

networks (T2FWN) for system identification using fuzzy differential and Lyapunov stability algorithm // *Applied Soft Computing*. – 2009. – 9(3). – P. 977-989. 11. Wu H., Mendel J.M. Uncertainty bounds and their use in the design of interval type-2 fuzzy logic system // *IEEE Tran. on Fuzzy Systems*. – 2002. – 10(5). – P. 622-639. 12. Mendel J. *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions* // Upper-Saddle River, NJ: Prentice Hall, 2001. – 555 p. 13. Бодянский Е.В., Винокурова Е.А. Вэйвлет-нейро-фаззи система типа-2 и алгоритм ее обучения в задачах интеллектуальной обработки информации // *Адаптивные системы автоматического управления*. – К.: НТУУ «КПИ», 2010. – 17(37). – С. 139–148. 14. Бодянский Е.В., Винокурова Е.А., Харченко А.А. Прогнозирующая вэйвлет-нейро-фаззи-система типа-2 // *Матеріали міжнародної наукової конференції «Інтелектуальні системи прийняття рішень та проблеми обчислювального інтелекту»*, Херсон. – 2011. – Т. 1. – С. 215–220. 15. Бодянский Е.В., Винокурова Е.А. Адаптивный вэйвлет-нейронный предиктор // *Проблемы бионики*. – 2003. – Вып. 58. – С. 10-17. 16. Yamakawa T., Uchino E., Miki T., Kusanagi H. A neo-fuzzy neuron and its application to system identification and prediction of the system behavior // *Proc. 2nd Int. Conf. on Fuzzy Logic and Neural Networks, IZUKA-92, Iizuka, Japan, 1992*. – vol. II. – P. 477–483. 17. Mitaim S., Kosko B. Adaptive joint fuzzy sets for function approximation // *Proc. Int. Conf. on Neural Networks*. – 1997. – P. 537–542. 18. Shepherd A. J. *Second-Order Methods for Neural Networks* – London: Springer-Verlag, 1997. – 145 p. 19. Бодянский Е. В., Михальов О. І., Плісс І. П. Адаптивне виявлення розладнань в об'єктах керування за допомогою штучних нейронних мереж. – Дніпропетровськ: Системні технології, 2000. – 140 с. 20. Otto P., Budyanskiy Ye., Kolodyazhniy V. A new learning algorithm for a forecasting neuro-fuzzy network // *Integrated Computer-Aided Engineering*. – 10. – N4. – Amsterdam: IOS Press, 2003. – P. 399–409.

УДК 621.39

Я. Соколовський, О. Мокрицька

Національний лісотехнічний університет України

ОБ'ЄКТНО-ОРИЄНТОВАНА РЕАЛІЗАЦІЯ МЕТОДУ СКІНЧЕННИХ ЕЛЕМЕНТІВ ДЛЯ РОЗРАХУНКУ В'ЯЗКОПРУЖНОГО СТАНУ КАПЛІАРНО-ПОРИСТИХ МАТЕРІАЛІВ

© Соколовський Я., Мокрицька О., 2011

У межах об'єктно-орієнтованого підходу та UML розроблено програмне за забезпечення для реалізації математичної моделі в'язкопружного деформування капілярно-пористих матеріалів з використанням методу скінченних елементів. Створене об'єктно-орієнтоване програмне забезпечення складається зі задокументованих класів, які можуть повторно використовуватись для реалізації нових моделей.

Ключові слова: математична модель, об'єктно-орієнтований підхід, метод скінченних елементів, в'язкопружний стан, пористий матеріал.

In the object-oriented approach and UML developed software to implement the mathematical model of viscoelastic deformation of capillary-porous materials using the finite element method. The created object-oriented software consists of documented classes that can be reused to implement new models.

Keywords: mathematical model, object-oriented approach, finite element method, viscoelasticity, porous material.

Актуальність досліджень

Програмне забезпечення автоматизованих систем розрахунку та аналізу механічних систем переважно створюють за підходами структурного програмування з використанням мов програмування високого рівня. Подальшого вдосконалення та підвищення ефективності програмного

забезпечення автоматизації розрахунків фізико-механічних полів можна досягти шляхом переходу до об'єктно-орієнтованого аналізу та проектування програмних систем.

