

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»

РАСЧЕТНО-ГРАФИЧЕСКОЕ ЗАДАНИЕ

по дисциплине “Анализ функционирования распределенных
вычислительных систем”

на тему

**МЕТОДИКА РАСЧЕТА ПОКАЗАТЕЛЕЙ
ЖИВУЧЕСТИ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ**

Выполнил студент _____
Ф.И.О.

Группы _____

Работу принял _____ профессор Павский К.В.
подпись

Защищена _____ Оценка _____

Оглавление

МЕТОДИКА РАСЧЕТА ПОКАЗАТЕЛЕЙ ЖИВУЧЕСТИ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ.....	3
1. Континуальный подход	3
a. Континуальный подход высокая производительность	4
b. Континуальный подход невысокая производительность	5
2. Расчет моментов. Процесс рождения и гибели	7
3. Аналитическая модель. (Высокая производительность).....	13
a. Аналитическая модель рождения гибели на языке Java	14
4. Модель функционирования ВС. Процесс рождения и гибели.....	15
a. Моделирование рождения гибели на языке Java	16
5. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	19
ПРИЛОЖЕНИЕ 1. ЛИСТИНГ ПРОГРАММЫ	20

МЕТОДИКА РАСЧЕТА ПОКАЗАТЕЛЕЙ ЖИВУЧЕСТИ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

1. Континуальный подход

Случай 1. Восстанавливающая система имеет высокую производительность

Восстановление без очереди: $N - \mathcal{N}(i, t) \leq m, \quad N\lambda \leq m\mu$

$$\frac{d}{dt} N(i, t) = N\mu - (\lambda + \mu)N(i, t), \quad N(i, 0) = i, \quad i \in E_{N-m}^N$$

Линейным дифференциальным уравнением первого порядка называется уравнение, линейное относительно искомой функции и ее производной. Обыкновенное линейное дифференциальное уравнение пишут в виде:

$$\frac{dy}{dx} + Py = Q$$

где P и Q - заданные функции от x .

Общее решение линейного уравнения первого порядка находится двумя квадратурами

$$y = e^{-\int P dx} (C_1 + \int Q e^{\int P dx} dx)$$

Степанов В.В. Курс дифференциальных уравнений. – М.: Едиториал УРСС, 2006. – 472 с.

а. Континуальный подход высокая производительность

Решим уравнение высокой производительности:

$$\frac{dN(i, t)}{dt} + (\lambda + \mu)N(i, t) = N\mu$$

$$N(i, 0) = i$$

Уравнение вида:

$$\frac{dy}{dx} + Py = Q$$

имеет решение:

$$y = e^{-\int P dx} (C_1 + \int Q e^{\int P dx} dx)$$

В нашем случае:

$$(\lambda + \mu) = P$$

$$N\mu = Q$$

Сделаем подстановку, получаем:

$$N(i, t) = e^{-\int (\lambda + \mu) dt} (C_1 + \int N\mu e^{\int (\lambda + \mu) dt} dt)$$

$$N(i, t) = e^{-(\lambda + \mu)t} (C_1 + \frac{N\mu}{\lambda + \mu} e^{(\lambda + \mu)t})$$

Выполним подстановку, получим значение константы:

$$N(i, 0) = i$$

$$N(i, 0) = e^{-(\lambda + \mu)0} (C_1 + \frac{N\mu}{\lambda + \mu} e^{(\lambda + \mu)0}) = i$$

$$i = C_1 + \frac{N\mu}{\lambda + \mu}$$

$$C_1 = i - \frac{N\mu}{\lambda + \mu} = \frac{i(\lambda + \mu) - N\mu}{\lambda + \mu}$$

Выразим $N(i, t)$:

$$N(i, t) = e^{-(\lambda + \mu)t} \left(\frac{i(\lambda + \mu) - N\mu}{\lambda + \mu} + \frac{N\mu}{\lambda + \mu} e^{(\lambda + \mu)t} \right)$$

Получим ответ:

$$N(i, t) = \frac{N\mu}{\lambda + \mu} + \frac{i\lambda - (N - i)\mu}{\lambda + \mu} e^{-(\lambda + \mu)t}$$

б. Континуальный подход невысокая производительность

Случай 2. Восстанавливающая система имеет невысокую производительность

$$N - N(i, t) > m, \quad i \in E_0^{N-m-1} = \{0, 1, \dots, N - m - 1\}.$$

Очевидно, что в этом случае $M(i, t) = m$

$$\frac{d}{dt} N(i, t) = m\mu - \lambda N(i, t), \quad N(i, 0) = i, \quad i \in E_0^{N-m-1}.$$

Решим уравнение невысокой производительности:

$$\frac{dN(i, t)}{dt} = m\mu - \lambda N(i, t)$$

$$\frac{dN(i, t)}{dt} + \lambda N(i, t) = m\mu$$

$$N(i, 0) = i$$

Уравнение вида:

$$\frac{dy}{dx} + Py = Q$$

имеет решение:

$$y = e^{-\int P dx} (C_1 + \int Q e^{\int P dx} dx)$$

В нашем случае:

$$(\lambda) = P$$

$$m\mu = Q$$

Сделаем подстановку, получаем:

$$N(i, t) = e^{-\int \lambda dt} (C_1 + \int (m\mu) e^{\int \lambda dt} dt)$$

$$N(i, t) = e^{-\lambda t} (C_1 + \frac{m\mu}{\lambda} e^{\lambda t})$$

Выполним подстановку, получим значение константы:

$$N(i, 0) = i$$

$$N(i, 0) = C_1 + \frac{m\mu}{\lambda} = i$$

$$C_1 = i - \frac{m\mu}{\lambda}$$

Выразим $N(i, t)$:

$$N(i, t) = e^{-\lambda t} \left(i - \frac{m\mu}{\lambda} + \frac{m\mu}{\lambda} e^{\lambda t} \right)$$

$$N(i, t) = e^{-\lambda t} \left(\frac{i\lambda}{\lambda} - \frac{m\mu}{\lambda} + \frac{m\mu}{\lambda} e^{\lambda t} \right)$$

