|  |  |
| --- | --- |
| Змінна | Призначення |
|  | Точність з якою необхідно знайти корені розв’язків  () |
| Matrix | Матриця системи |
| b | Стовпець вільних членів |
| norm | Норма поточного розв’язку системи |
| flag | Ознака сходимості методу (false – метод не сходиться,  true – метод сходиться) |

Метод Якобі

1. ПОЧАТОК
2. Обчислити визначник матриці (*Matrix*) (**det**):
   1. ЯКЩО матриця має одиничну розмірність, ТО позначити як визначник даної матриці єдиний елемент цієї матриці. ІНАКШЕ обчислити визначник розкладом за першим стовпцем:

2.2.1 Ініціалізація визначника поточної матриці (*Matrix*) *det* = 0.

2.2.2 ЦИКЛ проходу по всіх елементах першого стовпця матриці (*Matrix*) :

2.2.2.1 Обчислити доповнювальний мінор для поточного елементу (**marixCut**):

2.2.2.1.1 Ініціалізація нової матриці (*newMatrix*) розмірності (n-1).

2.2.2.1.2 ЦИКЛ проходу по всіх елементах матриці (*Matrix*):

2.2.2.1.2.1 ЯКЩО поточний елемент не знаходиться на першому стовпці АБО   
(*i-тому*) рядку, ТО добавити його в нову матрицю (*newMatrix*).

2.2.2.2 Знайти визначник доповнювального мінору (тобто знайти визначник матриці *newMatrix*) (**det**) (аналогічно пункту 2).

2.2.2.3 Добавити до *det* добуток поточного елементу та визначник його доповнювального мінору.

1. Присвоїти *flag* = true.
2. ЯКЩО визначник не дорівнює нулю, ТО перевірити умови сходимості методу (**isSolution**):
   1. Приведення матриці до канонічного (діагонального) вигляду (**stableMatrix**):
      1. Ініціалізація масиву *element* (*element*[*i*] = *false*, *i* =1, 2, …, n).
      2. ЦИКЛ проходу по всіх рядках матриці (*Matrix*):
         1. Знайти номер індексу найбільшого за абсолютним значенням елементу поточного рядка (*index*).
         2. ЯКЩО такий індекс (*index*), вже був знайдений (*element[index]* = true), ТО для даної матриці (*Matrix*) метод не сходиться (присвоїти *flag* = false), ІНАКШЕ присвоїти *element[index]* = true.
      3. Ініціалізація нової матриці (*newMatrix*) такої ж розмірності, що й поточна матриця (*Matrix*).
      4. ЦИКЛ проходу по всіх рядках матриці (*Matrix*):
         1. Знайти номер індексу найбільшого за абсолютним значенням елементу поточного рядка (*index*) матриці (*Matrix*).
         2. Присвоїти поточний рядок матриці (*Matrix*) *index*-рядку матриці (*newMatrix*) (тобто рядку, який має номер *index*).
      5. Присвоїти матриці (*Matrix*) матрицю (*newMatrix*).
   2. ЦИКЛ проходу по всіх рядках матриці (*Matrix*):
      1. Знайти суму абсолютних значень всіх елементів поточного рядка (*sum*).
      2. Знайти найбільший за абсолютним значенням елемент поточного рядка (*max*).
      3. ЯКЩО *max(sum – max)*, ТО для даної матриці (*Matrix*) метод не сходиться (присвоїти *flag* = false).

ІНАКШЕ перейти до пункту 6.

1. ЯКЩО метод сходиться (), ТО розв’язати СЛАР (**Jacobi**):
   1. Задати попереднє наближення розв’язку (.
   2. Присвоїти *norm* = .
   3. ПОКИ :
      1. Обчислити наступне наближення (:
         1. Присвоїти .
         2. Помножити на .
      2. Обчислити поточне значення норми розв’язку (*norm*) (**normCalc**):
         1. Присвоїти *norm =* .
         2. ЦИКЛ проходу по всіх елементах поточного наближення :

4.3.2.2.1 ЯКЩО , ТО   
*norm =.*

* + 1. Присвоїти .

1. КІНЕЦЬ.

Метод Гауса-Зейделя

1. ПОЧАТОК
2. Обчислити визначник матриці (*Matrix*) (**det**) (аналогічно пункту 2 методу Якобі).
3. Присвоїти *flag* = true.
4. ЯКЩО визначник не дорівнює нулю, ТО перевірити умови сходимості методу (**isSolution**) (аналогічно пункту 4 методу Якобі), ІНАКШЕ перейти до пункту 6.
5. ЯКЩО метод сходиться (), ТО розв’язати СЛАР (**Seidel**):
   1. Задати попереднє наближення розв’язку (.
   2. Присвоїти *norm* = .
   3. ПОКИ :
      1. Обчислити наступне наближення (:
         1. ЦИКЛ проходу по всіх елементах поточного розв’язку (:
            1. Знайти суму:
            2. Знайти суму:
            3. Обчислити .
         2. Помножити на .
      2. Обчислити поточне значення норми розв’язку (*norm*) (**normCalc**) (аналогічно пункту 5.3.2 методу Якобі).
      3. Присвоїти .
6. КІНЕЦЬ.

Метод найшвидшого спуску

1. ПОЧАТОК
2. Обчислити визначник матриці (*Matrix*) (**det**) (аналогічно пункту 2 методу Якобі).
3. Присвоїти *flag* = true.
4. ЯКЩО визначник не дорівнює нулеві, ТО перевірити умови сходимості методу (**isSolved**):
   1. Перевірити матрицю на симетричність (**isSemetric**):
      1. ЦИКЛ проходу по всіх строках матриці (*Matrix*) (:
         1. ЦИКЛ проходу по всіх елементах поточної строки (

:

4.1.1.1.1 ЯКЩО, ТО для даної матриці (*Matrix*) метод не сходиться (присвоїти *flag* = false).

* 1. Перевірити матрицю на додатньовизначеність за критерієм Сильвестра (**isPositive**):
     1. ЦИКЛ перебору всіх кутових мінорів (:
        1. Ініціалізація нової матриці (*newMatrix*) розмірності .
        2. ЦИКЛ проходу по всіх елементах матриці (*Matrix*)   
           ():
           1. ЯКЩО для поточного елементу , ТО додати поточний елемент до нової матриці (*newMatrix*).
        3. Обчислити визначник нової матриці (*newMatrix*) (**det**) (аналогічно пункту 2 методу Якобі).
        4. ЯКЩО визначник нової матриці (*newMatrix*) від’ємний АБО дорівнює нулеві, ТО для даної матриці (*Matrix*) метод не сходиться (присвоїти *flag* = false).

ІНАКШЕ перейти до пункту 6.

1. ЯКЩО метод сходиться (), ТО розв’язати СЛАР (**GradientDecent**):
   1. Задати попереднє наближення розв’язку (.
   2. Задати попередню незв’язку як .
   3. Задати попередній вектор напрямку як .
   4. Присвоїти *norm* = .
   5. ПОКИ :
      1. Обчислити довжину для вектора напрямку
      2. Обчислити наступне наближення .
      3. Обчислити наступне значення незв’язки .
      4. Обчислити коефіцієнт
      5. Обчислити наступний вектор напрямку .
      6. Обчислити поточне значення норми розв’язку (*norm*) (**normCalc**) (аналогічно пункту 5.3.2 методу Якобі).
      7. Присвоїти .
      8. Присвоїти
      9. Присвоїти
2. КІНЕЦЬ.