Санкт-Петербургский Политехнический Университет Петра Великого

Кафедра компьютерных систем и программных технологий

Отчёт по дополнительному заданию

Курс: «Методы оптимизации и принятия решений»

Тема: «Проверка расчетов»

Выполнил студент:

Бояркин Никита Сергеевич
Группа: 13541/3

Проверил:

Сиднев Александр Георгиевич

Санкт-Петербург
2018 г.

Содержание

Дополнительное задание

1.1 Ход работы

1.1.1 Формулировка задачи

Таким образом, с учетом вышеприведенных формул, имеет место следующая оптимизационная задача:

$$\begin{aligned} \lambda_1 &= w_1 * G_M(N-1)/G_M(N) \to \sup \\ S(\mu) &= \sum_{i=1}^{M} \mu_i = S^* \end{aligned}$$

Здесь

Символ	Семантическое значение
N	Количество сообщений, циркулирующих в системе
$G_M(N)$	Нормирующйи коэффциент системы при заданных N и M

При этом даны следующие данные:

$$P = \begin{pmatrix} 0 & 0.2 & 0.3 & 0.5 \\ 0.5 & 0 & 0.3 & 0.2 \\ 0.4 & 0.1 & 0 & 0.5 \\ 0.6 & 0.2 & 0.2 & 0 \end{pmatrix}$$

Символ	Значение
N	5
M	4
S^*	12

1.1.2 Решение для проверки

Данное решение необходимо проверить:

Символ	Значение
μ_1	3.784773
μ_2	2.095212
μ_3	2.644322
μ_4	3.475693

i	$\lambda_i(\mu_1, \mu_2, \mu_3, \mu_4)$
1	2.585434
2	1.137591
3	1.577115
4	2.301036

1.1.3 Решение задачи через нормирующую константу методом Ньютона

Решение задачи через нормирующую константу методом Ньютона с начальным приближением $u = [3, 3, 3, 3]$ и значением ошибки $e = 1e^{-8}$:

```
clear all;
close all;
clc;
format long g;

% delete(gcp);
% distcomp.feature('LocalUseMpiexec', false);
% parpool();

p =    [0 0.2 0.3 0.5;
        0.5 0 0.3 0.2;
        0.4 0.1 0 0.5;
        0.6 0.2 0.2 0];

M = 4;
N = 5;
S = 12;

c = [1 1 1 1];
a = [1 1 1 1];

%% Initialize w

A = p' - diag(ones(1, M));
A(4, :) = [1; 1; 1; 1];
b = [0; 0; 0; 1];
w = inv(A) * b;
w = (1 / w(1)) * w';

%% Error

e = 1e-08;

%% First approximation

fprintf('Start Conditions\ne = %.8f\nw = [%.6f %.6f %.6f %.6f]\n\n', e, w);

%% Syms u

u = sym('u', [1 M]);

%% Find characteristics

G = gl3(u, w, c, M, N);
[L, Lam] = pl3(u, w, M, N, G);
[pL, pLam] = pl3(u, w, M, N - 1, G);

Function = sym(zeros(1, M));

for i = 1 : M
    Function(i) = simplify(S * (L(i) - pL(i)));
end

%% Newton method with jacobian matrix

uCurrent = [3 3 3 3];
uPrevious = zeros(1, M);

jaco = jacobian(Function - u);

index = 0;
condition = true;
```

```
while condition
    resf = zeros(M, 1);
    % parfor i = 1 : M
    for i = 1 : M
        resf(i) = subs(Function(i) - u(i), u, uCurrent);
    end

    fprintf('%d | u = [%.6f %.6f %.6f %.6f] | F(u) = [%.6f %.6f %.6f %.6f]\n', index, double(uCurrent), double(resf));

    uPrevious = uCurrent;

    resjaco = zeros(M, M);
    % parfor i = 1 : M
    for i = 1 : M
        for j = 1 : M
            resjaco(i, j) = subs(jaco(i, j), u, uPrevious);
        end
    end

    resf = zeros(M, 1);
    % parfor i = 1 : M
    for i = 1 : M
        resf(i) = subs(Function(i) - u(i), u, uPrevious);
    end

    uCurrent = uPrevious - (resjaco \ resf)';

    condition = false;
    for i = 1 : M
        condition = condition || abs(uCurrent(i) - uPrevious(i)) > e;
    end

    index = index + 1;

    if (~condition)
        fprintf('%d | u = [%.6f %.6f %.6f %.6f] | F(u) = [%.6f %.6f %.6f %.6f]\n', index, double(uCurrent), double(resf));
    end
end

resf = zeros(M, 1);
% parfor i = 1 : M
for i = 1 : M
    resf(i) = subs(Function(i) - u(i), u, uCurrent);
end

rU = double(uCurrent);
rL = double(subs(L, u, uCurrent));
rLam = double(subs(Lam, u, uCurrent));
fprintf('\nResult\nu = [%.6f %.6f %.6f %.6f]\nF(u) = [%.6f %.6f %.6f %.6f]\nL(N) = [%.6f %.6f %.6f %.6f]\nlam(N) = [%.6f %.6f %.6f %.6f]\n', rU, double(resf), rL, rLam);
```