Получим ответ:

$$N(i, t) = \frac{m\mu}{\lambda} + \frac{i\lambda - m\mu}{\lambda} e^{-\lambda t}$$

2. Расчет моментов. Процесс рождения и гибели

Итак, для решения системы введем производящую функцию:

$$F(z, t) = \sum_{k=0}^N z^k \cdot P_k(t),$$

Начальные условия:

$$F(z, 0) = z^i.$$

Суммируя первое уравнение системы (1) со вторым, умноженным на z^k , где $k = 1, 2, \dots, N-1$ и с последним, умноженным на z^N , получим

$$\frac{dF(z, t)}{dt} + (z-1)(\lambda z + \mu) \frac{dF(z, t)}{dz} = N\lambda(z-1)F(z, t)$$

Воспользуемся правилом дифференцирования произведения:

$$\frac{d(F(z)G(z))}{dz} = \frac{dF(z)}{dz}G(z) + F(z)\frac{dG(z)}{dz}$$

Дифференцируем по z :

$$(z-1)(\lambda z + \mu) = \lambda z^2 + \mu z - \lambda z - \mu$$

$$\frac{d}{dz}(z-1)(\lambda z + \mu) = 2\lambda z + \mu - \lambda$$

$$\begin{aligned} \frac{d}{dz} \left((z-1)(\lambda z + \mu) \frac{dF(z, t)}{dz} \right) &= \\ &= \frac{d}{dz} ((z-1)(\lambda z + \mu)) * \frac{dF(z, t)}{dz} + (z-1)(\lambda z + \mu) * \frac{d^2 F(z, t)}{dz^2} = \\ &= (2\lambda z + \mu - \lambda) \frac{dF(z, t)}{dz} + (z-1)(\lambda z + \mu) * \frac{d^2 F(z, t)}{dz^2} \end{aligned}$$

$$N\lambda(z-1)F(z, t) = (N\lambda z - N\lambda)F(z, t) = N\lambda z F(z, t) - N\lambda F(z, t)$$

$$\begin{aligned} \frac{d}{dz} (N\lambda z F(z, t)) &= \frac{d}{dz} (N\lambda z) F(z, t) + (N\lambda z) \frac{dF(z, t)}{dz} = \\ &= \frac{d}{dz} (N\lambda z) F(z, t) + (N\lambda z) \frac{dF(z, t)}{dz} = N\lambda F(z, t) + N\lambda z \frac{dF(z, t)}{dz} \end{aligned}$$

Соберем все подстановки вместе

$$\begin{aligned} \frac{d^2 F(z, t)}{dt dz} + (2\lambda z + \mu - \lambda) \frac{dF(z, t)}{dz} + (z - 1)(\lambda z + \mu) * \frac{d^2 F(z, t)}{dz^2} = \\ = N\lambda F(z, t) + N\lambda z \frac{dF(z, t)}{dz} + N\lambda \frac{dF(z, t)}{dz} \end{aligned}$$

$$\begin{aligned} \frac{d^2 F(z, t)}{dt dz} + (z - 1)(\lambda z + \mu) \frac{d^2 F(z, t)}{dz^2} + ((2\lambda z + \mu - \lambda) - N\lambda z - \\ - N\lambda) \frac{dF(z, t)}{dz} = N\lambda F(z, t) \end{aligned}$$

$$\begin{aligned} 2\lambda z + \mu - \lambda - N\lambda z - N\lambda &= 2\lambda z - N\lambda z - \lambda - N\lambda + \mu = \\ \lambda z(2 - N) + \lambda(N - 1) + \mu &= +(-(N - 2)\lambda z + (N - 1)\lambda + \mu \end{aligned}$$

Получим:

$$\begin{aligned} \frac{d^2 F(z, t)}{dt dz} + (z - 1)(\lambda z + \mu) \frac{d^2 F(z, t)}{dz^2} + (-(N - 2)\lambda z + (N - 1)\lambda \\ + \mu) \frac{dF(z, t)}{dz} = N\lambda F(z, t) \end{aligned}$$

Продифференцируем по z еще раз:

$$\begin{aligned} \frac{d^2 F(z, t)}{dt dz} + (z - 1)(\lambda z + \mu) \frac{d^2 F(z, t)}{dz^2} + (\lambda z(2 - N) + \lambda(N - 1) \\ + \mu) \frac{dF(z, t)}{dz} = N\lambda F(z, t) \end{aligned}$$

Вынесем везде члены с z

$$\begin{aligned} \frac{d^2 F(z, t)}{dt dz} + (z - 1)(\lambda z + \mu) \frac{d^2 F(z, t)}{dz^2} + \lambda z(2 - N) \frac{dF(z, t)}{dz} + \\ + (\lambda(N - 1) + \mu) \frac{dF(z, t)}{dz} = N\lambda F(z, t) \end{aligned}$$

$$\begin{aligned}
& \frac{d}{dz} \left((z-1)(\lambda z + \mu) \frac{d^2 F(z, t)}{dz^2} \right) = \\
& = \frac{d}{dz} ((z-1)(\lambda z + \mu)) * \frac{d^2 F(z, t)}{dz^2} + (z-1)(\lambda z + \mu) * \frac{d^3 F(z, t)}{dz^3} = \\
& = (2\lambda z + \mu - \lambda) \frac{d^2 F(z, t)}{dz^2} + (z-1)(\lambda z + \mu) * \frac{d^3 F(z, t)}{dz^3}
\end{aligned}$$

$$\begin{aligned}
& \frac{d}{dz} \left(\lambda z(2-N) \frac{dF(z, t)}{dz} \right) = \\
& = \frac{d}{dz} (\lambda z(2-N)) \frac{dF(z, t)}{dz} + \lambda z(2-N) \frac{d^2 F(z, t)}{dz^2} = \\
& = \lambda(2-N) \frac{dF(z, t)}{dz} + \lambda z(2-N) \frac{d^2 F(z, t)}{dz^2}
\end{aligned}$$

