ОГЛАВЛЕНИЕ

1 Задания для самостоятельной работы

1.1 Задание 1

1.1.1 Постановка задачи

Является ли заданная матрица A матрицей парных сравнений? Для матрицы A найти приближенное $\overline{W}$ и точное W значение главного собственного вектора. Оценить погрешность $\Delta\overline{W} = |\overline{W} - W|$. Определить, является ли матрица парных сравнений согласованной.

$$A = \begin{pmatrix} 1 & 4 & 6 & 8 \\ 1/4 & 1 & 3 & 2 \\ 1/6 & 1/3 & 1 & 3 \\ 1/8 & 1/2 & 1/3 & 1 \end{pmatrix}$$

1.1.2 Ход решения

Метод парных сравнений заключается в сравнении изучаемых объектов между собой. Объекты сравниваются попарно по отношению к их воздействию (“весу”, или “интенсивности”) на общую для них (вышестоящую в иерархии) характеристику.

Для матрицы парных сравнений всегда должно выдерживаться соотношение, отвечающее условию: если при сравнении i-го объекта с j-м объектом ставится оценка a_{ij}, то при сравнении j-го объекта с i-м, оценка a_{ji} должна быть обратной.

Видно, что для матрицы A такое соотношение строго выдерживается для каждой оценки. Поэтому матрица $\mathbf{A}$ является матрицей парных сравнений.

Определим приближенное $\overline{W}$ значение главного собственного вектора. Введем матрицу A (здесь и далее все вычисления выполняются на языке Julia; секция In соответствует введенному коду, секция Out - результатам вычислений; секции для удобства нумеруются).

In [1]:
```
A = [1 4 6 8; 1/4 1 3 2; 1/6 1/3 1 3; 1/8 1/2 1/3 1]
```

```
Out[1]: 4?4 Array{Float64,2}:
   1.0       4.0       6.0       8.0
   0.25      1.0       3.0       2.0
   0.166667  0.333333  1.0       3.0
   0.125     0.5       0.333333  1.0
```

Приближенное значение $\overline{W}$ главного собственного вектора с некоторой точностью ε можно получить по формуле

$$\overline{W} = \lim_{k \to N} \frac{A^k e}{e^T A^k e}, e = (1, 1, K, 1)^T,$$

при таком N, что $|\overline{W}_i - \overline{W}_{i-1}| < \varepsilon$. Выберем ε такое, чтобы абсолютная разница текущего и предыдущего значений не превышала 9 знаков после запятой.

```
In [2]: eps = 1e-9
```

```
Out[2]: 1.0e-9
```

Введем единичный вектор, соответствующий размерности матрицы A.

```
In [3]: e = ones(size(A)[1])
```

```
Out[3]: 4-element Array{Float64,1}:
   1.0
   1.0
   1.0
   1.0
```

Выполним вычисления согласно приведенному алгоритму.

```
In [4]: num = A * e
   denom = e' * num
   prev = num / denom
   num = A * num
   denom = e' * num
   next = num / denom
   while true
       prev = next
       num = A * num
       denom = e' * num
       next = num / denom
       result = abs(next - prev) .> eps
       done = true
       for r in result
```

```
            if !r
                done = false
                break
            end
        end
        if done
            break
        end
    end
    approx = next
```

Out[4]: 4?1 Array{Float64,2}:

```
 0.623881
 0.195949
 0.113291
 0.0668799
```

Таким образом, мы получили приближенное значение $\overline{\mathbf{W}}$ главного собственного вектора с точностью до 9 знаков после запятой.

Найдем аналитические представления максимального собственного числа и главного собственного вектора для матрицы A через ее элементы. Введем символьные обозначения для наддиагональных элементов матрицы и выполним вспомогательные вычисления.

In [5]:
```
a = A[1, 2]
b = A[1, 3]
c = A[1, 4]
d = A[2, 3]
e = A[2, 4]
f = A[3, 4]

B = (d*f/e+e/d/f)+(a*e/c+c/a/e)+(b*f/c+c/b/f)+(a*d/b+b/a/d)
```

Out[5]: 11.916666666666668

In [6]:
```
C = 3-(a*d*f/c+c/a/d/f)-(a*e/b/f+b*f/a/e)-(c*d/a/e)-(c*d/b/e+b*e/c/d)
```

Out[6]: -9.916666666666668

In [7]:
```
x = B^2/2+8C-8
```

Out[7]: -16.3298611111111

In [8]:
```
y = (4(C+3)/3)^3
```

Out[8]: -784.3443072702335

In [9]:
```
r = Float64((x+sqrt(Complex(y)+x^2))^(1/3) +
(x-sqrt(Complex(y)+x^2))^(1/3))
```

Out[9]: 4.5214327417548645

Вычислим точное значение $\lambda_{\mathbf{max}}$ максимального собственного числа.

```
In [10]: lmax = Float64((2+sqrt(r+4))/2 + sqrt((8-r)/4) + B/(2sqrt(r+4)))
```

Out[10]: 5.433240289413801

Определим точное значение главного собственного вектора.

```
In [11]: Q = (lmax-1)^3 + (c+f+e)*(lmax-1)^2 +
((a*e-3)+(b+d)f+c/a+c/b+e/d)*(lmax-1) +
(a*d*f-c-e-f+(b*e/d+b*f/a)+(c*d+a*e-a*d)/b+(c-b)/a/d)
```

Out[11]: 537.2229210859609

```
In [12]: first = c*(lmax-1)^2 + (a*e+b*f)*(lmax-1) + (a*d*f+b*e/d-c)
```

