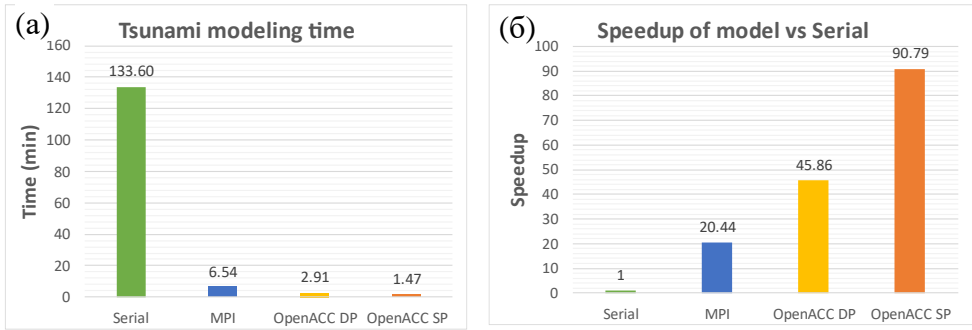


Рис. 9. Порівняння часів розрахунку (а) та прискорення (б) різних версій COASTOX-HD при моделюванні 100 хвилин нахату Тохокського цунамі 2011 р.

Залежність прискорення паралельної версії моделі від розміру розрахункової сітки досліджено на прикладі задач з моделювання повеней.

### 6.2. Повінь

Завдяки неструктурованій розрахунковій сітці з комірками змінного розміру і стійкій чисельній схемі Годуновського типу модель COASTOX-HD здатна відтворювати характерні особливості течії під час раптових повеней, викликаних природними причинами або руйнуванням дамб чи гребель. Зокрема, розповсюдження фронту зволоження-осушення на складному рельєфі, ударні хвилі, взаємодію течії з перешкодами.

Для дослідження впливу розміру розрахункової сітки на прискорення паралельної версії моделі була обрана задача затоплення берегів р. Дніпро в м. Києві під час весняного водопілля 1979 р.

Весняне водопілля у басейні Дніпра неодноразово супроводжувалось високими повенями у м. Києві, із затопленням його берегів та міської забудови. Для прогнозу наслідків київських повеней в ПММС була розроблена система моделювання паводків на р. Дніпро на ділянці від гирла р. Десна та греблі Київської ГЕС до греблі Канівської ГЕС [15].

Об'єднана цифрова модель рельєфу (ЦМР) дна річок і берегів Києва роздільною здатністю 10 м була створена центром «ГІС Аналітик» на основі даних ехолотної зйомки глибин Дніпра і Десни 2006 р., растрової статистичної поверхні з розрізненням пікселя 10 м, отриманої за даними аерофотозйомки території міста 2010 р., і актуальних даних міського рельєфу м. Києва з розрізненням пікселя 5 м за результатами лідарної зйомки 2020 р. Всі значення приведені до Балтійської системи висот (БС).

На її основі була побудована розрахункова сітка, з 77 111 вузлів та 152 112 комірок, що варіюються за розмірами від 50 м біля русел річок до 300 м на сухих границях області (Рис. 10а). Сітка побудована з ущільненням в районі русла Дніпра та островів, в неї також внесені основні особливості рельєфу берегів: захисні дамби, насипи залізничних шляхів та автодоріг.

Значення коефіцієнта тертя Манінга в розрахунковій області задавались від 0.015  $\text{с/м}^{1/3}$  у Канівському вдсх. до 0.026  $\text{с/м}^{1/3}$  у верхній частині Дніпра, тоді як на суходолі – в діапазоні 0.1-0.15  $\text{с/м}^{1/3}$  (Рис. 10а).

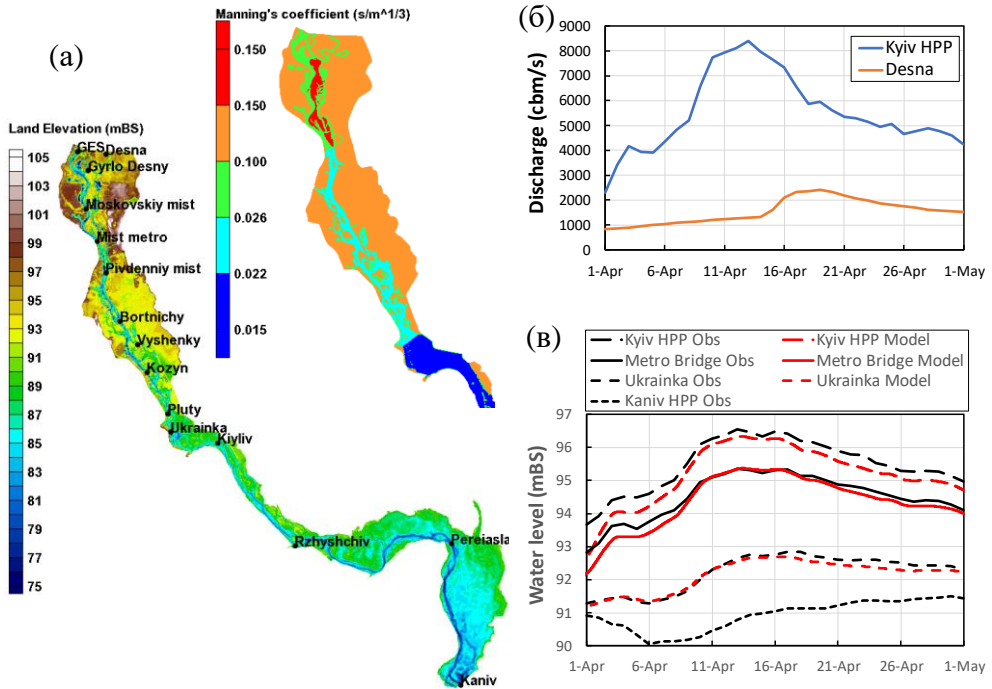


Рис. 10. а) Рельєф розрахункової області (в БС) і розподіл коефіцієнта Манінга для задачі затоплення берегів р. Дніпро в м. Києві під час весняного водопілля 1979 р. б) Виміряні витрати річок Дніпро і Десна під час водопілля 1979 р. в) Рівні води (БС) біля греблі Київської ГЕС, мосту Метро, Українки, Канівської ГЕС за спостереженнями та в моделі

