Министерство образования Республики Беларусь
Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей
Кафедра информатики
Дисциплина: Методы численного анализа

ОТЧЁТ

к лабораторной работе №4
на тему:

«Решение систем нелинейных уравнений»

Выполнил: студент группы 353502
Згирская Дарья Денисовна
Проверил: Анисимов Владимир Яковлевич

Минск 2024

ЦЕЛИ ВЫПОЛНЕНИЯ ЗАДАНИЯ

- изучить численное решение систем нелинейных уравнений методами простых итераций и Ньютона;
- провести отделение решений, построить и запрограммировать алгоритмы методов, численно решить тестовое задание, сравнить трудоемкость методов.

ЗАДАНИЕ

Вариант 5. Решить систему нелинейных уравнений:

$$\begin{aligned} &tg(xy+m)=x \\ &ax^2+2y^2=1, \end{aligned} \quad \text{где } x>0,\ y>0,$$

с точностью до 0.0001 методами простых итераций и Ньютона, принимая для номера варианта k значения параметров a и m из таблицы:

k	1	2	3	4	5	6	7	8	9	10	11	12	13	14
m	0,0	0,1	0,1	0,2	0,2	0,3	0,3	0,4	0,4	0,1	0,2	0,3	0,2	0,2
a	0,5	0,6	0,7	0,8	0,9	1,0	0,5	0,6	0,7	0,8	0,9	1,0	0,7	0,5

Начальные приближения найти графически. Сравнить скорость сходимости методов.

ХОД РАБОТЫ

Перед началом выполнения работы были изучены теоретические материалы на тему «Решение систем нелинейных уравнений».

Далее были построены (рис. 1, 2) и запрограммированы алгоритмы методов (рис. 3, 4).