Здійснено спробу побудувати пакет програмних модулів на основі об'єктно-орієнтованого підходу для математичного моделювання та аналізу деформаційно-релаксаційних процесів у капілярно-пористих матеріалах методом скінченних елементів (МСЕ). Здійснено аналіз та проектування структури об'єктно-орієнтованої реалізації МСЕ математичної моделі в'язко-пружного деформування капілярно-пористих матеріалів. Відношення між створеними класами та взаємодію об'єктів цих класів відображено у графічній нотації UML (United Modeling Language).

Аналіз результатів

Сьогодні існує порівняно небагато публікацій застосування об'єктно-орієнтованого підходу для програмної реалізації МСЕ [1–7], в яких запроєктовано класи базових компонентів МСЕ (елементів, вузлів, граничних умов та навантажень). Їх аналіз дає змогу стверджувати, що безпосереднє створення об'єктів у програмному коді породжує необхідність перекомпілювання програми для нових даних. Потребує подальшого вдосконалення задача візуального створення та редагування об'єктів шляхом їх інтегрування з генераторами розбиття геометричної області. Сам МСЕ потребує виконання значних обсягів числової інформації. В існуючих об'єктно-орієнтованих моделях введено об'єкти інтегрування в часі та розв'язання системи лінійних алгебраїчних рівнянь (СЛАР) [3–4]. Відсутність явно виділеного класу їх розв'язання зумовлює переписування коду ітераційного процесу та унеможливорює повторне використання програмних кодів. Наукові публікації [8–12] свідчать про використання об'єктної моделі адаптивних методів скінченних елементів, коли на основі аналізу отриманих розв'язків визначається стратегія подальших обчислень. Важливими є дослідження [13] декомпозицій та паралельних розподілених технологій для адаптивних методів скінчених елементів.

Алгоритмічні аспекти реалізації МСЕ в'язкопружного деформування потенціалів. Метод скінченних елементів полягає у такому: будь-яку неперервну величину (у цьому випадку компоненти напружено-деформівного стану) можна апроксимувати дискретною моделлю, яка будується на множині кусково-неперервних функцій, що визначені на скінченному числі підобластей [14]. Побудова дискретної моделі ґрунтується на основі співвідношень математичної моделі дослідження в'язкопружного стану капілярно-пористих тіл як трифазної системи, що складається з твердої фази, рідкої і пароповітряної фаз [15, 16]. Систему модельних рідин побудовано у межах механіки багатофазних систем з врахуванням реологічної поведінки матеріалу, зокрема механізму накопичення незворотних деформацій та деформацій механіко-сорбційної повзучості [15]. Для реалізації математичної моделі здійснено варіаційне формулювання задачі на основі принципу мінімуму повної потенціальної енергії [2, 6]. Функціонал Лапласа у переміщеннях $\bar{U} = (U_x, U_y)$ записується у вигляді

$$\Pi(u) = \frac{1}{2} \int_{\Omega} U^T B^T C B U d\Omega - \frac{1}{2} \int_{\Omega} U^T B^T C \int_0^t R(t,s) B U ds d\Omega, \quad (1)$$

де C – тензор пружних характеристик, $R(t,s)$ – тензор ядра релаксації, B – матриця виду

$$B^T = \begin{pmatrix} \frac{\partial}{\partial x} & 0 & \frac{1}{2} \frac{\partial}{\partial y} \\ 0 & \frac{\partial}{\partial y} & \frac{1}{2} \frac{\partial}{\partial x} \end{pmatrix}, \quad \Omega - \text{геометрична область } \Omega = \{(x, y) \mid 0 \leq x \leq l_1; 0 \leq y \leq l_2\}, \quad l_1, l_2 -$$

геометричні розміри, t – час. Просторово-часову апроксимацію переміщень подано у вигляді

$$U_i(t) = (U_i(t_{j-1}) + U_i(t_j)) \frac{t_j - t}{t_j t_{j-1}} \quad (2)$$

де $U_i(t_{j-1}), U_i(t_j)$ – вектор переміщень у вузлах t_{j-1}, t_j .

З умов мінімуму $\Pi(U) \rightarrow \min$ отримують основні співвідношення МСЕ, згідно з якими задача визначення переміщень U зводиться до визначення СЛАР.

