Якщо концентрація лімфоцитів у певних точках (скоріше за все – оптимальних) перевищує наперед задану, то деякі В-ліфмоцити стають супресорами і на наступних ітераціях призведуть до зменшення концентрації. Це один з інструментів регулювання популяції клітин в системі.

В результаті ітеративної репродукції лімфоцитів та їх реогранізації в просторі отримуємо певні групи клітин, кожна з яких відповідає точці оптимума.

Однією з переваг такого підходу є можливість навчання. Наприклад, після знаходження оптимума навчальної функції антитіла зосередяться в певних ділянках і це їх розміщення і буде слугувати первинною популяцією при роботі з іншими функціями. Це прискорить подальші пошуки і зробить їх більш ефективними, якщо навчальна функція відтворює поведінку наступних тестових функцій.

Висновок

Запропонований імунний метод є ефективним засобом для паралельного пошуку оптимумів функцій, зокрема мультимодальних. Механізми підтримки різноманіття векторів рішень та навчання забезпечують більш ефективний пошук у порівнянні з генетичним підходом. Метод відрізняють важливі властивості імунної теорії, а саме репродукція кандидатів на всьому просторі рішень, саморегуляція, масова генерація напрямів пошуку та пошук навколо найбільш оптимальних точок.

[1] D. Dasgupta, "Artificial Immune Systems and Their Applications," Springer-Verland, Inc., Berlin, Jan. 1999.

Алгебро-алгоритмічні засоби побудови клітинкових автоматів та застосування їх до розв'язку початково-крайових задач

Любов Захарія, Олег Захарія

Кафедра інформаційних систем та мереж, Національний університет "Львівська політехніка", УКРАЇНА, м.Львів, вул.С.Бандери, 12, E-mail:zlm.lviv@gmail.com

Інститут програмних систем НАН України, 03187, Київ, проспект Академіка Глушкова, 40., E-mail:ozakhar@gmail.com

***Abstract* – This work deals with the cellular automaton that was build using algebra-algorithmic facilities for solving linear boundary problems. As an example was taken "prey-predator" model which describes correlation of predators and victims populations. Obtained results was compared to solutions of Lotka-Voltera model, that is corresponding to "prey-predator" model, that have been received by using mathematical methods.**

Ключові слова – Cellular automaton, prey-predator model, algebra-algorithmic facilities.

I. Вступ

Розвиваються новітні технології, особливо в сфері програмування та моделювання математичних, фізичних та біологічних процесів. Серед напрямків дослідження найбільш поширеними є методи математичного, комп'ютерного, генетичного моделювання та інші. Великою перевагою математичних моделей є те, що вони чітко відображають залежність між основними факторами досліджуваного процесу. В той же час, такі моделі є досить чутливими до складності поставленої задачі, трапляються випадки, коли, навіть, чисельними методами дуже важко знайти її наближений розв'язок і цей процес потребує великих затрат ресурсів.

До методів комп'ютерного моделювання належать механізми клітинкових автоматів та штучних нейронних мереж. Поширенню цих методів сприяла їх стійкість до складності задачі, а оскільки більшість природних процесів є нелінійними і важко піддаються розв'язанню методами математичного моделювання, то ця їх особливість є великою перевагою при дослідженні. Механізм клітинкових автоматів можна використовувати не тільки при дослідженні чи моделюванні певних математичних, фізичних, біологічних чи інших процесів, одним із можливих їх застосувань в інформатиці є шифрування, а також стиснення даних. В той же час клітинкові автомати можуть описувати складні топології паралельних процесів, що є дуже важливим для дослідження алгоритмічних засобів моделювання задач предметних областей. Зокрема лінійний клітинковий автомат був використаний В.М. Глушковим для формалізації синхронних паралельних процесів.

В роботі алгеброалгоритмічними засобами будується клітинковий автомат розв'язання лінійних крайових задач, що описує модель –«жертва-хижак», результати застосування клітинкового автомату порівнюються з результатами моделі Лотки-Вольтера, що описує взаємозв'язок між популяціями хижаків та жертв.