$$\frac{d}{dz} \left((\lambda(N-1) + \mu) \frac{dF(z, t)}{dz} \right) = (\lambda(N-1) + \mu) \frac{d^2 F(z, t)}{dz^2}$$

$$N\lambda F(z, t) = N\lambda \frac{dF(z, t)}{dz}$$

Соберем все члены вместе:

$$\begin{aligned}
& \frac{d^3 F(z, t)}{dt dz^2} + (2\lambda z + \mu - \lambda) \frac{d^2 F(z, t)}{dz^2} + (z-1)(\lambda z + \mu) * \frac{d^3 F(z, t)}{dz^3} \\
& + \lambda(2-N) \frac{dF(z, t)}{dz} + \lambda z(2-N) \frac{d^2 F(z, t)}{dz^2} \\
& + (\lambda(N-1) + \mu) \frac{d^2 F(z, t)}{dz^2} = N\lambda \frac{dF(z, t)}{dz}
\end{aligned}$$

$$\begin{aligned}
& \frac{d^3 F(z, t)}{dt dz^2} + (z-1)(\lambda z + \mu) * \frac{d^3 F(z, t)}{dz^3} + \\
& + ((2\lambda z + \mu - \lambda) + \lambda z(2-N) + (\lambda(N-1) + \mu)) \frac{d^2 F(z, t)}{dz^2} + \\
& + (2\lambda - \lambda N - N\lambda) \frac{dF(z, t)}{dz} = 0
\end{aligned}$$

$$\begin{aligned}
& (2\lambda z + \mu - \lambda) + \lambda z(2 - N) + (\lambda(N - 1) + \mu) \\
& = (2\lambda z + \mu - \lambda + 2\lambda z - \lambda z N + \lambda N - \lambda + \mu) \\
& = 4\lambda z - 2\lambda + 2\mu = 2(2\lambda z - \lambda + \mu)
\end{aligned}$$

$$2\lambda - \lambda N - N\lambda = 2\lambda - 2\lambda N = 2\lambda(1 - N) = -2(N - 1)\lambda$$

Получим:

$$\begin{aligned}
& \frac{d^3 F(z, t)}{dt dz^2} + (z - 1)(\lambda z + \mu) \frac{d^3 F(z, t)}{dz^3} + 2(2\lambda z - \lambda + \mu) \frac{d^2 F(z, t)}{dz^2} \\
& - 2(N - 1)\lambda \frac{dF(z, t)}{dz} = 0
\end{aligned}$$

В полученные производные подставим $z=1$

$$\begin{cases} \frac{d^2 F(1, t)}{dt dz} + (1 - 1)(\lambda z + \mu) \frac{d^2 F(1, t)}{dz^2} + (-(N - 2)\lambda 1 + (N - 1)\lambda + \mu) \frac{dF(1, t)}{dz} = N\lambda F(z, t) \\ \frac{d^3 F(1, t)}{dt dz^2} + (1 - 1)(\lambda 1 + \mu) \frac{d^3 F(1, t)}{dz^3} + 2(2\lambda 1 - \lambda + \mu) \frac{d^2 F(1, t)}{dz^2} - 2(N - 1)\lambda \frac{dF(1, t)}{dz} = 0 \end{cases}$$

$$\begin{cases} \frac{d^2 F(1, t)}{dt dz} + (-(N - 2)\lambda 1 + (N - 1)\lambda + \mu) \frac{dF(1, t)}{dz} = N\lambda F(1, t) \\ \frac{d^3 F(1, t)}{dt dz^2} + 2(\lambda + \mu) \frac{d^2 F(1, t)}{dz^2} - 2(N - 1)\lambda \frac{dF(1, t)}{dz} = 0 \end{cases}$$

$$\begin{cases} \frac{d^2 F(1, t)}{dt dz} + (-N\lambda + 2\lambda + N\lambda - \lambda + \mu) \frac{dF(1, t)}{dz} = N\lambda 1 \\ \frac{d^3 F(1, t)}{dt dz^2} + 2(\lambda + \mu) \frac{d^2 F(1, t)}{dz^2} = 2(N - 1)\lambda \frac{dF(1, t)}{dz} \end{cases}$$

$$\begin{cases} \frac{d^2 F(1, t)}{dt dz} + (\lambda + \mu) \frac{dF(1, t)}{dz} = N\lambda \\ \frac{d^3 F(1, t)}{dt dz^2} + 2(\lambda + \mu) \frac{d^2 F(1, t)}{dz^2} = 2(N - 1)\lambda \frac{dF(1, t)}{dz} \end{cases}$$

Выразим Математическое ожидание и Дисперсию через моменты:

$$\begin{cases} M_i(t) = \frac{\partial}{\partial z} F(1, t); \\ D_i(t) = \frac{\partial^2}{\partial z^2} F(1, t) - (M_i(t))^2 + M_i(t). \end{cases}$$

$$Q(t)_i = \frac{\partial^2}{\partial z^2} F(z, t) \Big|_{z=1}$$

Выполнил подстановки:

$$\frac{d^2 F(1, t)}{dt dz} = \frac{dM_i(t)}{dt}$$

$$\frac{dF(1, t)}{dz} = M_i(t)$$

$$\frac{d^3 F(1, t)}{dt dz^2} = \frac{dQ_i}{dt}$$

$$\frac{d^2 F(1, t)}{dz^2} = Q_i$$

Получим следующие

$$\begin{cases} \frac{dM_i(t)}{dt} + (\lambda + \mu)M_i(t) = N\lambda \\ \frac{dQ_i}{dt} + 2(\lambda + \mu)Q_i = 2(N-1)\lambda M_i(t) \end{cases}$$

Полученной системы недостаточно для получения решения, попробуем получить $M_i(t)$