Програмна реалізація моделі МСЕ. Для побудови програмної системи виділено програмні модулі, які є одиницею об'єктно-орієнтованої реалізації алгоритму (1)–(2) у контексті основних понять і термінів МСЕ. Такий підхід дає змогу розкласти побудовану дискретну модель у вигляді класів (пакетів класів) та відношень між ними. Пояснення зв'язків між спроектованими класами та взаємодію об'єктів цих класів здійснено за допомогою діаграм UML. В окремі пакети виділено класи, які реалізують: геометричні та фізико-механічні характеристики об'єкта досліджень; розбиття області на скінченні елементи за допомогою сітки вузлів; визначення базисних функцій у межах скінченних елементів; обчислювальні класи (квадратури для чисельного інтегрування); інтерполяційні функції; розв'язання систем лінійних алгебраїчних рівнянь (СЛАР); класи орієнтовані на конкретні обчислення матричної та векторної алгебри; класи збереження введення та виведення даних; інтерфейс користувача.

Для опису кожної такої групи використано механізм інтерфейсів. Кожен з таких інтерфейсів містить сигнатури функцій, які описують такі сутності:

- температурні та вологісні коефіцієнти, що містить задача визначення потоків масо-перенесення у рідкій, твердій та газоподібній фазах, описані у вигляді функцій, що залежать від температури та вологості та інших аргументів (ITemperatureCoefficients);
- коефіцієнти, необхідні для опису задачі в'язкопружності, які також обчислюються залежно від температури та вологовмісту матеріалу (IStrainCoefficients);
- параметри зовнішнього середовища, а саме температура середовища t_c , відносна вологість j , виділені в окремий інтерфейс (IEnvironmentCoefficients);
- параметри числового розв'язування – такі, як кількість розбиттів за часом, кількість розбиттів за координатними осями, порядок квадратурних формул для обчислення інтегралів тощо (IProblemOptions);
- початкові значення температури t_0 , вологовмісту u_0 матеріалу у початкових умовах границі умови для компонент переміщень \vec{U} та напружень S на границях області Ω , а також геометричні розміри матеріалу (l_1 і l_2) та тривалість процесу Θ (IProblemConditions).

Ці п'ять інтерфейсів (рис. 1) реалізовані у чотирьох класах (фізико-механічні коефіцієнти задач масоперенесення та визначення напружено-деформівного стану об'єднані в один клас). Класи являють собою візуальні компоненти – діалогові вікна, у яких користувач може задати значення вказаних параметрів. Вони наслідують спільний абстрактний клас (FCoefsDialog), який об'єднує функції, необхідні для роботи діалогових вікон.

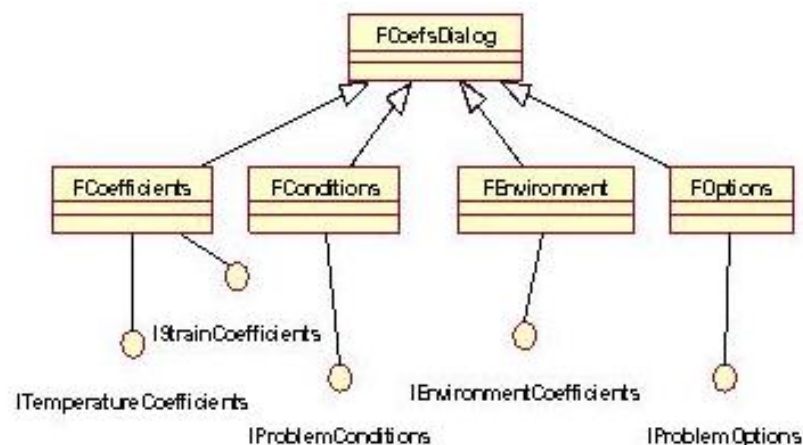


Рис. 1. Розбиття вхідних даних задачі на класи

Після того, як задано усі необхідні умови задачі, можна приступати до реалізації інших етапів. Відразу ж постає питання вибору форми скінченних елементів та спосіб розбиття області на ці елементи. Зважаючи на те, що область Ω задачі має прямокутну форму, можна успішно використати прямокутні елементи.