Дані по гідрографам річок та рівнях води на гідропостах були надані Українським гідрометеорологічним центром (Рис. 10б, 10в). Для моделювання на границях задавались, на вхідних – витрата Десни і скид Київської ГЕС, на вихідній – рівень води у верхньому б'єфі Канівської ГЕС. У якості початкових умов рівень води уздовж всієї ділянки Дніпра прирівнювався початковому рівню Канівського водосховища – 90.91 м.

З порівняння промодельованих і спостережених рівнів води (Рис. 10в) слідує, що модель вірно відтворює еволюцію рівня Дніпра під час повені 1979 р. уздовж всієї досліджуваної ділянки, від греблі Київської до греблі Канівської ГЕС.

Для дослідження впливу розміру розрахункової сітки на ефективність паралельної версії моделі з базової розрахункової сітки було створено ще три сітки послідовно більшого розміру шляхом розділення кожного ребра сітки навпіл і кожної трикутної комірки на чотири трикутники. Створені чотири сітки мають довжину ребер базових трикутників уздовж русла Дніпра 50 м, 25 м, 12.5 м і 6.25 м та розміри: 77 092/152 074, 306 257/608 296, 1 220 809/2 433 184, 4 874 801/9 732 736 вузлів/комірок.

Тестування проводилось на багатоядерних процесорах робочої станції Dell і графічних картах Quadro RTX 5000 і GeForce RTX 3080. Для всіх сценаріїв, окрім одного, оцінювався час моделювання 1 дня повені 1979 р. Для найбільшої сітки час моделювання на 1 ядрі робочої станції завеликий. Він оцінювався за часом моделювання 1 години повені. Результати тестування зібрані в Табл. 2.

Таблиця 2. Час обчислення 1 дня паводку 1979 г. у м. Києві на багатоядерній робочій станції і графічних процесорах, для розрахункових сіток різної деталізації. Прискорення паралельних версій COASTOX-HD порівняно з непаралельною. Прискорення GPU версії на графічних процесорах порівняно з MPI версією на робочій станції

$\Delta l$ , m	# of Cells	Time, h				Speedup vs Serial			Speedup vs MC	
		Dell 7920		RTX	RTX	Dell	RTX	RTX	RTX	RTX
		Serial	MC	5000	3080	MC	5000	3080	5000	3080
50	152 074	1.25	0.055	0.037	0.020	22.63	33.57	62.72	1.48	2.77
25	608 296	10.88	0.50	0.19	0.092	21.72	57.54	118.33	2.65	5.45
12.5	2 433 184	113.9	5.28	1.23	0.65	21.58	92.77	174.49	4.30	8.09
6.25	9 732 736	972.5	50.04	10.23	5.38	19.43	95.03	180.64	4.89	9.29

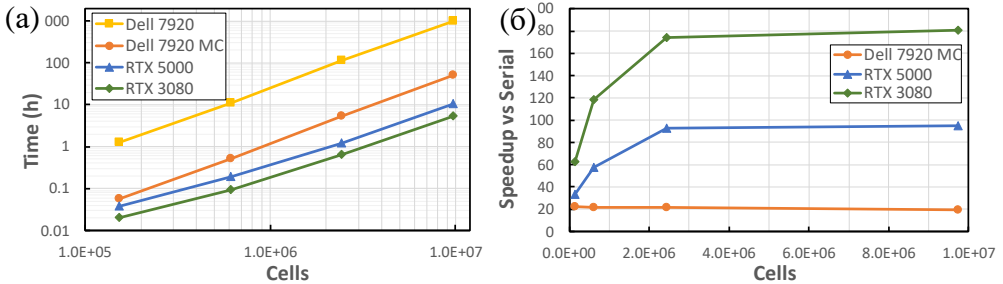


Рис. 11. а) Порівняння часів розрахунку різних версій COASTOX-HD при моделюванні 1 дня паводку 1979 г. у м. Києві на багатоядерній робочій станції і графічних процесорах. б) Прискорення MPI та GPU версій COASTOX-HD порівняно з непаралельною версією

Час обчислення задачі на робочій станції, як у паралельному, так і у непаралельному режимі, має близьку до степеневій залежність від розміру розрахункової сітки (Рис. 11а). Це пов'язано з тим, що чисельна схема розв'язку РМВ в моделі явна, часовий крок обирається з умови Куранта, тому кількість обчислень обернено пропорційна до куба лінійних розмірів комірки. А оскільки кількість комірок обернено пропорційна квадрату лінійних розмірів, кількість обчислень і час приблизно пропорційні кількості комірок у степені 3/2.

Залежність часу моделювання на графічних процесорах від розміру сітки дещо повільніша за степінь 3/2. Саме моделювання на обох використаних у тестуванні графічних процесорах швидше, ніж на робочій станції. Навіть для самої великої сітки, з близько 10 мільйонами комірок, час моделювання менший реального часу і модель може бути використана для прогнозу київських повеней в оперативному режимі.

Прискорення MPI версії моделі слабо залежить від розміру сітки, трохи зменшуючись при зростанні кількості комірок (Рис. 11б). Прискорення GPU версії навпаки різко зростає зі збільшенням розміру сітки до певної її величини (декілька мільйонів комірок), далі зріст уповільнюється, досягаючи насичення (Рис. 11б).