Уравнение вида:

$$\frac{dy}{dx} + Py = Q$$

имеет решение:

$$y = e^{-\int P dx} (C_1 + \int Q e^{\int P dx} dx)$$

В нашем случае:

$$(\lambda + \mu) = P$$

$$N\lambda = Q$$

Сделаем подстановку, получаем:

$$M_i(t) = e^{-\int (\lambda + \mu) dt} (C_1 + \int N \lambda e^{\int (\lambda + \mu) dt} dt)$$

$$M_i(t) = e^{-(\lambda + \mu)t} (C_1 + \frac{N\lambda}{\lambda + \mu} e^{(\lambda + \mu)t})$$

Выполним подстановку, получим значение константы:

$$M(0) = i$$

$$M_i(0) = e^{-(\lambda + \mu)0} (C_1 + \frac{N\lambda}{\lambda + \mu} e^{(\lambda + \mu)0}) = i$$

$$i = C_1 + \frac{N\lambda}{\lambda + \mu}$$

$$C_1 = i - \frac{N\lambda}{\lambda + \mu} = \frac{i(\lambda + \mu) - N\lambda}{\lambda + \mu}$$

Выразим $M_i(t)$:

$$M_i(t) = e^{-(\lambda + \mu)t} (\frac{i\mu - N\lambda + i\lambda}{\lambda + \mu} + \frac{N\lambda}{\lambda + \mu} e^{(\lambda + \mu)t})$$

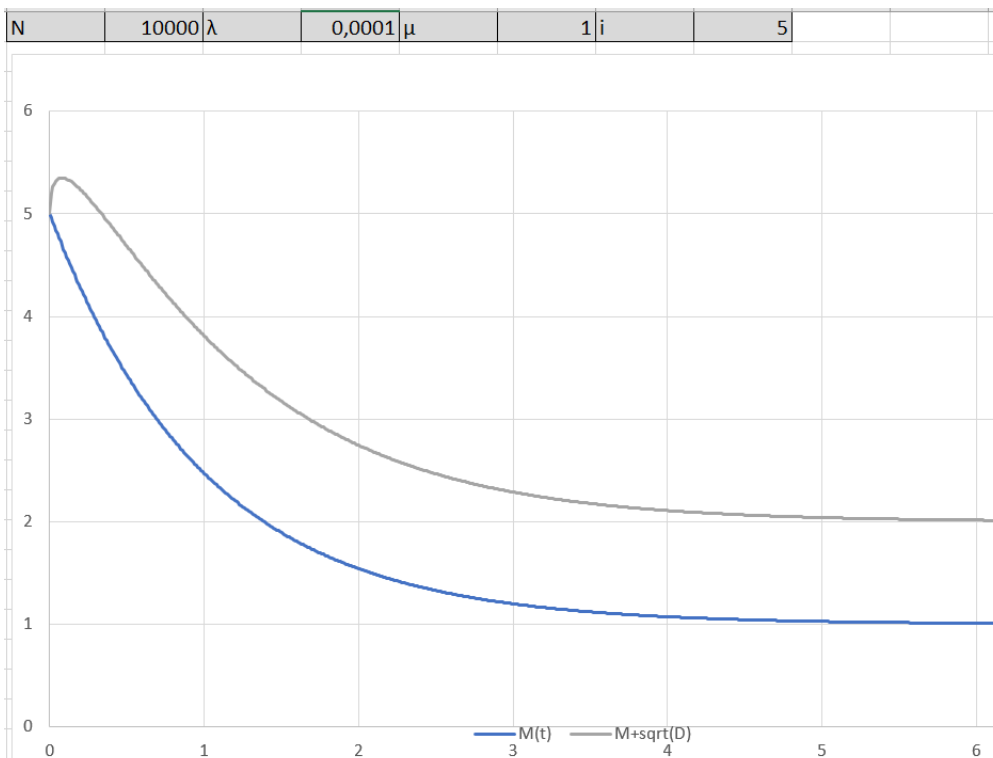
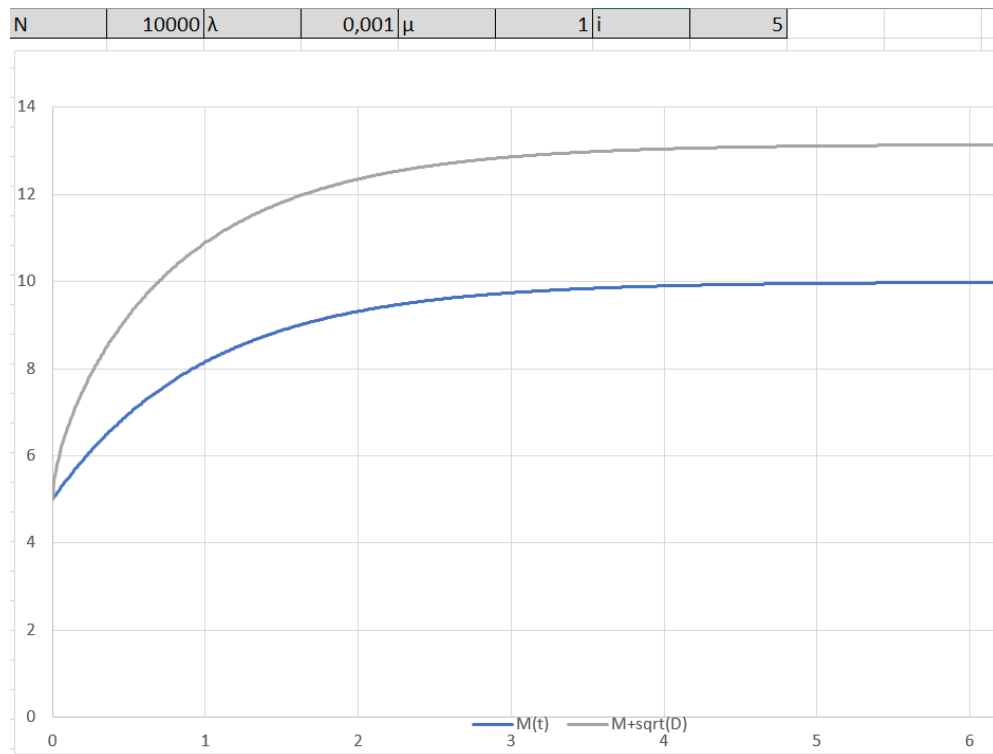
Получим ответ:

$$M_i(t) = \frac{N\mu}{\lambda + \mu} + \frac{i\mu - \lambda(N + i)}{\lambda + \mu} e^{-(\lambda + \mu)t}$$