Поділивши область на множину прямокутних елементів, у межах кожного елемента перейдемо до локальної системи безрозмірних координат $\Omega_e = \{\{\mathbf{x}, \mathbf{h}\}: -1 \leq x \leq 1, -1 \leq h \leq 1\}$ за допомогою перетворень:

$$\mathbf{x} = \frac{2x - x_{1\max} - x_{1\min}}{x_{1\max} - x_{1\min}} \quad ; \quad \mathbf{h} = \frac{2x - x_{2\max} - x_{2\min}}{x_{2\max} - x_{2\min}},$$

де $(x_{1\min}, x_{2\min})$ та $(x_{1\max}, x_{2\max})$ – координати відповідно лівого нижнього та правого верхнього кутів прямокутного скінченного елемента (рис. 2).

Найпоширенішими є два сімейства прямокутних елементів: лагранжеві (із внутрішніми вузлами) та сирендипові (без внутрішніх вузлів). У програмній реалізації використано лагранжеві елементи, які походять безпосередньо від поліномів Лагранжа:

$$L_k^n(\mathbf{x}) = \frac{(\mathbf{x} - \mathbf{x}_0)\mathbf{L}(\mathbf{x} - \mathbf{x}_{k-1})(\mathbf{x} - \mathbf{x}_{k+1})\mathbf{L}(\mathbf{x} - \mathbf{x}_n)}{(\mathbf{x}_k - \mathbf{x}_0)\mathbf{L}(\mathbf{x}_k - \mathbf{x}_{k-1})(\mathbf{x}_k - \mathbf{x}_{k+1})\mathbf{L}(\mathbf{x}_k - \mathbf{x}_n)}$$

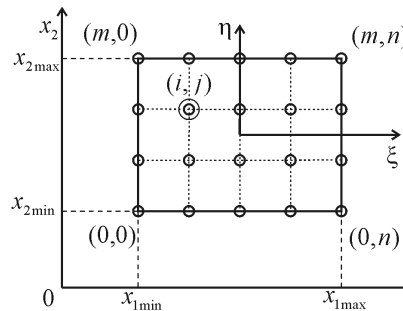


Рис. 2. Стандартний скінченний елемент типу Лагранжа

Якщо розглянути прямокутний елементи (рис. 2) із рівномірно віддаленими n вузлами в напрямку радіальної осі x_1 та m – в напрямку тангентальної осі x_2 , то для вузла ij базисна функція виражається через добуток двох поліномів Лагранжа

$$N_{ij}(\mathbf{x}, \mathbf{h}) = L_i^n(\mathbf{x}) \cdot L_j^m(\mathbf{h}) \quad (3)$$

Прикладна програма реалізує три типи елементів Лагранжа: лінійний, квадратичний та кубічний. Визначені в межах скінченного елемента базисні функції різних порядків отримуємо із формули (3).

Алгоритм методу скінченних елементів передбачає, що скінченні елементи та базисні функції взаємопов'язані. При програмуванні цих двох сутностей як класів між ними можна поставити зв'язок узагальнення: *скінченний елемент* (TUElem) наслідує порядок, що зберігається у класі *базисних функцій* (TUInterpolation) (рис. 3). Також клас (TUElem) використовується для обчислення об'ємних та криволінійних інтегралів, що входять у варіаційні формулювання задачі (1), (2). Для обчислення одинарних інтегралів по границі області, визначений клас *граничний елемент* (TUBound), який також наслідує клас *базисних функцій* (TUInterpolation) (рис. 3).

Окремо запрограмований клас (TUInterpolation) містить віртуальні процедури обчислення базових інтерполяційних функцій, а також похідних за коефіцієнтами. Інтегральні характеристики скінченних елементів, зокрема матриці жорсткості обчислюється інтегруванням за елементом. Для цього клас (TUInterpolation) містить методи інтегрування заданої змінної матриці в області, яка має канонічну форму у локальних координатах. Можуть бути утворені похідні класи для інтегрування квадратур для областей різної форми.