При дослідженні даної системи таке моделювання має відчутні переваги, оскільки саму взаємодію між особинами дозволяє відтворити більш наближено до природної, ніж за допомогою математичних методів. Таку гнучкість конструювання клітинковим автоматам надає те, що вони враховують розміщення кожної особини ареалу, а також їх рух, чого не враховує відповідна математична модель. Серед недоліків можна назвати складність вибору правил сусідства так, щоб отримана таким чином система відповідала моделі Лотки-Вольтера. Щоб реалізувати цю аналогію, кожному з коефіцієнтів математичної моделі було поставлено у відповідність певний елемент управління в програмі. Таким чином, при зміні певних коефіцієнтів, щоб подивитись які зміни відбудуться в популяціях не потрібно розв'язувати диференціальні рівняння, що значно полегшує процес дослідження. Це є прикладом однієї з переваг клітинкових автоматів, а саме стійкості до складності задачі.

II. Постановка задачі

Система Лотки-Вольтера описується наступним чином:

$$\begin{aligned} \frac{dx}{dt} &= k_1 x - k_2 xy \\ \frac{dy}{dt} &= -k_3 y + k_4 xy \end{aligned} \qquad (1)$$

де k_1, k_2, k_3, k_4 – коефіцієнти моделі.

В даній моделі «хижак-жертва» припускають, що на одній території проживають два види тварин, один з них – це хижаки, а інший – це жертви.

Популяції хижаків ставлять у відповідність y(t), а популяції жертв – x(t). Враховуючи це, можна пояснити суть коефіцієнтів математичної моделі. Припускаючи, що харчів для жертв є достатньо, а також що c_1 – це рівень народжуваності жертв, а c_2 – їх рівень природної смертності. Тоді $k_1 x = (c_1 - c_2)x$ буде відображати приріст популяції жертв. Вираз $k_2 xy$ відповідає зменшенню чисельності жертв, за рахунок вдалого полювання хижаків. Таким чином перше рівняння системи (1) повністю відображає зміни в популяції жертв. Аналогічно, рівень народжуваності хижаків залежить від кількості доступної їди, тобто жертв, тоді якщо припустити, що k_3 – рівень смертності хижаків, а $k_4 x$ – рівень народжуваності хижаків, ми отримаємо друге рівняння системи (1), яке повністю відображає зміни в популяції хижаків.

Метою роботи є створення клітинкового автомату з такими правилами сусідства, що давали б результат, близький до математичного розв'язку, така постановка дозволить визначити ті межі, за рамками яких клітинковий автомат набагато точніше представляє модель у порівнянні з математичним методом.

Для побудови клітинкового автомату моделювання задачі «хижак-жертва» використовувались системи алгоритмічних алгебр Глушкова, що дозволяють формалізувати опис алгоритму алгебраїчними засобами.

Система яка складається з пари алгебри:, алгебри операторів і алгебри умов, називається системою алгоритмічних (мікропрограмних) алгебр.

Основами САА при описі алгоритмів є безліч операторів і умов, при описі структур даних - безліч об'єктів і умов.

Оператор – відображення інформаційної множини М в себе (A:M→M), що здійснює перетворення М з одного стану в інший.

Умова – предикат, визначений на М, що набуває значення істина, фальш або невизначеність (позначимо їх відповідно символами 1,0,m), тобто умова - це відображення вигляду a:M→{0,1,m}, що характеризує поточний стан інформаційної множини М.

Об'єкт – Визначений набір конфігурацій інформаційної множини М.

Операції САА можна класифікувати на наступні 3 типи:

Тип 1. На множині умов САА визначені операції логіки - диз'юнкція, кон'юнкція і заперечення, операції лівого і правого множення оператора (об'єкту) на умову.

Тип 2. На множині операторів визначені операції композиції, α-диз'юнкції, α-ітерації, синхронної і асинхронної диз'юнкції, фільтрації і синхронізації.

Тип 3. На множині об'єктів визначені ті ж операції, що і для операторів, які в разі опису структур даних інтерпретуються в термінах використання об'єкту в режимах розпізнавання або породження. Режим розпізнавання використовується для верифікації дослідження відповідності реальних вхідних, проміжних і результуючих даних тим структурам даних, які передбачалися при проектуванні програми. У режимі породження здійснюється опис необхідної структури даних, що відображується потім в мові програмування, а також породження конкретних даних описаної структури, які будуть використані як тести при відладці програм.

Регулярною схемою (РС) в САА, яка описує алгоритми, називається суперпозиція операції перших двох типів над множиною елементарних операторів і умов.