GeForce RTX 3080 на всіх сітках приблизно в 2 рази швидша за Quadro RTX 5000. Порівняння продуктивності відеокарт, 29.77 TFLOPS проти 11.15 TFLOPS, і пропускної здатності пам'яті, 760.3 GB/s проти 448.0 GB/s, підтверджує припущення, що для розробленої паралельної моделі, чисельні схеми якої відносяться до алгоритмів з невисокою обчислювальною інтенсивністю, пропускна здатність пам'яті є більш важливим обмежуючим фактором прискорення, чим продуктивність.

## 7. Висновки

Розроблена паралельна модель гідродинаміки мілкої води на неструктурованих сітках COASTOX-HD, з комплексу моделей COASTOX-UN, дозволяє обчислювати великі задачі гідродинаміки річкових систем і прибережної зони моря на багатопроцесорних комп'ютерних системах і графічних процесорах у десятки разів швидше порівняно з непаралельною моделлю. При цьому моделювання на сучасних графічних процесорах може бути у декілька разів швидше, ніж на багатоядерній робочій станції. В підсумку, обчислення задач з розрахунковими сітками з декількох мільйонів комірок може бути в рази швидшим, ніж реальний час. Таким чином, створена паралельна гідродинамічна модель може бути основою систем оперативного прогнозу надзвичайних ситуацій, пов'язаних із гідродинамікою поверхневих вод, таких, наприклад, як високі водопілля, дощові паводки, штормові нагони або цунамі. Такі системи не потребують наявності суперкомп'ютерів і можуть бути розвернуті на персональних комп'ютерах або ноутбуках, обладнаних сучасними високопродуктивними графічними картами.

Розвинений паралельний скінченно-об'ємний алгоритм розв'язку PMB на базі гібридного підходу MPI+OpenACC також є основою розпаралелення чисельних схем розв'язку рівнянь інших моделей системи COASTOX-UN: моделі переносу наносів і морфодинаміки та моделі транспорту радіонуклідів і радіоактивного забруднення дна водою. Результати розпаралелення цих моделей і структура взаємодії всіх моделей системи COASTOX-UN будуть представлені у наступних публікаціях.

Розвинений гібридний MPI+OpenACC алгоритм, з геометричною декомпозицією розрахункової сітки на підобласті та багатопотоковістю всередині підобластей, може бути адаптований до використання декількох графічних процесорів, що пов'язані через загальну пам'ять або мережевим з'єднанням. Це буде напрямком подальшого дослідження та розвитку системи моделей COASTOX-UN.

## Подяки

Дослідження частково підтримано Національним фондом досліджень України, проєкт № 2020.01/0421.

Автор щиро вдячний науковому керівнику, професору Железняку М.Й. (Institute of Environmental Radioactivity, Fukushima University, Japan) за запропонований напрямок дослідження, плідне обговорення результатів роботи, зауваження та поради під час підготовки матеріалу статті.



## СПИСОК ЛІТЕРАТУРИ

1. Zheleznyak, M. J., Demchenko, R. I., Khursin, S. L., Kuzmenko, Y. I., Tklich, P. V., & Vitiuk, N. Y. (1992). Mathematical modeling of radionuclide dispersion in the Pripyat-Dnieper aquatic system after the Chernobyl accident. *Science of The Total Environment*, 112(1), 89–114. [https://doi.org/10.1016/0048-9697\(92\)90241-j](https://doi.org/10.1016/0048-9697(92)90241-j)
2. Zheleznyak, M. (1997). The mathematical modelling of radionuclide transport by surface water flow from the vicinity of the Chernobyl nuclear power plant. *Condensed Matter Physics*, 12, 37-49. <https://doi.org/10.5488/CMP.12.37>
3. Zheleznyak, M., Shepeleva, T., Sizonenko, V., Mezhueva, I. (1997). Simulation of countermeasures to diminish radionuclide fluxes from the Chernobyl zone via aquatic pathways. *Radiation Protection Dosimetry*, 73(1–4), 181–186. <https://doi.org/10.1093/oxfordjournals.rpd.a032129>
4. Zheleznyak, M., Kivva, S., Pylypenko, O., Sorokin, M. (2022). Modeling of Behavior of Fukushima-Derived Radionuclides in Freshwater Systems. In: Nanba, K., Konoplev, A., Wada, T. (eds) *Behavior of Radionuclides in the Environment III*. Springer, Singapore. 2022, 199–252. [https://doi.org/10.1007/978-981-16-6799-2\\_11](https://doi.org/10.1007/978-981-16-6799-2_11)
5. DHI. (2014). MIKE 21 Flow Model: Hydrodynamic Module User Guide. Danish Hydraulic Institute Water and Environment: Hørsholm, Denmark, p. 132.
6. Ata, R., Goery, C., Lang, P., Desombre, J., Hervouet, J.M. (2014). TELEMAC MODELLING SYSTEM - Release 7.0 - USER MAN-UAL.
7. Sánchez, A., Wu W., Li H., Brown M. E., Reed C. W., Rosati J. D., and Demirbilek Z. (2014). Coastal Modeling System: Mathematical Formulations and Numerical Methods. ERDC/CHL-TR-14-2. U.S. Army ERDC, Vicksburg, MS.
8. Fenical, S., Kolomiets, P., Kivva, S., Zheleznyak, M. (2007). Numerical Modeling of Passing Vessel Impacts on Berthed Vessels and Shoreline. *Coast. Eng.* 2006, 1234-1246.
9. Zheleznyak, M., Khomitsky, V., Kivva, S., Demchenko, R., Kolomiets, P., Shepeleva, T. (2006). Laboratory and Numerical Studies of Waves, Currents and Sediment Transport at the Deepwater Navigation Channel in the “Bystroe” Arm of the Danube Delta. *ICCE-2006, Book of abstracts*, San Diego, California, USA, 2–8 Sept 2006, 1271, p. 400.
10. Kivva, S., Zheleznyak, M. (2008). Modelling of the Risks of the Sidoarjo Mud Flow, Indonesia. *4th International Congress on Environmental Modelling and Software*. Barcelona, Catalonia, Spain. July 2008, pp. 1500-1507.
11. Kolomiets P., Zheleznyak M., Tklich P. (2010). Dynamical Downscaling of Storm Surges in South-China Sea and Singapore Strait. *Storm Surges Congress*, 13-17 September 2010, University of Hamburg, Germany.
12. Дикий П.В., Железняк М.Й., Коломієць П.С., Ківва С.Л., Сорокін М.В. (2013). Моделювання впливу регіональних змін клімату на ерозію берегів Західного Криму. *Тези доповідей Восьмої міжнародної науково-практичної конференції Математичне та імітаційне моделювання систем МОДС 2013*, Чернівці-Жукин, 24-28 червня 2013, с. 56-59.
13. Nanba, K., Zheleznyak, M., Kivva, S., Konoplev, A., Maderich, V., Koshebutsky, V., Gallego, E., Papush, L., Mikhalskyi, O. (2016). Implementation of hydrological dispersion module of JRODOS for the assessment of <sup>137</sup>Cs transport and fate in rivers, reservoirs and ponds of the Fukushima Prefecture. *Radioprotection*, 51(HS2), S145–S148. <https://doi.org/10.1051/radiopro/2016052>
14. Zheleznyak, M., Kivva, S., Ievdin, I., Boyko, O., Kolomiets, P., Sorokin, M., Mikhalskyi, O., Gheorghiu, D. (2016). Hydrological dispersion module of JRODOS: renewed chain of the emergency response models of radionuclide dispersion through watersheds and rivers. *Radioprotection*, 51(HS2), S129–S131. <https://doi.org/10.1051/radiopro/2016048>
15. Бойко В.М., Євдін Є.О., Железняк М.Й., Коломієць П.С., Ішук О.О. (2012). Особливості формування весняного стоку Дніпра та моделювання зон затоплення у межах м. Києва на основі сучасної гідролого-гідрравлічної моделі. *Гідрологія, Гідрохімія, Гідроecологія*. Т. 1(26). с. 55-63.