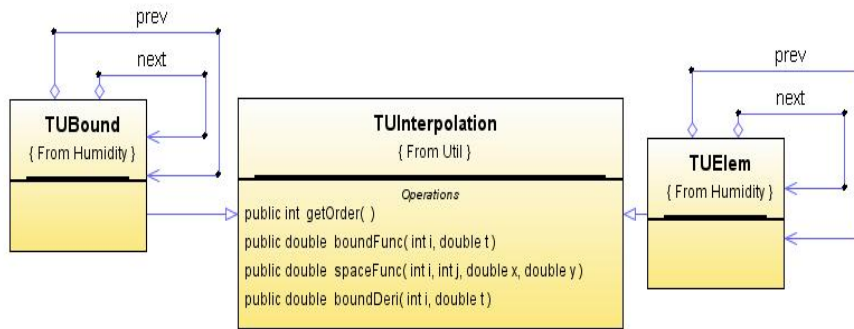


Рис. 3. Зв'язок узагальнення між базисною функцією та скінченним елементом

Скінченні і граничні елементи будуються на певній сітці вузлів, якою покривається область задачі. Постає питання, як реалізувати зв'язок між елементом та вузлами, на яких він побудований. Відомі програмні реалізації методу скінченних елементів передбачають використання масивів різних розмірностей. Зокрема перший масив зберігав усі вузли (їхні номери, координати тощо), другий встановлював відповідність номерів скінченних елементів номерам вузлів, що їм належать, третій зберігав номери вузлів, що знаходяться на границі. Це вимагало значних зусиль при написанні та відлагодженні програмного коду, узгодженості за відповідністю розмірностей та індексів масиву.

У цій програмній реалізації МСЕ для збереження усієї множини вузлів, які накладаються на область задачі, використано (6) двозв'язний замкнений список. (Кожен елемент списку містить вказівники на попередній та на наступний елементи (перші та останній, а останній на перший) та будь-які власні дані. Вказівники на початок та на поточний елемент списку дають змогу здійснювати всі потрібні дії: перемішуватись за елементами, додавати або видаляти їх.

Здійснити обхід елементів списку (незалежно від їхньої кількості) не складніше ніж перебрати елементи масиву. Наступна частина коду (мовою програмування Java) перебирає усі елементи певного списку і уможливорює використання методу calculateMatrice() для кожного елемента:

```

elems.setHead();
do {
elems.getCurr().calculateMatrice();
} while (elems.move(1)!=elems.getHead());
  
```

Шукані вузлові значення окремо виділено в клас (TUValues). Отже, кожен вузол (TUNode) – це точка (Point) з певними координатами. Він містить вказівники на попередній та наступний вузли та масив структур вузлових значень (TUValue) для різних моментів часу (рис. 4).

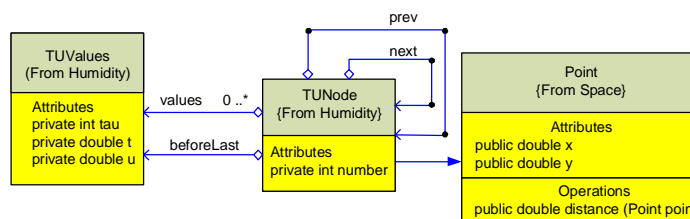


Рис. 4. Відношення між класами, що зберігають вузли і вузлові значення

Збереження множини скінченних та граничних елементів також організовано на основі двозв'язних замкнених списків. Скінченні та граничні елементи містять вказівники на вузли, що їм належать, а отже, звертаються напряму до своїх вузлів.

Зображені на рис. 5 класи відображають лише фрагмент прикладної програми, а саме структуру, необхідну для зберігання розбиття області, вузлових значень та базисних функцій,

зберігання компонент переміщень, деформацій та напружень. Для реалізації решти етапів, які потрібно здійснити згідно зі загальним алгоритмом МСЕ, запрограмовано класи, які організують: зберігання матриць та розв'язування СЛАР, чисельне інтегрування на основі квадратурних формул Гаусса та графічний інтерфейс користувача. Для здійснення операцій над векторами та матрицями, розв'язування систем лінійних алгебраїчних рівнянь було використано пакет класів *Jama* [17].