Об'єктною регулярною схемою (ОРС) в САА називається суперпозиція операцій першого і третього типів над множиною елементарних умов і об'єктів.

Незалежно від вибраної стратегії проектування програм (від даних до алгоритму, від алгоритму до даних або при одночасному проектуванні даних і алгоритмів) необхідно вказувати в алгоритмі динамічний зв'язок між даними і операторами їх обробки. Це можна здійснити за допомогою параметрів, що вказуються в операторах, або за допомогою переміщення вказівників і встановлення/видалення маркерів в структурі даних, які в кожен момент часу вказують місце в цій структурі даних, де буде використовуватись конкретний оператор.

САА/М=<A, B, O; W>,

де A, B, O – основи алгебри, тобто множини операторів, умов і об'єктів відповідно, а W – сигнатура операцій над A, B, O.

Операції диз'юнкції "∨", кон'юнкції "∧" і заперечення "¬" визначаються відомими таблицями істинності над логічними змінними. Множину операторів складають невизначений оператор N, тотожний оператор E та набір реалізованих в мові програмування базових (елементарних) для даної предметної області операторів. Невизначений оператор N перетворить будь-який стан М в спеціальний стан w∈M, що називається невизначеним. Тотожний оператор довільний стан m∈M перетворить в себе.

Операції композиції операторів, α-диз'юнкції, α-ітерації складають перший набір операцій структурного програмування, необхідний для представлення будь-якого послідовного алгоритму і відповідають послідовному виконанню операторів, оператору організації розгалужень та циклічному оператору.

Для формалізації синхронних паралельних алгоритмів в САА введена операція A $\vee$ B синхронної диз'юнкції операторів A та B, що забезпечує виконання обох операторів A та B, якщо вони збігаються, або одного з них, якщо інший в цей час не визначений. У всіх інших випадках результатом синхронної диз'юнкції буде невизначений оператор.

Для формалізації асинхронних паралельних алгоритмів введена операція A+B асинхронної диз'юнкції операторів A та B, що полягає в паралельному виконанні операторів A та B на операційних підструктурах, що не перетинаються. Для забезпечення узгодженої взаємодії паралельних процесів введені синхронізатори і фільтри.

Синхронізатор по умові α - це оператор виду

$$S(a)= \alpha\{E\},$$

де E - тотожній оператор. Він призначений для затримки виконання по одній з паралельних гілок до виконання умови α, яке називається умовою синхронізації. Дуже часто, в ролі умов синхронізації використовуються умови, що змінюють свої значення після досягнення програмою певних місць в ній, так званих контрольних точок.

III. Реалізація.

Випадковим чином ініціалізується матриця M, розміру n*n, що відповідає полю клітикового автомату і кожна комірка якої може приймати значення 0, 1 або 2, кожне з цих значень визначає наявність в клітинці автомату хижака, жертви, або є вільною.

Алгоритм переходу клітинкового автомату від стану α_i до стану α_{i+1}

Для того, щоб задовольнялось одне з основних правил клітинкового автомату, а саме, те що зміна значень усіх клітинок має відбуватись одночасно, тому введемо 2n різних вказівників, що будуть проходити по полю клітинкового автомату для реалізації такого синхронного паралелізму

M: Y_{k1} поставимо в початкову позицію M(0,0)

Y_{k2} поставимо в початкову позицію M(1,0)

Y_{kn} поставимо в початкову позицію M(n,0)

Y_{z1} поставимо в початкову позицію M(0,1)

Y_{z2} поставимо в початкову позицію M(1,1)

Y_{zn} поставимо в початкову позицію M(n,1)

Обчислюємо значення першого стовпця матриці, але не затираємо існуючі його значення до тих пір поки не будуть обчисленні нові значення другого стовпця матриці. Коли всі значення другого стовпця обчислені, затираємо значення першого стовпця і переводимо вказівники $Y_{kj},\ (j=1..n)$ на 2 комірки вправо і обчислюємо значення третього стовпця матриці. Обчисливши значення третього стовпця, ми можемо затерти старі значення другого стовпця і перевести вказівники $Y_{zj},\ (j=1..n)$ на дві комірки вправо. Таким чином ми перейдемо по всьому полі клітинкового автомату, при цьому дотримуючись правила, про одночасну зміну значень всіх сусідніх клітинок. При обчисленні нових значень клітинок, будемо використовувати процедуру V_1, яка в залежності від існуючого значення клітинки і правил сусідства визначає нове значення клітинки і процедуру V_2, яка замінить існуючі значення на нові після обрахування наступного стовпця. Процедура V_3 буде здійснювати переміщення вказівника на дві комірки вправо.