16. Zheleznyak, M., Kolomiets, P., Dzijuba, N., Ievdin, I., Sorokin, M., Denisov, N., Ischuk, O., and Koeppel, S. (2015). Numerical Modeling for Flood Mapping under Climate Change Impacts: Transboundary Dniester River Study. *EGU General Assembly 2015, Geophysical Research Abstracts*, Vol. 17, EGU2015-9361.
17. Pylypenko, O., Zheleznyak, M., Boyko, O., Kovalets, I., Kivva, S., Khalchenkov, A., Mikhalsky, O., Sorokin, M. (2018). Flood forecasting and flood inundation mapping system developed for Ukrainian parts of Prut and Siret river basins within EAST AVERT project. *EGU General Assembly 2018, Geophys. Res. Abstr.*, Vol. 20, EGU2018-10652-2.
18. Сорокін М.В. (2011). Паралельний явний кінцево-різницевий алгоритм розв'язку рівнянь моделі двовимірної гідроморфодинаміки прибережної зони на неструктурованих сітках. *Тези доповідей Шостої науково-практичної конференції з міжнародною участю Математичне та імітаційне моделювання систем, МОДС 2011*, Чернігів, 27-30 червня 2011, с. 103-105.
19. Sorokin, M., Zheleznyak, M., Kivva, S., Kolomiets, P., and Pylypenko, O. (2020). High performance computing of waves, currents and contaminants in rivers and coastal areas of seas on multi-processors systems and GPUs, *EGU General Assembly 2020*, 4–8 May 2020, EGU2020-11372, <https://doi.org/10.5194/egusphere-egu2020-11372>
20. Leandro, J., Chen, A. S., & Schumann, A. (2014). A 2D parallel diffusive wave model for floodplain inundation with variable time step (P-DWave). *Journal of Hydrology*, 517, 250–259. <https://doi.org/10.1016/j.jhydrol.2014.05.020>
21. Zhang, S., Z. Xia, R. Yuan, and X. Jiang. (2014). Parallel Computation of a Dam-Break Flow Model using OpenMP on a Multi-Core Computer. *J. Hydrol.* 512, 126–133.
22. Zhang, S., Li, W., Jing, Z., Yi, Y., & Zhao, Y. (2017). Comparison of Three Different Parallel Computation Methods for a Two-Dimensional Dam-Break Model. *Mathematical Problems in Engineering*, 2017, 1–12. <https://doi.org/10.1155/2017/1970628>
23. Liu, J.-Y., Smith, M. R., Kuo, F.-A., & Wu, J.-S. (2015). Hybrid OpenMP/AVX acceleration of a Split HLL Finite Volume Method for the Shallow Water and Euler Equations. *Computers & Fluids*, 110, 181–188. <https://doi.org/10.1016/j.compfluid.2014.11.01>
24. Ginting, B.; Mundani, R.P.; Rank, E. (2018). Parallel simulations of shallow water solvers for modelling overland flows. *In Proceedings of the 13th International Conference on Hydroinformatics (HIC 2018)*, EPiC Series in Engineering, Palermo, Italy, 1–6 July 2018; La Loggia, G., Freni, G., Puleo, V., De Marchis, M., Eds.; Volume 3, pp. 788–799.
25. Dietrich, J.C., Zijlema, M., Westerink, J.J., Holthuijsen, L.H., Dawson, C.N., Luettich, R.A. Jr., Jensen, R.E., Smith J.M., Stelling G.S., Stone G.W. (2011). Modeling hurricane waves and storm surge using integrally-coupled, scalable computations. *Coastal Engineering*, 58, 45–65. <https://doi.org/10.1016/j.coastaleng.2010.08.001>
26. Taylor, A., Liu, H. (2020). Latest Developments in the NWS' Sea Lake and Overland Surges from Hurricanes Model. *18th Symposium on the Coastal Environment*, Boston, MA, Amer. Meteor. Soc., 4.1, 8 pp.
27. Rao, P. (2005). A parallel RMA2 model for simulating large-scale free surface flows. *Environmental Modelling & Software*, 20(1), 47–53. <https://doi.org/10.1016/j.envsoft.2003.12.008>
28. Villanueva, I., Wright, N.G. (2006). An Efficient Multi-processor Solver for the 2D Shallow Water Equations. *Hydroinformatics 2006*, Nice, France.
29. Neal, J. C., Fewtrell, T. J., Bates, P. D., & Wright, N. G. (2010). A comparison of three parallelisation methods for 2D flood inundation models. *Environmental Modelling & Software*, 25(4), 398–411. <https://doi.org/j.envsoft.2009.11.007>
30. Shang, Z. (2014). High performance computing for flood simulation using Telemac based on hybrid MPI/OpenMP parallel programming. *International Journal of Modeling, Simulation, and Scientific Computing*, 05(04), 1472001. <https://doi.org/10.1142/s1793962314720015>