Out[12]: 304.4932032342129

```
In [13]: second = e*(lmax-1)^2 + (d*f+c/a)*(lmax-1) + (b*f/a+c*d/b-e)
```

Out[13]: 94.57288211091533

```
In [14]: third = f*(lmax-1)^2 + (e/d+c/b)*(lmax-1) + (c/a/d+a*e/b-f)
```

Out[14]: 66.82733896987288

```
In [15]: fourth = (lmax-1)^3 - 3*(lmax-1) - (b/a/d+a*d/b)
```

Out[15]: 71.32949677095984

```
In [16]: accurate = [first; second; third; fourth]
```

```
Out[16]: 4-element Array{Float64,1}:
 304.493
  94.5729
  66.8273
  71.3295
```

Вычислим точное значение $\overline{\mathbf{W}}$ главного собственного вектора.

```
In [17]: accurate /= Q
```

```
Out[17]: 4-element Array{Float64,1}:
 0.566791
 0.17604
 0.124394
 0.132774
```

Оценим погрешность $\Delta\overline{W} = |\overline{W} - W|$.

```
In [18]: abs(accurate - approx)
```

Out[18]: 4?1 Array{Float64,2}:

```
0.0570894
0.0199085
0.0111034
0.0658946
```

Расчеты показывают, что абсолютная разница между точным и приближенным значениями главного собственного вектора составляет не более 10%.

Вычислим ранг матрицы A.

In [19]:
```
rank(A)
```

Out[19]: 3

Ранг, не равный 1, означает, что матрица A не является согласованной.

1.2 Задание 2

1.2.1 Постановка задачи

Преобразовать матрицу парных сравнений A из задания 1 таким образом, чтобы она стала абсолютно согласованной. При этом оставить без изменений: * первую строку матрицы, * последнюю строку матрицы.

1.2.2 Ход решения

Матрица A из задания 1 имеет вид.

In [20]:
```
A
```

Out[20]: 4?4 Array{Float64,2}:

```
1.0       4.0       6.0       8.0
0.25      1.0       3.0       2.0
0.166667  0.333333  1.0       3.0
0.125     0.5       0.333333  1.0
```

Добьемся абсолютной согласованности матрицы, не изменяя первую строку. Запомним исходные значения первого столбца матрицы.

In [21]:
```
tmp = deepcopy(A)
origin = A[:,1]
```

Out[21]: 4-element Array{Float64,1}:

```
1.0
0.25
0.166667
0.125
```

Для каждой из строк (2-3) будем последовательно вычитать строку (1), умноженную на первый элемент текущей строки для обнуления первого столбца.

In [22]:
```
for i in 2:size(origin)[1]
    A[i,:] -= A[i,1] * A[1,:]
end
A
```

Out[22]:
```
4?4 Array{Float64,2}:
 1.0   4.0       6.0       8.0
 0.0   0.0       1.5       0.0
 0.0  -0.333333  0.0       1.66667
 0.0   0.0      -0.416667  0.0
```

Обнулим все элементы получившейся матрицы, кроме элементов 1-го столбца и 1-ой строки.

In [23]:
```
A[2:4, 2:4] = 0
A
```

Out[23]:
```
4?4 Array{Float64,2}:
 1.0  4.0  6.0  8.0
 0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0
```

"Вернем" сделанные изменения обратно. После этого имеем матрицу, полученную путем применения преобразований Жордана-Гаусса к исходной матрице без изменения первой строки.

In [24]:
```
for i in 2:size(origin)[1]
    A[i,:] += origin[i] * A[1,:]
end
A
```

Out[24]:
```
4?4 Array{Float64,2}:
 1.0       4.0       6.0   8.0
 0.25      1.0       1.5   2.0
 0.166667  0.666667  1.0   1.33333
 0.125     0.5       0.75  1.0
```

Вычислим ранг полученной матрицы.

In [25]:
```
rank(A)
```

Out[25]:
```
1
```

Ранг равен 1. Это означает, что полученная матрица абсолютно согласована.

Теперь проделаем то же самое, только на этот раз оставим неизменной последнюю строку.

```
In [26]: A = tmp

Out[26]: 4?4 Array{Float64,2}:
 1.0       4.0       6.0       8.0
 0.25      1.0       3.0       2.0
 0.166667  0.333333  1.0       3.0
 0.125     0.5       0.333333  1.0

In [27]: origin = A[:,4]

Out[27]: 4-element Array{Float64,1}:
 8.0
 2.0
 3.0
 1.0

In [28]: for i in 1:size(A)[1]-1
             A[i,:] -= A[i,4] * A[4,:]
         end
         A

Out[28]: 4?4 Array{Float64,2}:
  0.0        0.0       3.33333   0.0
  0.0        0.0       2.33333   0.0
 -0.208333  -1.16667   0.0       0.0
  0.125      0.5       0.333333  1.0

In [29]: A[1:3, 1:3] = 0
         A

Out[29]: 4?4 Array{Float64,2}:
 0.0    0.0  0.0       0.0
 0.0    0.0  0.0       0.0
 0.0    0.0  0.0       0.0
 0.125  0.5  0.333333  1.0

In [30]: for i in 1:size(origin)[1]-1
             A[i,:] += origin[i] * A[4,:]
         end
         A

Out[30]: 4?4 Array{Float64,2}:
 1.0    4.0  2.66667   8.0
```

0.25	1.0	0.666667	2.0
0.375	1.5	1.0	3.0
0.125	0.5	0.333333	1.0

In [31]: rank(A)

Out[31]: 1

Таким образом, получили абсолютно согласованную матрицу, не изменяя последнюю строку.