В термінах САА-схем це можна записати так:

$$\underbrace{\mathrm{C}(T_{k1})*V_1*\quad V_2*V_3*V_1*T_{k2}\quad //\ldots//\quad \mathrm{C}(T_{k1})*V_1* *V_2*V_3*V_1*T_{k2}//}_{}$$

Для вказівників $Y_{kj},\ (j=1..n)$

$$\underbrace{//V_1*T_{k1}*\mathrm{C}(T_{k2})*V_2*V_3//\ldots//V_1*T_{k1}*\mathrm{C}(T_{k2})* *V_2*V_3}_{},$$

Для вказівників $Y_{zj},\ (j=1..n)$

де T_{k1} і T_{k2} - контрольні точки. C(u) – синхронізатор.

Наведений вище алгоритм побудови клітинкового автомату був використаний для написання двох програм, що реалізовують два клітинкові автомати з різними правилами сусідства.

При написанні даної роботи було написано дві програми, в яких було побудовано і реалізовано два клітинкових автомати, які за своєю поведінкою можна віднести до автоматів четвертого класу, оскільки отримані результати в повній мірі залежать від початкових умов, а також від особливостей правил сусідства.

Розглянемо послідовно кожен з цих автоматів. Перший автомат було побудовано як аналог математичної моделі, але особливістю його було те, що правила сусідства вибирались таким чином, щоб вони відповідали реальним процесам, що відбуваються в природі. Для того щоб отримані результати можна було порівняти з результатами математичної моделі, в програму були добавлені компоненти, що корегують правила сусідства автомату, і тим самим змінюють отриманий результат, відповідно до того, як коефіцієнти $k_1 - k_4$ впливають на результат математичної моделі. Виходячи з цього, перший клітинковий автомат було побудовано з такими правилами сусідства:

в кожній клітинці поля може бути не більше однієї особини певного виду, кожна з особин випадковим чином переміщається по полю на одну клітинку,

якщо в полі зору хижака, що визначається ймовірністю зустрічі, з'являється жертва і при умові, що хижак вбив не більше ніж дозволено жертв, він її вбиває,

приріст популяції жертв та рівень смертності хижаків встановлюються відповідно до значень коефіцієнта приросту жертв та коефіцієнта смертності хижаків відповідно,

приріст популяції хижаків залежить від кількості вбитих жертв.

Коефіцієнт, що в моделі Лотки-Вольтера відповідає за успішність полювання хижаків, в даній програмі представлений як математичне сподівання зустрічі жертви з хижаком. Всі інші коефіцієнти мають чіткий аналог у вигляді певного елемента управління в програмі.

Крім того, ще в програмі додано елемент управління, який відповідає за апетит хижака, тобто за максимальну кількість жертв, яку може вбити хижак за одне полювання, таким чином отримано більш-менш однорідну ситуацію при моделюванні.

Такий вибір правил дає можливість досить близько відтворити за допомогою клітинкових автоматів розв’язок моделі Лотки-Вольтера і співставити його з відповідним математичним розв’язком цієї моделі.

Другий побудований клітинковий автомат має давати результати якомога ближчі до математичного розв’язку моделі, тому правила сусідства будуть визначатися коефіцієнтами математичної моделі, щоб мати змогу порівняти отримані результати. Якщо проаналізувати вплив коефіцієнтів $k_1 - k_4$ математичної моделі на зміни в популяціях хижаків і жертв, то легко побачити, що смертність жертв і приріст хижаків напряму залежать від математичного сподівання їх зустрічі. Виходячи з цього, при реалізації клітинкового автомату в програмі було додано два поля, за допомогою яких можна коригувати значення математичного сподівання зустрічі. В цих полях задаються значення, які визначають розміри області сусідства для особин кожної з популяцій, хижаків чи жертв, окремо. Тобто для кожного індивідуума, в залежності від величини області, обчислюється кількість хижаків і жертв в цій області і, відповідно до отриманих значень та коефіцієнтів $k_1 - k_4$ визначається критична кількість особин тої чи іншої популяції, перевищення якої призводить до змін в чисельності популяцій, іншими словами, якщо розглянути жертву, і в певній області навколо неї знаходиться більше хижаків ніж обчислене критичне значення, то особина «помирає». Аналогічна ситуація і з особиною з популяції хижаків, якщо кількість жертв навколо є більшою ніж вищезгадана, критична кількість, то необхідно збільшити кількість хижаків.