31. Williams S.W., Waterman A., Patterson D.A. (2008). Roofline: An Insightful Visual Performance Model for Floating-Point Programs and Multicore Architectures. *Technical Report No. UCB/EECS-2008-134*.
32. Zhang, Y., & Jia, Y. (2013). Parallelized CCHE2D flow model with CUDA Fortran on Graphics Processing Units. *Computers & Fluids*, 84, 359–368. <https://doi.org/10.1016/j.compfluid.2013.06.021>
33. Huxley, C., Syme, B. (2016). TUFLOW GPU - Best Practice Advice for Hydrologic and Hydraulic Model Simulations. In *Proceedings of the 37th Hydrology and Water Resources Symposium (HWRS)*, Queenstown, New Zealand.
34. García-Feal O., González-Cao J., Gómez-Gesteira M., Cea L., Domínguez J.M., Formella A. (2018). An Accelerated Tool for Flood Modelling Based on Iber. *Water*, 10(10), 1459. <https://doi.org/10.3390/w10101459>
35. Morales-Hernández, M., Sharif, M. B., Kalyanapu, A., Ghafoor, S. K., Dullo, T. T., Gangrade, S., Kao S.-C., Norman M.R., Evans, K. J. (2021). TRITON: A Multi-GPU open source 2D hydrodynamic flood model. *Environmental Modelling & Software*, 141, 105034. <https://doi.org/10.1016/j.envsoft.2021.105034>
36. Hoshino, T., Maruyama, N., Matsuoka, S., & Takaki, R. (2013). CUDA vs OpenACC: Performance Case Studies with Kernel Benchmarks and a Memory-Bound CFD Application. *2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing*, 136-143. <https://doi.org/10.1109/ccgrid.2013.12>
37. Hu, X., Song, L. (2018). Hydrodynamic modeling of flash flood in mountain watersheds based on high-performance GPU computing. *Nat Hazards*, 91, 567–586. <https://doi.org/10.1007/s11069-017-3141-7>
38. Liu Q., Qin Y., Li G. (2018). Fast Simulation of Large-Scale Floods Based on GPU Parallel Computing. *Water*, 10(5), 589. <https://doi.org/10.3390/w10050589>
39. Xia, Y., Luo, L., Luo, H., Lou, J., Edwards, J. R., & Mueller, F. (2014). On the Multi-GPU Computing of a Reconstructed Discontinuous Galerkin Method for Compressible Flows on 3D Hybrid Grids. *7th AIAA Theoretical Fluid Mechanics Conference*. <https://doi.org/10.2514/6.2014-3081>
40. Xue, W., Jackson, C. W., & Roy, C. J. (2018). Multi-CPU/GPU parallelization, optimization and machine learning based autotuning of structured grid CFD codes. *2018 AIAA Aerospace Sciences Meeting*. <https://doi.org/10.2514/6.2018-0362>
41. Zia A., Banihashemi M.A. (2008). Simple efficient algorithm (SEA) for shallow water flows with shock wave on dry and irregular beds. *International Journal for Numerical Methods in Fluids*, 56(11), 2021–2043. <https://doi.org/10.1002/flid.1585>
42. Begnudelli L., Sanders B.F. (2006). Unstructured grid finite-volume algorithm for shallow-water flow and scalar transport with wetting and drying. *Journal of Hydraulic Engineering*, 132(4), 371-384. [https://doi.org/10.1061/\(ASCE\)0733-9429\(2006\)132:4\(371\)](https://doi.org/10.1061/(ASCE)0733-9429(2006)132:4(371))
43. Toro, E.F., Spruce, M. & Speares, W. (1994). Restoration of the contact surface in the HLL-Riemann solver. *Shock Waves*, 4(1), 25–34. <https://doi.org/10.1007/BF01414629>
44. Roe P.L. (1981). Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43(2), 357-372. [https://doi.org/10.1016/0021-9991\(81\)90128-5](https://doi.org/10.1016/0021-9991(81)90128-5)
45. Karypis G., Kumar V. METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices, Version 4.0. *Technical Report*, Dept. of Computer Science and Eng. University of Minnesota, Minneapolis, MN. 1998.
46. Amdahl, G. M. (1967). Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities. *AFIPS Conference Proceedings*, 30, 483–485. <https://doi.org/10.1145/1465482.1465560>
47. Synolakis C.E., Bernard E.N., Titov V.V., Kânoğlu U., and González F.I. (2007). Standards, criteria, and procedures for NOAA evaluation of tsunami numerical models. *NOAA Tech. Memo. OAR PMEL-135*, NOAA/PMEL, Seattle, WA, 2007, 55 pp.