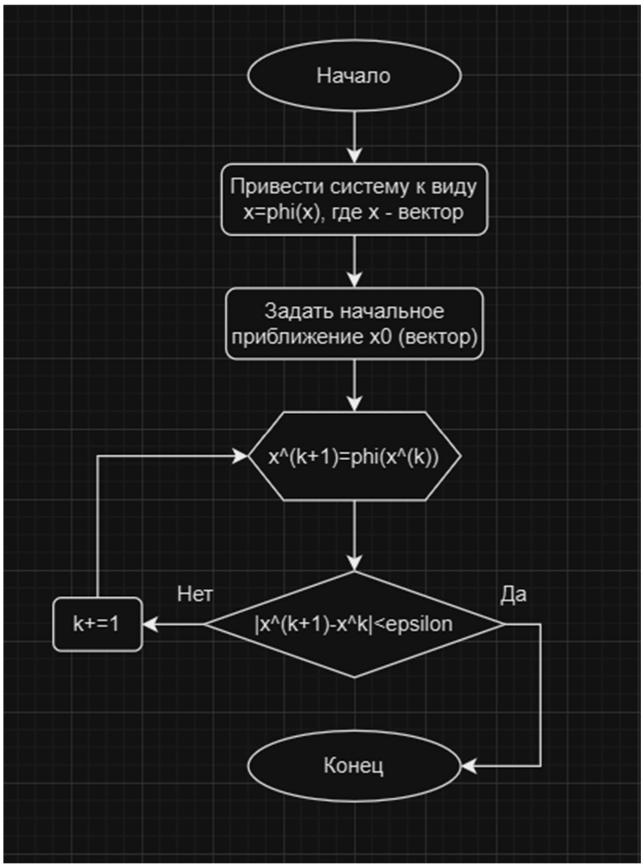

Рисунок 1 – Алгоритм метода простых итераций

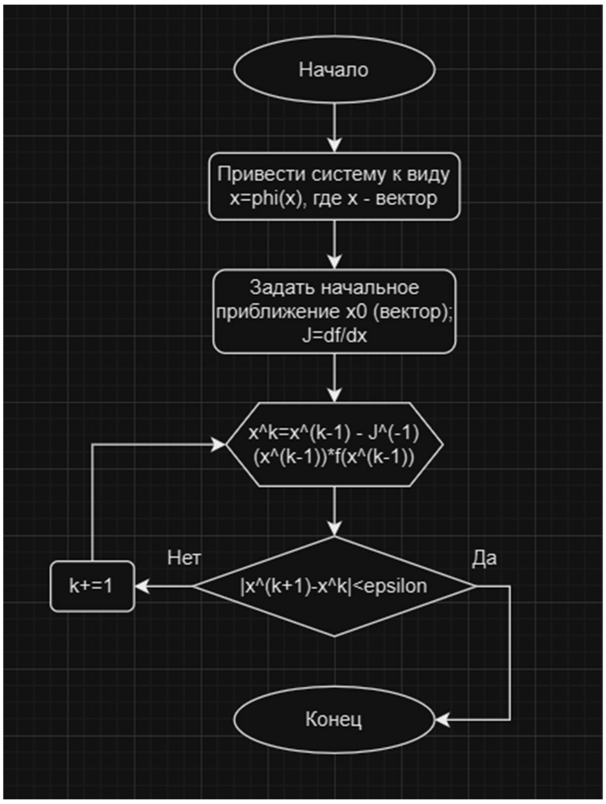

Рисунок 2 – Алгоритм метода Ньютона

```
> simple_iter_method := proc(SystemOfEquations, SystemOfAssumptions, precision)
    local i, j, k, count_of_eq, count_of_unknowns, tempEq, unknowns, PrevAssumptions, NewAssumptions, TEquations, less_then_prec;

    less_then_prec := 0;
    NewAssumptions := [op(SystemOfAssumptions)];
    TEquations := [op(SystemOfEquations)];
    count_of_eq := numelems(SystemOfEquations);
    count_of_unknowns := numelems(SystemOfAssumptions);

    unknowns := { };
    for i to count_of_eq do
      unknowns := unknowns union indets(SystemOfEquations[i], name);
    end do;

    for k to infinity do
       less_then_prec := 0;
       PrevAssumptions := NewAssumptions;

      for i to count_of_eq do
        tempEq := SystemOfEquations[i];
        for j to numelems(unknowns) do
          tempEq := subs(unknowns[j] = PrevAssumptions[j], tempEq);
        end do;
        NewAssumptions := subsop(i = evalf(tempEq), NewAssumptions);
      end do;

      print("Новое предположение:", NewAssumptions);

      for j to count_of_unknowns do
        if eval(abs(PrevAssumptions[j] - NewAssumptions[j])) < precision then
          less_then_prec := less_then_prec + 1;
        end if;
      end do;

      if less_then_prec = count_of_unknowns then
        print("Сходимость достигнута на итерации:", k);
        return NewAssumptions;
      end if;
    end do;
  end proc:
```

Рисунок 3 – Запрограммированный метод простых итераций

```
> newthon_method := proc(SystemOfEquations, SystemOfAssumptions, precision)
    local i, j, k, t, less_then_prec, J, TempValues, unknowns, NewAssumptions, PrevAssumptions, TempAssumptions, count_of_eq, count_of_unknowns, tempEq, AssumpDelta, rref_result;

    TempAssumptions := [op(SystemOfAssumptions)];
    PrevAssumptions := [op(SystemOfAssumptions)];
    AssumpDelta := [op(SystemOfAssumptions)];
    TempValues := [ ];
    NewAssumptions := [op(SystemOfAssumptions)];

    count_of_eq := numelems(SystemOfEquations);
    count_of_unknowns := numelems(SystemOfAssumptions);

    unknowns := [x, y];

    for t from 1 to infinity do
      J := eval(Jacobian(SystemOfEquations, unknowns), [x = NewAssumptions[1], y = NewAssumptions[2]]);
      print("Якобиан:", J);

      PrevAssumptions := NewAssumptions;

      TempValues := [ ];
      for k to count_of_eq do
        tempEq := eval(SystemOfEquations[k], [x = NewAssumptions[1], y = NewAssumptions[2]]);
        TempValues := [op(TempValues), evalf(-tempEq)];
      end do;

      # Преобразование TempValues в вектор для работы с матрицей Якобиана
      TempValues := convert(TempValues, Vector);

      rref_result := ReducedRowEchelonForm(⟨J | TempValues⟩);
      TempValues := convert(Column(rref_result, ColumnDimension(rref_result)), list);

      # Обновление предположений
      for j to count_of_unknowns do
        AssumpDelta[j] := TempValues[j];
        NewAssumptions[j] := NewAssumptions[j] + AssumpDelta[j];
      end do;

      print("Новое предположение:", NewAssumptions);

      less_then_prec := 0;
      for j to count_of_unknowns do
        if abs(PrevAssumptions[j] - NewAssumptions[j]) < precision then
          less_then_prec := less_then_prec + 1;
        end if;
      end do;
```

Рисунок 4 – Запрограммированный метод Ньютона

После чего было решено тестовое задание (рис. 5).