Расчет нормирующей константы:

```
function [G] = gl3(u, w, c, M, N)
    G = sym(zeros(M, N + 1));
    G(:, 1) = 1;

    for k = 0 : N
        %% Find G(1, k)

        tempMul = 1;
        for j = 1 : k
            tempMul = tempMul * min(j, c(1)) * u(1);
        end

```

```
        G(1, k + 1) = (w(1) ^ k) / tempMul;
    end

    for r = 2 : M
        for k = 1 : N
            %% Find G(r, k)

            tempSum = 0;
            for h = 0 : k
                Z = 0;
                if (h == 0)
                    Z = 1;
                else
                    %% Find Z(r, h)
                    tempMul = 1;
                    for j = 1 : h
                        tempMul = tempMul * min(j, c(r)) * u(r);
                    end

                    Z = (w(r) ^ h) / tempMul;
                end

                tempSum = tempSum + Z * G(r - 1, k - h + 1);
            end

            G(r, k + 1) = tempSum;
        end
    end

    G = simplify(G);
end
```

Расчет основных характеристик СМО:

```
function [L, Lam] = pl3(u, w, M, N, G)
    L = sym(zeros(1, M));
    Lam = sym(zeros(1, M));

    for i = 1 : M
        tempSum = 0;
        for n = 1 : N
            tempSum = tempSum + ((w(i) / u(i)) ^ n) * G(M, N - n + 1);
        end

        L(i) = tempSum / G(M, N + 1);
        Lam(i) = w(i) * G(M, N - 1 + 1) / G(M, N + 1);
    end

    L = simplify(L);
    Lam = simplify(Lam);
end
```

Результат решения задачи методом Ньютона с начальным приближением $u = [3, 3, 3, 3]$ и значением ошибки $e = 1e^{-8}$:

```
Start Conditions
e = 0.00000001
w = [1.000000 0.439698 0.610553 0.893216]

0 | u = [3.000000 3.000000 3.000000 3.000000] | F(u) = [2.700814 -2.261395 -1.503638
  1.064219]
1 | u = [3.889288 1.731090 2.683429 3.696193] | F(u) = [-0.506297 1.543902 -0.216229
  -0.821376]
2 | u = [3.811699 2.027378 2.669695 3.491228] | F(u) = [-0.111054 0.233893 -0.089068
  -0.033771]
3 | u = [3.782389 2.090288 2.644331 3.482992] | F(u) = [-0.002673 0.007734 -0.001436
  -0.003626]
```

```
4 | u = [3.781634 2.092510 2.643886 3.481970] | F(u) = [-0.000004 0.000009 -0.000003 -0.000003]
5 | u = [3.781633 2.092512 2.643885 3.481970] | F(u) = [-0.000000 0.000000 -0.000000 -0.000000]
6 | u = [3.781633 2.092512 2.643885 3.481970] | F(u) = [-0.000000 0.000000 -0.000000 -0.000000]

Result
u = [3.781633 2.092512 2.643885 3.481970]
F(u) = [0.000000 -0.000000 -0.000000 0.000000]
L(N) = [1.474996 0.979669 1.152046 1.393289]
lam(N) = [2.582525 1.135533 1.576768 2.306753]
```

Алгоритм успешно сходится за 7 итераций. С каждой итерацией алгоритма значения целевых функций стремятся к нулю.

Таблица 1.1 Сравнительный анализ решений

i	$\mu_{task}(i)$	$\|\mu_{my}(i)\|$	$\Delta\mu(i)$	$\lambda_{task}(i)$	$\lambda_{my}(i)$	$\|\Delta\lambda(i)\|$
1	3.784773	3.781633	0.00314	2.585434	2.582525	0.002909
2	2.095212	2.092512	0.0027	1.137591	1.135533	0.002058
3	2.644322	2.643885	0.000437	1.577115	1.576768	0.000347
4	3.475693	3.481970	0.006277	2.301036	2.306753	0.005717

1.2 Вывод

Решение сошлось с заданным до третьего знака после запятой. Небольшая погрешность объясняется различием в алгоритме расчета нормирующей константы G, а также вычислением при помощи различных языков программирования (R и Matlab).