3. Аналитическая модель. (Высокая производительность)

$$\begin{cases} M_i(t) = \frac{N \cdot \lambda}{\lambda + \mu} + \frac{i\mu - (N - i)\lambda}{\lambda + \mu} \cdot e^{-(\lambda + \mu)t}, \\ D_i(t) = \frac{N \cdot \lambda \cdot \mu}{(\lambda + \mu)^2} + \frac{\lambda^2(N - i) + \mu(i\mu - \lambda N)}{(\lambda + \mu)^2} \cdot e^{-(\lambda + \mu)t} - \frac{\lambda^2(N - i) + i\mu^2}{(\lambda + \mu)^2} \cdot e^{-2(\lambda + \mu)t}. \end{cases}$$

а. Аналитическая модель рождения гибели в программе excel.



а. Аналитическая модель рождения гибели на языке Java

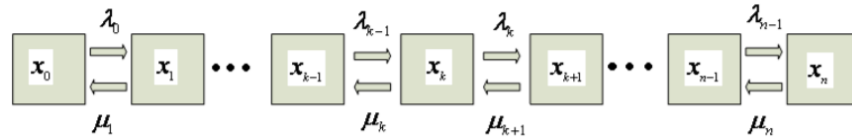
//аналитическая модель рождения гибели на языке Java

```
public Data start() {
    Data data = new Data(T);
    for (int t=0; t<T; t++) {
        data.x[t]= t*h;
        data.y1[t] = ((lambda*N)/(lambda+mu)) +
            ((mu*i-lambda*(N-i))/(lambda+mu))*
            Math.exp(-1.0*(lambda+mu)*data.x[t] );
        double d = ((N*lambda*mu)/((lambda+mu)*(lambda+mu))) +
            ((lambda*lambda*(N-i)+mu*(i*mu-
            lambda*N))/((lambda+mu)*(lambda+mu)))*
            Math.exp(-(lambda+mu)* data.x[t] ) -
            ((lambda*lambda*(N-i)
            +i*mu*mu)/((lambda+mu)*(lambda+mu)))*
            Math.exp(-2.0*(lambda+mu)* data.x[t] );
        data.y2[t]=data.y1[t]+Math.sqrt(d);
        data.h=Math.max(data.h, data.y2[t]);
    }
    return data;
}
```

4. Модель функционирования ВС. Процесс рождения и гибели.

Требуется найти $P_k(t)$, вероятность того, что в момент времени $t \in [0, \infty)$ в ВС в состоянии отказа находится k ЭМ при условии, что в начальный момент $t = 0$ в системе в состоянии отказа находилось i ЭМ, $i = 0, 1, \dots, N$.

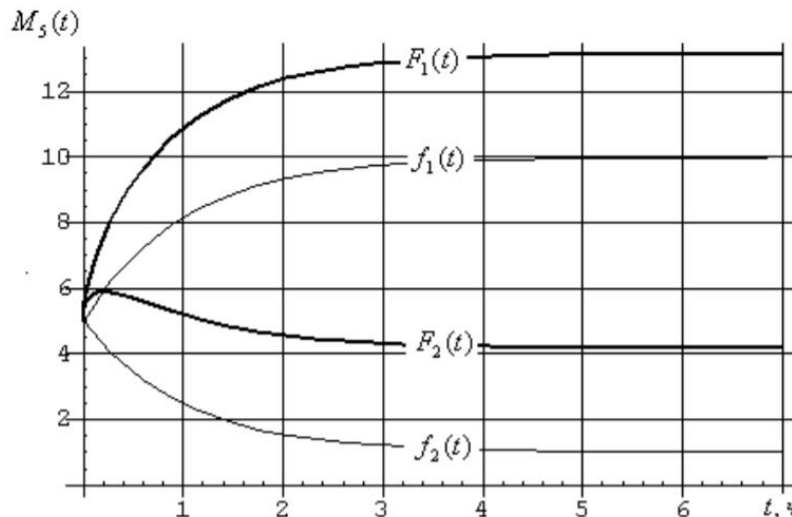
В рамках процесса рождения и гибели:



Формализация процесса представлена системой дифференциальных уравнений:

$$\begin{cases} \frac{d}{dt} P_0(t) = -\lambda_0 \cdot P_0(t) + \mu_1 \cdot P_1(t), \\ \frac{d}{dt} P_k(t) = -(\lambda_k + \mu_k) \cdot P_k(t) + \lambda_{k-1} \cdot P_{k-1}(t) + \mu_{k+1} \cdot P_{k+1}(t), 1 \leq k < N, \\ \frac{d}{dt} P_N(t) = -\mu_N \cdot P_N(t) + \lambda_{N-1} \cdot P_{N-1}(t), \end{cases}$$

с начальными условиями $P_i(0) = 1, P_k(0) = 0, \forall k \neq i$



При $N = 10^4$, $\lambda = 10^{-3}$ 1/ч, $\mu = 1$ 1/ч, получаем кривые $f_1(t) = M_5(t)$, $F_1(t) = M_5(t) + \sigma(t)$.