Якщо ж проаналізувати, як змінюється приріст жертв і смертність хижаків у математичній моделі, то легко побачити, що вони залежать лише від загальної кількості особин в популяціях та відповідних коефіцієнтів моделі. Аналогічний механізм був реалізований і в клітинковому автоматі. Підраховується загальна кількість хижаків і жертв, і випадковим чином на поле клітинкового автомату додаються і видаляються відповідна кількість особин.

Такий вибір правил дає можливість чисельно порівняти результати, отримані за допомогою автомату, та відповідні результати математичного розв’язку моделі Лотки-Вольтера, і проаналізувати характер впливу кожного з коефіцієнтів моделі на результати.

В обох програмах також задається початкова кількість особин обох видів, що буде розподілена по всьому полю.

Початкове розміщення особин встановлюється рівномірно, відповідно до вказаної кількості хижаків та жертв. Нерівномірне розміщення призводить до різких змін чисельності популяцій, а саме, один з видів починає різко домінувати над іншим, що в деяких випадках призводить до вимирання одного з видів.

Програму, яка реалізує перший клітинковий автомат було побудовано в середовищі Borland C++ Builder 6, для швидкої роботи автомату було використано бітову матрицю, кожна комірка якої відповідає за окрему клітинку поля. Програма має дві форми, на одній формі (рис.1) розміщено всі елементи управління автоматом, а також відображено саме поле і графіки, що відображають співвідношення між популяціями хижаків та жертв, при кожній зміні правил сусідства графіки співвідношень малюються заново. Крім того є можливість переглянути на графіку, що зображає фазовий простір, зміни в популяціях впродовж певної кількості кроків, що в деякій мірі спрощує спостереження.

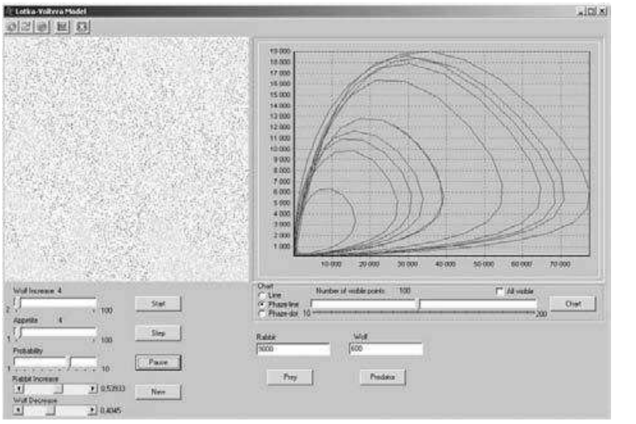

Рис. 1.

На рисунку 1 поряд з полем зображено графік фазового простору, що одержується при такому моделюванні. Шкала, зображена під графіком саме і визначає впродовж скількох останніх кроків треба відображати зміни в графіку фазового простору. На другій формі (рис.2) зображуються всі графіки від початку моделювання, завдяки чому можна подивитись як зміна того чи іншого параметру впливає на взаємозв’язок між популяціями обох видів.

Другу програму було написано на мові програмування C# в середовищі Microsoft Visual Studio 2005. Для графічного зображення результатів використано компоненти графічної бібліотеки ZedGraph. Вигляд основного вікна програми зображено на рис.3. Як видно з рисунка в програмі реалізована можливість редагувати всі вхідні дані, що використовуються при моделюванні, а також дані, необхідні для знаходження розв’язку методом Рунге-Кутта.

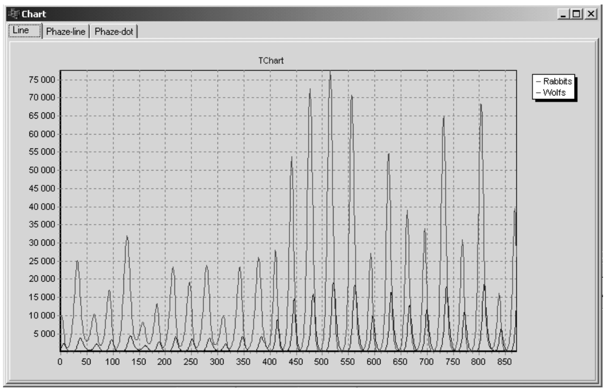

Рис. 2.