48. Amante, C., Eakins, B. W. (2009). Etopo1 1 arc-minute global relief model: Procedures, data sources and analysis. *NOAA Technical Memorandum NESDIS NGDC-24*, March 2009, 19 pp.
49. Jarvis A., Reuter, H. I., Nelson, A., and Guevara, E.: Hole-filled seamless SRTM data V4, International Centre for Tropical Agriculture. <http://srtm.csi.cgiar.org>
50. Muhari, A., Imamura, F., Suppasri, A., & Mas, E. (2012). Tsunami arrival time characteristics of the 2011 East Japan Tsunami obtained from eyewitness accounts, evidence and numerical simulation. *Journal of Natural Disaster Science*, 34(1), 91–104. <https://doi.org/10.2328/jnds.34.91>
51. Japan Meteorological Agency, Tsunami observation data at Soma. <http://www.jma.go.jp/jma/press/1104/13a/201104131600.html>
52. Japan Society of Geoinformatics, Inundation Areas of 2011 Tsunami in East Japan. <http://www.jsgi-map.org/tsunami/>

Стаття надійшла до редакції 22.03.2023 і прийнята до друку після рецензування 09.06.2023

## REFERENCES

1. Zheleznyak, M. J., Demchenko, R. I., Khursin, S. L., Kuzmenko, Y. I., Tklich, P. V., & Vitiuk, N. Y. (1992). Mathematical modeling of radionuclide dispersion in the Pripyat-Dnieper aquatic system after the Chernobyl accident. *Science of The Total Environment*, 112(1), 89–114. [https://doi.org/10.1016/0048-9697\(92\)90241-j](https://doi.org/10.1016/0048-9697(92)90241-j)
2. Zheleznyak, M. (1997). The mathematical modelling of radionuclide transport by surface water flow from the vicinity of the Chernobyl nuclear power plant. *Condensed Matter Physics*, 12, 37-49. <https://doi.org/10.5488/CMP.12.37>
3. Zheleznyak, M., Shepeleva, T., Sizonenko, V., Mezheva, I. (1997). Simulation of countermeasures to diminish radionuclide fluxes from the Chernobyl zone via aquatic pathways. *Radiation Protection Dosimetry*, 73(1–4), 181–186. <https://doi.org/10.1093/oxfordjournals.rpd.a032129>
4. Zheleznyak, M., Kivva, S., Pylypenko, O., Sorokin, M. (2022). Modeling of Behavior of Fukushima-Derived Radionuclides in Freshwater Systems. In: Nanba, K., Konoplev, A., Wada, T. (eds) *Behavior of Radionuclides in the Environment III*. Springer, Singapore. 2022, 199–252. [https://doi.org/10.1007/978-981-16-6799-2\\_11](https://doi.org/10.1007/978-981-16-6799-2_11).
5. DHI. (2014). MIKE 21 Flow Model: Hydrodynamic Module User Guide. Danish Hydraulic Institute Water and Environment: Hørsholm, Denmark, p. 132.
6. Ata, R., Goery, C., Lang, P., Desombre, J., Hervouet, J.M. (2014). TELEMAC MODELLING SYSTEM - Release 7.0 - USER MAN-UAL.
7. Sánchez, A., Wu W., Li H., Brown M. E., Reed C. W., Rosati J. D., and Demirbilek Z. (2014). Coastal Modeling System: Mathematical Formulations and Numerical Methods. ERDC/CHL-TR-14-2. U.S. Army ERDC, Vicksburg, MS.
8. Fenical, S., Kolomiets, P., Kivva, S., Zheleznyak, M. (2007). Numerical Modeling of Passing Vessel Impacts on Berthed Vessels and Shoreline. *Coastal Engineering 2006*, 1234-1246.
9. Zheleznyak, M., Khomitsky, V., Kivva, S., Demchenko, R., Kolomiets, P., Shepeleva, T. (2006). Laboratory and Numerical Studies of Waves, Currents and Sediment Transport at the Deepwater Navigation Channel in the “Bystroe” Arm of the Danube Delta. *ICCE-2006, Book of abstracts*, San Diego, California, USA, 2–8 September 2006, No: 1271, p. 400.
10. Kivva, S., Zheleznyak, M. (2008). Modelling of the Risks of the Sidoarjo Mud Flow, Indonesia. *4th International Congress on Environmental Modelling and Software*. Barcelona, Catalonia, Spain. July 2008, pp. 1500-1507.
11. Kolomiets P., Zheleznyak M., Tklich P. (2010). Dynamical Downscaling of Storm Surges in South-China Sea and Singapore Strait. *Storm Surges Congress*, 13-17 September 2010, University of Hamburg, Germany.