> *simple_iter_method*(*SystemOfEquations*, *SystemOfAssumptions*, *precision*);

"Новое предположение:", [0.3016828745, 0.5718566254]
"Новое предположение:", [0.3907643590, 0.6775281171]
"Новое предположение:", [0.5013835551, 0.6567240266]
"Новое предположение:", [0.5849376654, 0.6219940022]
"Новое предположение:", [0.6322946531, 0.5882444793]
"Новое предположение:", [0.6437144235, 0.5657663496]
"Новое предположение:", [0.6328046326, 0.5599413214]
"Новое предположение:", [0.6190835363, 0.5655097113]
"Новое предположение:", [0.6132397180, 0.5723032490]
"Новое предположение:", [0.6144255905, 0.5751275265]
"Новое предположение:", [0.6177558491, 0.5745576883]
"Новое предположение:", [0.6199178323, 0.5729484880]
"Новое предположение:", [0.6202564998, 0.5718967096]
"Новое предположение:", [0.6196220175, 0.5717314435]
"Новое предположение:", [0.6189783140, 0.5720409512]
"Новое предположение:", [0.6187342908, 0.5723544628]
"Новое предположение:", [0.6188095043, 0.5724731826]
"Новое предположение:", [0.6189706456, 0.5724365981]
"Новое предположение:", [0.6190669214, 0.5723581947]
"Сходимость достигнута на итерации:", 19
[0.6190669214, 0.5723581947]

> $eval(\tan(0.5723581947 \cdot 0.6190669214 + m));$

0.6190760159

> $eval\left(\left(\frac{(1 - a \cdot 0.6190669214 \cdot 0.6190669214)}{2}\right)^{\frac{1}{2}}\right);$

0.5723113367

> *newthon_method*(*SystemOfEquations*, *SystemOfAssumptions*, *precision*);

"Якобиан:", $\begin{bmatrix} -0.8363481164 & 0.6764277853 \\ 1.116 & 0.60 \end{bmatrix}$

"Новое предположение:", [0.795842207726009, 0.838000160296291]

"Якобиан:", $\begin{bmatrix} 1.00075046187426 & 1.90009708843476 \\ 1.43251597390682 & 3.35200064118516 \end{bmatrix}$

"Новое предположение:", [1.69831715264087, 0.161590191355725]

"Якобиан:", $\begin{bmatrix} -0.795792491949114 & 2.14622626974555 \\ 3.05697087475356 & 0.646360765422901 \end{bmatrix}$

"Новое предположение:", [1.09015691227335, 0.488114227982552]

"Якобиан:", $\begin{bmatrix} -0.117615444486486 & 1.97072235827285 \\ 1.96228244209204 & 1.95245691193021 \end{bmatrix}$

"Новое предположение:", [0.736229494386942, 0.564119323833364]

"Якобиан:", $\begin{bmatrix} -0.154010543782987 & 1.10409688747219 \\ 1.32521308989650 & 2.25647729533345 \end{bmatrix}$

"Новое предположение:", [0.623851586432417, 0.575035883941596]

"Якобиан:", $\begin{bmatrix} -0.200203016493377 & 0.867693009981173 \\ 1.12293285557835 & 2.30014353576639 \end{bmatrix}$

"Новое предположение:", [0.618982235399423, 0.572368092566800]

"Якобиан:", $\begin{bmatrix} -0.208310415304146 & 0.856165455834330 \\ 1.11416802371896 & 2.28947237026720 \end{bmatrix}$

"Новое предположение:", [0.619017287659643, 0.572335496475387]
"Сходимость достигнута на шаге:", 7
[0.619017287659643, 0.572335496475387]

>
> $eval(\tan(0.619017287659643 \cdot 0.572335496475387 + m));$

0.6190172861

> $eval\left(\left(\frac{(1 - a \cdot 0.619017287659643 \cdot 0.619017287659643)}{2}\right)^{\frac{1}{2}}\right);$

0.5723354951

Рисунок 5

Теперь необходимо сравнить трудоемкость методов. В случае метода простых итераций, количество необходимых шагов равно 19. При применении метода Ньютона – это 7 шагов. Тогда трудоемкость метода Ньютона приблизительно в 2.7 раза меньше трудоемкости метода простых итераций в данном случае. В общем случае трудоемкость отличается примерно в 2 раза.

ВЫВОДЫ

В ходе выполнения данной работы были изучены методы решения нелинейных уравнений.

Были построены и запрограммированы алгоритмы методов, а также численно решено тестовое задание.

Также было произведено сравнение трудоемкости методов путем сравнения количества итераций для каждого из изученных методов. Результат – в 2.7 раза.