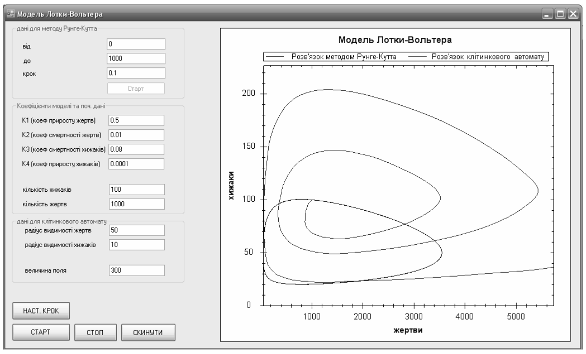

Рис. 3.

Результати, які отримані при розв'язуванні математичним методом і при моделюванні, зображаються кривими у фазовому просторі, що дає змогу легко проаналізувати їх і визначити які корективи необхідно внести до вхідних даних, щоб покращити отриманий результат.

Така структура програми дає змогу досить ретельно аналізувати процес моделювання.

IV. References

[1]. Г.Б. Астафьев, А.А. Короновский, А.Е. Храмов "Клеточные автоматы", Саратов 2003

[2]. Л. Наумов, А. Шалыто "Клеточные автоматы. Реализация и эксперименты",

[3]. M. Abdellaoui, A. El Jai "Cellular Automata Model for a Contact Problem"

[4]. А. Федотьев "Клеточный автомат" – http://rain.ifmo.ru/~fedotiev/.

[5]. Інтернет ресурс [http://en.wikipedia.org/]

[6]. Інтернет ресурс [math.fullerton.edu]

[7]. Глушков В.М., Цейтлин Г.Е., Ющенко Е.Л. «Алгебра. Языки. Программирование» Київ 1978

[8]. Цейтлін Г.О. «Теорія клонів та її додатки», Київ 2003

Evaluation of lower natural frequencies of axisymmetric vibrations of laminate circular plates clamped on the boundary

Andrzej Katunin

Faculty of Mechanical Engineering, Silesian University of Technology, Konarskiego Str., 18A, Gliwice, 44-100, POLAND, e-mail: andrzej.katunin@polsl.pl

In the paper the possibility of using Cauchy's influence function method in the case of material nonhomogeneity as layered structures was showed. For evaluation the lower natural frequencies of axisymmetric vibrations of laminate circular plates the double-sided estimators of Bernstein-Keropian tables and simplified analytical model of plates vibration were used, obtained results were compared with results obtained from finite element method. The difference growing between obtained results during frequency growing was noticed, but differences were in tolerance limit. Proposed algorithm of evaluating lower natural frequencies of axisymmetric vibrations can be used as approximative and can find an application in degradation degree evaluation of composite laminates.

Key words – clamped circular plates, laminate, Cauchy's influence function method.

I. Introduction

Modern engineering constructions used more frequently unconventional construction materials such as composites. Significant part of it is layered structures with polymer matrix like GFRP and CFRP laminates. Main advantage of polymer laminates is in its lightweight with simultaneously high durability properties, which creates a possibility to use it in most responsible applications applied in aircraft and aerospace industries, automotive industry, military industry and many others. The behavior of it must be predictable in all phases of exploitation and in research it must also take into consideration some additional phenomena (e.g. crack initiation and propagation [1,2], self-activating temperature [3], etc.).

One of the main criteria of degradation degree evaluation is dynamic behavior of the structures. Based on frequency responses it is a possibility to detect faults in layered structures using algorithm based on wavelet transform [4]. In this paper were investigated vibrations of clamped circular plate. This model can represent many responsible constructional elements like bottoms of cylindrical containers, rotors etc. For evaluation of natural frequencies of vibrations the Cauchy's influence function method and Bernstein-Keropian tables were used. Previous works shows, that proposed method gives good results for circular clamped plates with homogeneous and inhomogeneous materials [5,6] and additional properties like thickness variability [7,8] and additional masses [9]. In this paper the lower frequencies of axisymmetrical vibrations were investigated, which can find an application in degradation degree evaluation of layered structures.