При $N = 10^4$, $\lambda = 10^{-4}$ 1/ч, $\mu = 1$ 1/ч, получаем кривые $f_2(t) = M_5(t)$, $F_2(t) = M_5(t) + \sigma(t)$.

a. Моделирование рождения гибели на языке Java

//метод моделирования рождения гибели на языке Java

```
public Data start() {
    Data data = new Data(T);
    double listM2[] = new double[T];
    double listM[] = new double[T];
    double listD[] = new double[T];

    for (int k = 0; k < K; k++) {
        int listN[] = new int[T];
        listN[0] = N-i;

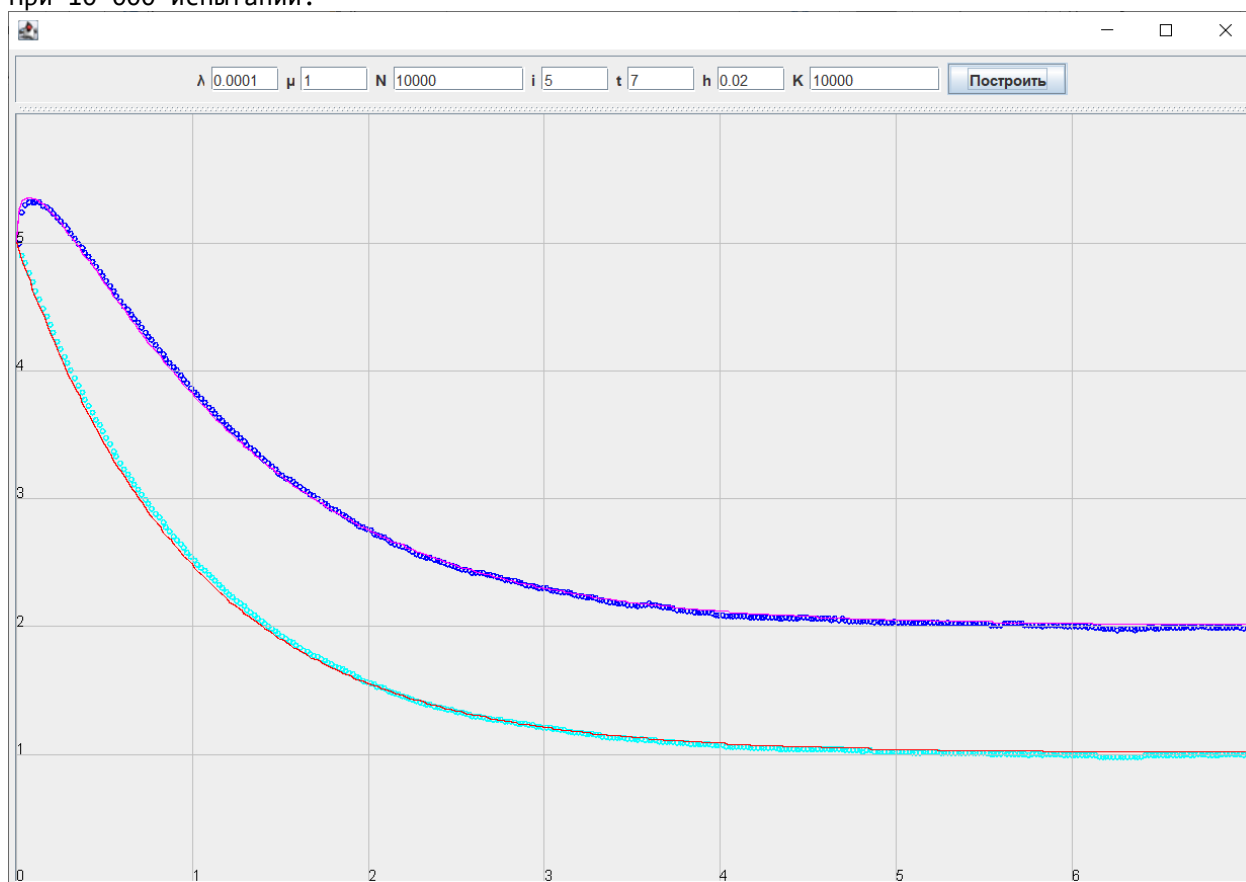
        for (int i = 1; i < T; i++) {
            double t1 = i * h;
            double t2 = t1 + h;
            listN[i] = listN[i - 1];
            if (listN[i] > 0) {
                double R = Math.exp(-lambda * listN[i] * h);
                double r = Math.random();
                if (r > R) {
                    listN[i]--;
                }
            }
            if (listN[i] < N) {
                double R = Math.exp(-mu * (N - listN[i]) * h);
                double r = Math.random();
                if (r > R) {
                    listN[i]++;
                }
            }
        }
        for (int i = 0; i < T; i++) {
            listM[i] += listN[i];
            listM2[i] += (listN[i] * listN[i]);
        }
    }
    data.h = 0;
    for (int i = 0; i < T; i++) {
        listM[i] /= K;
        listD[i] = listM2[i] / K - (listM[i] * listM[i]);
        data.x[i] = i * h;
        data.y1[i]=(N - listM[i]);
        data.y2[i]=(N - listM[i])+Math.sqrt( listD[i]);
        data.h = Math.max(data.h , data.y2[i]);
    }

    data.w = T*h;
    return data
}
```


При 50 испытаниях:



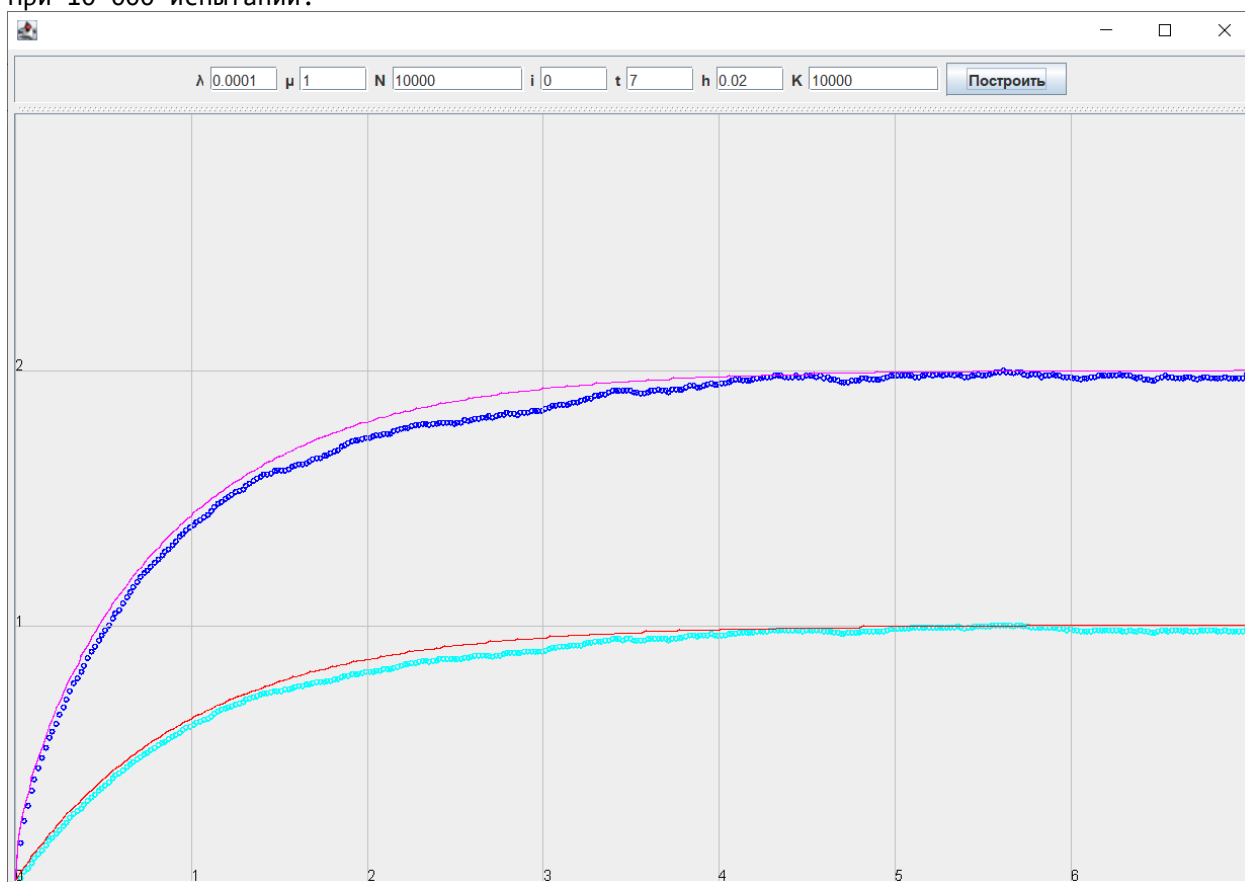
При 10 000 испытаний:



При других параметрах $i=0$, 50 испытаний



При 10 000 испытаний:



5. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Павский В. А., Павский К. В. Анализ функционирования распределенных вычислительных систем (расчет показателей надежности) : Учебное пособие / Сибирский государственный университет телекоммуникаций и информатики. – Новосибирск, 2018. – 102 с.

ПРИЛОЖЕНИЕ 1. ЛИСТИНГ ПРОГРАММЫ

```
import java.awt.BorderLayout;
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JSplitPane;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.Graphics;
import java.awt.Color;
import javax.swing.JTextField;
import javax.swing.SwingConstants;
import javax.swing.JLabel;
import org.eclipse.wb.swing.FocusTraversalOnArray;
import java.awt.Component;

class GraphicPanel2 extends JPanel {
    private Data d;
    private Data d2;
    private int width;
    private int height;
    private Graphics g;

    public GraphicPanel2(String lamb, String mu, String n, String i, String t,
String h, String k) {

        super();
        double ll = Double.valueOf(lamb);
        double mm = Double.valueOf(mu);
        double hh = Double.valueOf(h);
        int nn = Integer.valueOf(n);
        int tt = (int)(Double.valueOf(t)/Double.valueOf(h));
        int kk = Integer.valueOf(k);
        int ii = Integer.valueOf(i);

        Experiment e = new Experiment(ll, mm, nn, tt, hh, kk, ii);
        Analitic a = new Analitic(ll, mm, nn, tt, hh, kk, ii);
        this.d = e.start();
        this.d2 = a.start();
    }

    public void paint(Graphics g)
    {
        super.paint(g);
        width = getWidth(); // сохраняем текущую ширину панели
        height = getHeight(); // и высоту
        drawGraphic(g);
    }

    private void drawGrid(Graphics g, int w, int h) {
        g.setColor(Color.LIGHT_GRAY); //задаем серый цвет
        for(int x=0; x<w; x++){ // цикл от центра до правого края
            g.drawLine(x*width/w, 0, x*width/w, height); // вертикальная линия
        }
        for(int y=0; y<h; y++){ // цикл от центра до верхнего края
            g.drawLine(0, height-y*height/h, width, height-y*height/h); //
горизонтальная линия
        }
    }
}
```

```

    }
    g.setColor(Color.BLACK);
    for(int x=0; x<w; x++){ // цикл от центра до правого края
        g.drawString(Integer.toString(x), x*width/w, height);
    }
    for(int y=0; y<h; y++){ // цикл от центра до верхнего края
        g.drawString(Integer.toString(y), 0,height-y*height/h);
    }
}

public void drawGraphic(Graphics g) {

    int w = (int)(d.w);
    int h = (int) Math.max((d.h+1),(d2.h+1)) ;

    int x[] =new int[d.T];
    int y1[] = new int[d.T];
    int y2[] = new int[d.T];

    for (int i=0; i<d.T; i++) {
        g.setColor(Color.cyan);
        x[i]= (int) ( d.x[i]* width/w); //H
        y1[i]=(int)( height - (d.y1[i] * height/h));
        g.drawOval(x[i], y1[i], 4, 4);
        g.setColor(Color.blue);
        y2[i]=(int)( height - (d.y2[i] * height/h));
        g.drawOval(x[i], y2[i], 4, 4);
    }

    drawGrid(g, w,h); // рисуем сетку
    g.setColor(Color.red); //MAGENTA
    for (int i=1; i<d.T; i++) {
        g.setColor(Color.red);

        x[i]= (int) ( d2.x[i]* width/w); //H
        y1[i]=(int)( height - (d2.y1[i] * height/h));

        g.drawLine(x[i-1], y1[i-1],x[i], y1[i]);
        g.setColor(Color.MAGENTA);
        y2[i]=(int)( height - (d2.y2[i] * height/h));
        g.drawLine(x[i-1], y2[i-1],x[i], y2[i]);
    }
}
}
}