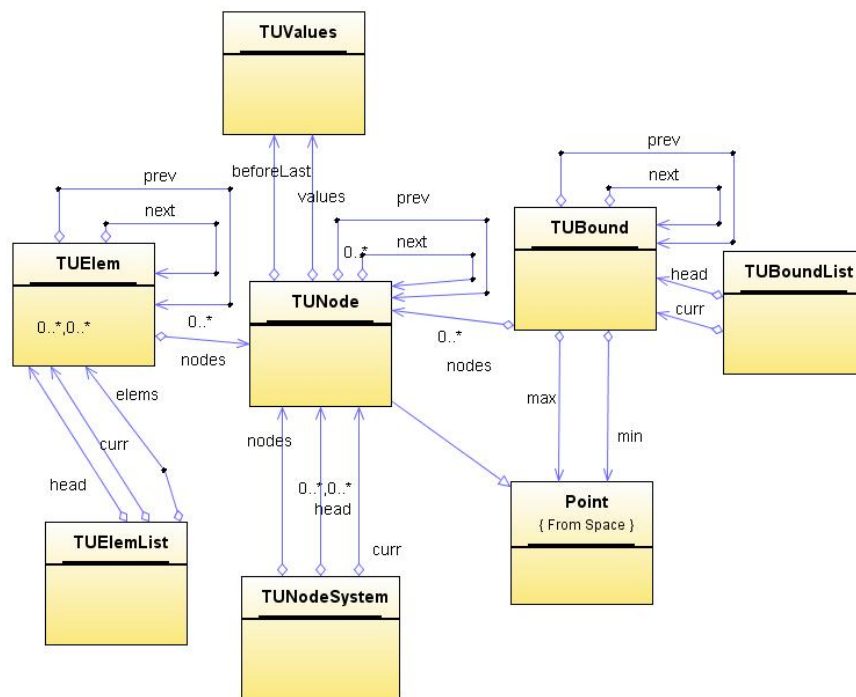


Рис. 5. Діаграма класів реалізації розбиття області

Для проектування послідовності взаємодії об'єктів в обчислювальному процесі використано діаграми послідовностей UML. Графічний інтерфейс дає можливість одночасно створювати декілька задач (до 4-х) з різними умовами, розв'язувати їх та візуалізувати інформацію у вигляді графіків. Під час створення задач необхідно побудувати об'єкти класів, які описані на рис.1. Конструктори відповідних класів задають значення початкових умов за замовчуванням та створюють діалогові вікна, в яких користувач може змінювати умови задачі.

У процесі обчислення створюється об'єкт класу *TURezult*, який, взаємодіючи із структурою елементів та вузлів, дає змогу апроксимувати шукані величини у невузлових точках. Визначення компонент напружень на новому часовому кроці згідно з математичною моделлю [15, 16] вимагає наявності розподілів величин $Q_{1,2}$ і $Q_{1,3}$. Значення $Q_{1,2}$ і $Q_{1,3}$ визначаються із розв'язання задачі масоперенесення у різних фазах у середовищі. Оскільки величини ітерацій у задачах визначення $Q_{1,2}$ і $Q_{1,3}$ та знаходження компонент напружень можуть не збігатися, то перед обчисленням напружень на наступному часовому кроці необхідно мати можливість здійснити необхідну кількість ітерацій визначення величин $Q_{1,2}$ і $Q_{1,3}$.

Числовий експеримент

Знайдемо використання програмної системи для розрахунку та аналізу в'язко-пружного стану деревини на основі моделі [15, 16]. Необхідні для розрахунку емпіричні залежності фізичних властивостей деревини з базовою густиною $\rho = 530 \text{ кг/м}^3$ наведено у роботі [5]. Параметри зовнішнього середовища вважаються сталими: температура середовища $T_c = 60^\circ \text{C}$; відносна вологість $j = 70\%$. Початкові температура і вологість відповідно дорівнюють $T_0 = 20^\circ \text{C}$, $u_0 = 0,3$. Пружні властивості матеріалу: модулі пружності $C_{11} = 400 \text{ МПа}$; $C_{12} = 220 \text{ МПа}$;

$C_{22} = 1040$ МПа; коефіцієнти Пуассона $\nu_{11} = 0,55$, $\nu_{12} = 0,35$. Коефіцієнти механіко-сорбційної повзучості: $m_{11} = 0,15$ МПа⁻¹; $m_{12} = 0,2$ МПа⁻¹; $m_{22} = 0,1 \cdot 10^{-3}$ МПа⁻¹.

Експериментальні значення реологічних властивостей матеріалу визначали згідно з [18]. На рис. 6б показано динаміку зміни напруженого стану у матеріалі з розмірами 0,2x0,02 м. Рис. 6а схематично зображає матеріал (у силу симетрії задачі відображено лише четверту частину). Механізм формування нормальних напружень на поверхнях A_1A_2 та A_2A_3 є різним.