12. Dikiy P.V., Zheleznyak M.J., Kolomiets P.S., Kivva S.L., Sorokin M.V. (2013). Modeling the impact of regional climate change on coastal erosion in the Western Crimea. *Book of Abstracts of the Eighth International scientific-practical conference "Mathematical Modeling and Simulation of Systems. MODS 2013"*, Chernihiv-Zhukyn, June 24-28 2013, pp. 56-59 [in Ukrainian].
13. Nanba, K., Zheleznyak, M., Kivva, S., Konoplev, A., Maderich, V., Koshebutsky, V., Gallego, E., Papush, L., Mikhalskyi, O. (2016). Implementation of hydrological dispersion module of JRODOS for the assessment of <sup>137</sup>Cs transport and fate in rivers, reservoirs and ponds of the Fukushima Prefecture. *Radioprotection*, 51(HS2), S145–S148. <https://doi.org/10.1051/radiopro/2016052>
14. Zheleznyak, M., Kivva, S., Ievdin, I., Boyko, O., Kolomiets, P., Sorokin, M., Mikhalskyi, O., Gheorghiu, D. (2016). Hydrological dispersion module of JRODOS: renewed chain of the emergency response models of radionuclide dispersion through watersheds and rivers. *Radioprotection*, 51(HS2), S129–S131. <https://doi.org/10.1051/radiopro/2016048>
15. Boyko V.M., Ievdin I.O., Zheleznyak M.J., Kolomiets P.S., Ischuk O.O. (2012). Peculiarities of spring runoff formation of the Dnieper River and modeling of flood zones within the city of Kiev on the basis of modern hydrological and hydraulic model. *Hydrology, hydrochemistry and hydroecology*, 1(26), 55-63 [in Ukrainian].
16. Zheleznyak, M., Kolomiets, P., Dzijuba, N., Ievdin, I., Sorokin, M., Denisov, N., Ischuk, O., and Koepfel, S. (2015). Numerical Modeling for Flood Mapping under Climate Change Impacts: Transboundary Dniester River Study. *EGU General Assembly 2015, Geophysical Research Abstracts*, Vol. 17, EGU2015-9361.
17. Pylypenko, O., Zheleznyak, M., Boyko, O., Kovalets, I., Kivva, S., Khalchenkov, A., Mikhalsky, O., Sorokin, M. (2018). Flood forecasting and flood inundation mapping system developed for Ukrainian parts of Prut and Siret river basins within EAST AVERT project. *EGU General Assembly 2018, Geophys. Res. Abstr.*, Vol. 20, EGU2018-10652-2.
18. Sorokin M.V. (2011). A parallel explicit finite-difference algorithm for solving the equations of the two-dimensional hydromorphodynamics model of the coastal zone on unstructured grids. *Book of Abstracts of the Sixth International scientific-practical conference "Mathematical Modeling and Simulation of Systems. MODS 2011"*, Chernihiv, June 27-30 2011, pp. 103-105 [in Ukrainian].
19. Sorokin, M., Zheleznyak, M., Kivva, S., Kolomiets, P., and Pylypenko, O. (2020). High performance computing of waves, currents and contaminants in rivers and coastal areas of seas on multi-processors systems and GPUs, *EGU General Assembly 2020*, 4–8 May 2020, EGU2020-11372, <https://doi.org/10.5194/egusphere-egu2020-11372>
20. Leandro, J., Chen, A. S., & Schumann, A. (2014). A 2D parallel diffusive wave model for floodplain inundation with variable time step (P-DWave). *Journal of Hydrology*, 517, 250–259. <https://doi.org/10.1016/j.jhydrol.2014.05.020>
21. Zhang, S., Z. Xia, R. Yuan, and X. Jiang. (2014). Parallel Computation of a Dam-Break Flow Model using OpenMP on a Multi-Core Computer. *J. Hydrol.* 512, 126–133.
22. Zhang, S., Li, W., Jing, Z., Yi, Y., & Zhao, Y. (2017). Comparison of Three Different Parallel Computation Methods for a Two-Dimensional Dam-Break Model. *Mathematical Problems in Engineering*, 2017, 1–12. <https://doi.org/10.1155/2017/1970628>
23. Liu, J.-Y., Smith, M. R., Kuo, F.-A., & Wu, J.-S. (2015). Hybrid OpenMP/AVX acceleration of a Split HLL Finite Volume Method for the Shallow Water and Euler Equations. *Computers & Fluids*, 110, 181–188. <https://doi.org/10.1016/j.compfluid.2014.11.01>
24. Ginting, B.; Mundani, R.P.; Rank, E. (2018). Parallel simulations of shallow water solvers for modelling overland flows. *In Proceedings of the 13th International Conference on Hydroinformatics (HIC 2018)*, EPiC Series in Engineering, Palermo, Italy, 1–6 July 2018; La Loggia, G., Freni, G., Puleo, V., De Marchis, M., Eds.; Volume 3, pp. 788–799.
25. Dietrich, J.C., Zijlema, M., Westerink, J.J., Holthuijsen, L.H., Dawson, C.N., Luettich, R.A. Jr., Jensen, R.E., Smith J.M., Stelling G.S., Stone G.W. (2011). Modeling hurricane waves and storm surge using integrally-coupled, scalable computations. *Coastal Engineering*, 58, 45–65. <https://doi.org/10.1016/j.coastaleng.2010.08.001>