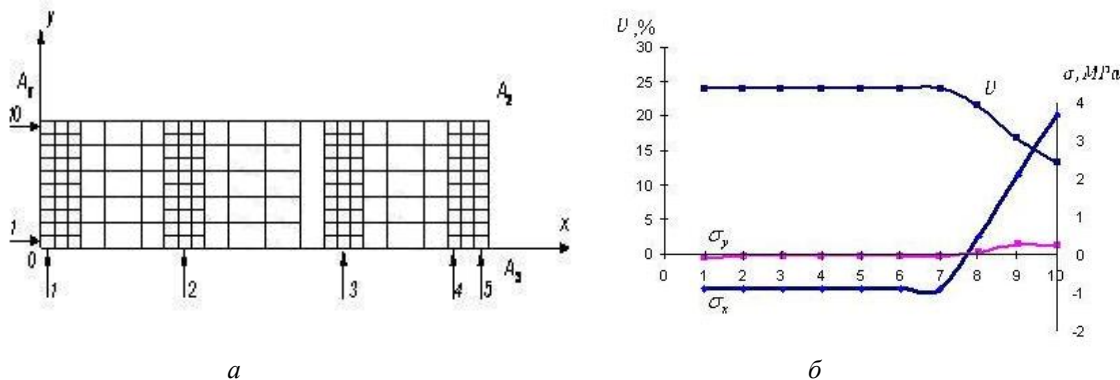


Рис. 6. а – схематичне зображення об'єкта;
б – зміна напружень у матеріалі для різних моментів часу і вологості

Аналіз чисельного експерименту зіставлення результатів розрахунків з результатами відомих одновимірних моделей свідчить про те, що розроблене програмне забезпечення дає змогу з необхідною точністю здійснювати розрахунок та аналіз в'язко-пружного стану капілярно-пористих матеріалів.

Важливим є те, що за об'єктно-орієнтованим підходом можна отримувати програмну систему відкритої структури. Це дає змогу використовувати його для чисельного моделювання задач різного фізичного змісту.

Висновок

У концепціях об'єктно-орієнтованого підходу та нотації UML розроблене програмне забезпечення для чисельного моделювання та аналізу в'язко-пружного стану капілярно-пористих матеріалів. Створене програмне забезпечення складається із задокументованих класів, які відповідають концепції застосування МСЕ і можуть бути повторно використані для реалізації інших фізично подібних моделей з використанням методу скінченних елементів.

1. Forde, B. *Object-oriented finite element analysis*/ B. Forde, R. Foschi, S. Steimer// *Computers and Structures*. – 1994. – Vol. 34, no. 3. – Pp. 355 – 374. 2. Zeglinski, G.W. *Object-Oriented Matrix Classes For Use In A Finite Element Code Using C++*/ G.W. Zeglinski, R.P.S. Han, A.P.// *International Journal For Numerical Methods In Engineering*. – 1994. – Vol. 37, no. 22. – Pp. 3921 – 3937. 3. Zimmermann, T. *Object-Oriented Finite Element Programming: I. Governing Principles*/ T. Zimmermann, Y. Dubois-Pelerin, P. Bomme// *Computer Methods in Applied Mechanics and Engineering*. – 1992. – Vol. 98, no. 2. – Pp. 291 – 303. 4. Dubois-Pelerin, Y. *Object-Oriented Finite Element Programming: III. An Efficient Implementation in C++*/ Y. Dubois-Pelerin, T. Zimmermann// *Computer Methods in Applied Mechanics and Engineering*. – 1993. – Vol. 108, no. 1 – 2. – Pp. 165 – 183. 5. Соколовський Я.І. *Моделювання деформаційно-релаксаційних процесів у висушуваній деревині методом скінченних елементів*/ Я.І. Соколовський, А.В. Бакалець// *Вісник Нац. ун-ту «Львівська політехніка»: Комп'ютерні науки та інформаційні технології*. – 2006. – Вип. 565. – С. 51 – 57. 6. Соколовський Я.І. *Математичні та програмні аспекти моделювання та оптимізації гідротермічних процесів*/ Я.І. Соколовський, А.В. Бакалець// *Комп'ютерні науки та інформаційні технології CSIT-2009: IV міжнародна науково-технічна конференція*. – Львів: НУ «Львівська політехніка», 2009. – С. 141 – 144. 7. Бакулин В.Н. *Объектно-ориентированная реализация метода конечных элементов*/ В.Н. Бакулин, В.О. Каледин, Вл.О. Каледин// *Математическое моделирование*. – 2003. – Т. 15. №2 – С. 77 – 82. 8. Альес М.Ю. *Построение и адаптация конечно-элементной сетки при решении эллиптической задачи второго*

порядка/ М.Ю. Альес, С.П. Копысов, А.К. Новиков// Математическое моделирование. – 1997. – Т. 9, №2. – С. 43 – 45. 9. Копысов, С.П. Комбинированное уточнение по p – и h -версии МКЭ при решении трехмерных задач теории упругости/ С.П. Копысов, С.Л. Устюжанин // XIII Всероссийской конференции посвященной памяти К.И. Бабенко «Теоретические основы и конструирование численных алгоритмов для решения задач математической физики». – Пуццоно: 2000. 10. Zienkiewicz O.C. The Finite Element Method: Its Basis and Fundamentals/ Zienkiewicz O.C., Taylor R.L., Zhou. Z.// Elsevier. Oxford, 6-th Edition, 2005. – 792 p. 11. Коваль В. Генерація адаптивних сіток з чотиризначних елементів у скінченно-елементному аналізі/ В. Марал, Я. Савула// Вісник Львівського національного університету ім. І. Франка, серія «Прикладна математика та інформатика». – 2002. – №10. – С. 64 – 71. 12. Макара В. Автоматизована числова схема розв'язання крайових задач на основі h -адаптивного методу скінченних елементів/ В. Макара, О. Куліш// Вісник національного університету «Львівська політехніка», 2011, №694, стор. 406 – 413. 13. Копысов С.П. Реализация объектно-ориентированной модели метода безконечных областей на основе параллельных распределённых элементов// С.П. Копысов, И.В. Краснощёров, В.Н. Рычков// Вычислительные методы и программирование. – 2003. –Т.4., №1. – с.194 – 206. 14. Савула Я.Г. Числовий аналіз задач математичної фізики варіаційними методами/ Я.Г. Савула. – Львів: ВЦ ЛНУ ім. І. Франка, 2004. – с. 222. 15. Sokolowskij Ya. Mathematical simulation of deformational and relaxation process in capillary-porous materials/ Ya. Sokolowskij, O. Mokritska// Forestry, Forest, Paper and Woodworking Industry, 2010, №36. p.116 – 124. 16. Соколовський Я.І. Математична модель в'язко-пружного деформування капілярно-пористих матеріалів/ Я. Соколовський, О. Мокрицька// Науковий вісник: Збірник науково-технічних праць. – Львів, НЛТУ України, 2011, Вип. 21.2. стр. 320 – 328. 17. JAMA: A Java Matrix Package [електронний ресурс]. – <http://math.nist.gov/javanumerics/jama/> 18. Соколовський Я.І. Методика та результати експерименту нових досліджень реологічної поведінки деревини/ Я. Соколовський, Й. Андреняк/ Науковий вісник: Збірник науково-технічних праць. – Львів, УкрДЛТУ. – 1999. – Вип. 9.13. – С. 15 – 26.

УДК 531.36+534

І. Дронюк

Національний університет “Львівська політехніка”,
кафедра автоматизованих систем управління

МОДЕЛЮВАННЯ ЗБУРЕНИХ КОЛИВНИХ СИСТЕМ ЗАСОБАМИ АТЕВ-ФУНКЦІЙ

Ї Дронюк І., 2011

Здійснено моделювання періодичних процесів у коливних системах на основі Ateb-функцій без збурень. Розглянуто типи малих збурень, які діють на коливну систему. Розроблено відповідне програмне забезпечення, яке подає результати у графічному і табличному вигляді.

Ключові слова: Ateb-функції, коливні системи з малим збуренням, моделювання.

A simulation of periodic processes in vibration systems based on Ateb-functions without perturbation was done. The types of small perturbations that act on the vibration system are considered. Appropriate software that provides results in graphical and tabular form is developed.

Keywords: identification, Ateb-function, oscillating system with small perturbation, simulation.

Вступ

Методи математичного моделювання коливних процесів ґрунтуються на диференціальних рівняннях. Побудова точних розв'язків ґрунтується на асимптотичних методах та теорії Ateb-