26. Taylor, A., Liu, H. (2020). Latest Developments in the NWS' Sea Lake and Overland Surges from Hurricanes Model. *18th Symposium on the Coastal Environment*, Boston, MA, Amer. Meteor. Soc., 4.1, 8 pp.
27. Rao, P. (2005). A parallel RMA2 model for simulating large-scale free surface flows. *Environmental Modelling & Software*, 20(1), 47–53. <https://doi.org/10.1016/j.envsoft.2003.12.008>
28. Villanueva, I., Wright, N.G. (2006). An Efficient Multi-processor Solver for the 2D Shallow Water Equations. *Hydroinformatics 2006*, Nice, France.
29. Neal, J. C., Fewtrell, T. J., Bates, P. D., & Wright, N. G. (2010). A comparison of three parallelisation methods for 2D flood inundation models. *Environmental Modelling & Software*, 25(4), 398–411. <https://doi.org/j.envsoft.2009.11.007>
30. Shang, Z. (2014). High performance computing for flood simulation using Telemac based on hybrid MPI/OpenMP parallel programming. *International Journal of Modeling, Simulation, and Scientific Computing*, 05(04), 1472001. <https://doi.org/10.1142/s1793962314720015>
31. Williams S.W., Waterman A., Patterson D.A. (2008). Roofline: An Insightful Visual Performance Model for Floating-Point Programs and Multicore Architectures. *Technical Report No. UCB/EECS-2008-134*.
32. Zhang, Y., & Jia, Y. (2013). Parallelized CCHE2D flow model with CUDA Fortran on Graphics Processing Units. *Computers & Fluids*, 84, 359–368. <https://doi.org/10.1016/j.compfluid.2013.06.021>
33. Huxley, C., Syme, B. (2016). TUFLOW GPU - Best Practice Advice for Hydrologic and Hydraulic Model Simulations. *In Proceedings of the 37th Hydrology and Water Resources Symposium (HWRS)*, Queenstown, New Zealand.
34. García-Feal O., González-Cao J., Gómez-Gesteira M., Cea L., Domínguez J.M., Formella A. (2018). An Accelerated Tool for Flood Modelling Based on Iber. *Water*, 10(10), 1459. <https://doi.org/10.3390/w10101459>
35. Morales-Hernández, M., Sharif, M. B., Kalyanapu, A., Ghafoor, S. K., Dullo, T. T., Gangrade, S., Kao S.-C., Norman M.R., Evans, K. J. (2021). TRITON: A Multi-GPU open source 2D hydrodynamic flood model. *Environmental Modelling & Software*, 141, 105034. <https://doi.org/10.1016/j.envsoft.2021.105034>
36. Hoshino, T., Maruyama, N., Matsuoka, S., & Takaki, R. (2013). CUDA vs OpenACC: Performance Case Studies with Kernel Benchmarks and a Memory-Bound CFD Application. *2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing*, 136-143. <https://doi.org/10.1109/ccgrid.2013.12>
37. Hu, X., Song, L. (2018). Hydrodynamic modeling of flash flood in mountain watersheds based on high-performance GPU computing. *Nat Hazards*, 91, 567–586. <https://doi.org/10.1007/s11069-017-3141-7>
38. Liu Q., Qin Y., Li G. (2018). Fast Simulation of Large-Scale Floods Based on GPU Parallel Computing. *Water*, 10(5), 589. <https://doi.org/10.3390/w10050589>
39. Xia, Y., Luo, L., Luo, H., Lou, J., Edwards, J. R., & Mueller, F. (2014). On the Multi-GPU Computing of a Reconstructed Discontinuous Galerkin Method for Compressible Flows on 3D Hybrid Grids. *7th AIAA Theoretical Fluid Mechanics Conference*. <https://doi.org/10.2514/6.2014-3081>
40. Xue, W., Jackson, C. W., & Roy, C. J. (2018). Multi-CPU/GPU parallelization, optimization and machine learning based autotuning of structured grid CFD codes. *2018 AIAA Aerospace Sciences Meeting*. <https://doi.org/10.2514/6.2018-0362>
41. Zia A., Banihashemi M.A. (2008). Simple efficient algorithm (SEA) for shallow water flows with shock wave on dry and irregular beds. *International Journal for Numerical Methods in Fluids*, 56(11), 2021–2043. <https://doi.org/10.1002/flid.1585>
42. Begnudelli L., Sanders B.F. (2006). Unstructured grid finite-volume algorithm for shallow-water flow and scalar transport with wetting and drying. *Journal of Hydraulic Engineering*, 132(4), 371-384. [https://doi.org/10.1061/\(ASCE\)0733-9429\(2006\)132:4\(371\)](https://doi.org/10.1061/(ASCE)0733-9429(2006)132:4(371))
43. Toro, E.F., Spruce, M. & Speares, W. (1994). Restoration of the contact surface in the HLL-Riemann solver. *Shock Waves*, 4(1), 25–34. <https://doi.org/10.1007/BF01414629>

44. Roe P.L. (1981). Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43(2), 357-372. [https://doi.org/10.1016/0021-9991\(81\)90128-5](https://doi.org/10.1016/0021-9991(81)90128-5)
45. Karypis G., Kumar V. METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices, Version 4.0. *Technical Report*, Dept. of Computer Science and Eng. University of Minnesota, Minneapolis, MN. 1998.
46. Amdahl, G. M. (1967). Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities. *AFIPS Conference Proceedings*, 30, 483-485. <https://doi.org/10.1145/1465482.1465560>
47. Synolakis C.E., Bernard E.N., Titov V.V., Kânoğlu U., and González F.I. (2007). Standards, criteria, and procedures for NOAA evaluation of tsunami numerical models. *NOAA Tech. Memo. OAR PMEL-135*, NOAA/PMEL, Seattle, WA, 2007, 55 pp.
48. Amante, C., Eakins, B. W. (2009). Etopo1 1 arc-minute global relief model: Procedures, data sources and analysis. *NOAA Technical Memorandum NESDIS NGDC-24*, March 2009, 19 pp.
49. Jarvis A., Reuter, H. I., Nelson, A., and Guevara, E.: Hole-filled seamless SRTM data V4, International Centre for Tropical Agriculture. <http://srtm.csi.cgiar.org>
50. Muhari, A., Imamura, F., Suppasri, A., & Mas, E. (2012). Tsunami arrival time characteristics of the 2011 East Japan Tsunami obtained from eyewitness accounts, evidence and numerical simulation. *Journal of Natural Disaster Science*, 34(1), 91-104. <https://doi.org/10.2328/jnds.34.91>
51. Japan Meteorological Agency, Tsunami observation data at Soma. <http://www.jma.go.jp/jma/press/1104/13a/201104131600.html>
52. Japan Society of Geoinformatics, Inundation Areas of 2011 Tsunami in East Japan. <http://www.jsgi-map.org/tsunami/>

*The article was received 22.03.2023 and was accepted after revision 09.06.2023*

**Сорокін Максим Вікторович**

науковий співробітник

Інститут проблем математичних машин і систем НАН України

**Адреса робоча:** 03187 Україна, м. Київ, проспект Академіка Глушкова, 42

**ORCID ID:** <https://orcid.org/0000-0002-3165-6040> **e-mail:** maxim.sorockin@gmail.com