```

```

public class FMy extends JFrame {

    private JPanel contentPane;
    JSplitPane splitPane;
    private JTextField textField_mu;
    private JLabel lblNewLabel;
    private JLabel lblNewLabel_1;
    private JTextField textField_lamb;
    private JLabel lblNewLabel_2;
    private JTextField textField_N;
    private JLabel lblNewLabel_3;
    private JTextField textField_i;
    private JLabel lblNewLabel_4;
    private JTextField textField_t;
    private JLabel lblNewLabel_5;
}

```

```

private JTextField textField_h;
private JLabel lblNewLabel_6;
private JTextField textField_k;

/**
 * Launch the application.
 */
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                FMy frame = new FMy();
                frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

/**
 * Create the frame.
 */
public FMy() {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 1000, 700);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    contentPane.setLayout(new BorderLayout(0, 0));

    setContentPane(contentPane);

    splitPane = new JSplitPane();
    splitPane.setOrientation(JSplitPane.VERTICAL_SPLIT);
    contentPane.add(splitPane, BorderLayout.CENTER);

    JPanel panel = new JPanel();
    splitPane.setLeftComponent(panel);

    lblNewLabel_1 = new JLabel("\u03BB");
    panel.add(lblNewLabel_1);

    textField_lamb = new JTextField();
    textField_lamb.setText("0.0001");
    panel.add(textField_lamb);
    textField_lamb.setColumns(5);

    lblNewLabel = new JLabel("\u00B5");
    panel.add(lblNewLabel);

    textField_mu = new JTextField();
    textField_mu.setText("1");
    panel.add(textField_mu);
    textField_mu.setColumns(5);

    JButton btnNewButton = new
JButton("\u041F\u043E\u0441\u0442\u0440\u043E\u0438\u0442\u044C");
    btnNewButton.setHorizontalAlignment(SwingConstants.LEFT);
    btnNewButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {

```

```

        GraphicPanel2 GraphicPanel2= new GraphicPanel2(
            textField_lamb.getText(),
            textField_mu.getText(),
            textField_N.getText(),
            textField_i.getText(),
            textField_t.getText(),
            textField_h.getText(),
            textField_k.getText());

        splitPane.setRightComponent(GraphicPanel2);
    }
});

lblNewLabel_2 = new JLabel("N");
panel.add(lblNewLabel_2);

textField_N = new JTextField();
textField_N.setText("10000");
panel.add(textField_N);
textField_N.setColumns(10);

lblNewLabel_3 = new JLabel("i");
panel.add(lblNewLabel_3);

textField_i = new JTextField();
textField_i.setText("5");
panel.add(textField_i);
textField_i.setColumns(5);

lblNewLabel_4 = new JLabel("t");
panel.add(lblNewLabel_4);

textField_t = new JTextField();
textField_t.setText("6");
panel.add(textField_t);
textField_t.setColumns(5);

lblNewLabel_5 = new JLabel("h");
panel.add(lblNewLabel_5);

textField_h = new JTextField();
textField_h.setText("0.01");
panel.add(textField_h);
textField_h.setColumns(5);

lblNewLabel_6 = new JLabel("K");
panel.add(lblNewLabel_6);

textField_k = new JTextField();
textField_k.setText("1000");
panel.add(textField_k);
textField_k.setColumns(10);
panel.add(btnNewButton);
panel.setFocusTraversalPolicy(new FocusTraversalOnArray(new
Component[]{lblNewLabel_1, textField_lamb, lblNewLabel, textField_mu, lblNewLabel_2,
textField_N, lblNewLabel_3, textField_i, lblNewLabel_4, textField_t, lblNewLabel_5,
textField_h, lblNewLabel_6, textField_k, btnNewButton}));

}
package pkg;

```

```

public class Experiment {
    private double lambda;
    private double mu;
    private int N;
    private int T;
    private double h;
    private int K;
    private int i;

    public Experiment(double lambda, double mu, int n, int t, double h, int k, int
i) {
        super();
        this.lambda = lambda;
        this.mu = mu;
        N = n;
        T = t;
        this.h = h;
        K = k;
        this.i = i;
    }

    //метод моделирования рождения гибели
    public Data start() {
        Data data = new Data(T);
        double listM2[] = new double[T];
        double listM[] = new double[T];
        double listD[] = new double[T];

        for (int k = 0; k < K; k++) {
            int listN[] = new int[T];
            listN[0] = N-i;

            for (int i = 1; i < T; i++) {
                double t1 = i * h;
                double t2 = t1 + h;
                listN[i] = listN[i - 1];
                if (listN[i] > 0) {
                    double R = Math.exp(-lambda * listN[i] * h);
                    double r = Math.random();
                    if (r > R) {
                        listN[i]--;
                    }
                }
                if (listN[i] < N) {
                    double R = Math.exp(-mu * (N - listN[i]) * h);
                    double r = Math.random();
                    if (r > R) {
                        listN[i]++;
                    }
                }
            }
            for (int i = 0; i < T; i++) {
                listM[i] += listN[i];
                listM2[i] += (listN[i] * listN[i]);
            }
        }
        data.h = 0;
        for (int i = 0; i < T; i++) {
            listM[i] /= K;
            listD[i] = listM2[i] / K - (listM[i] * listM[i]);
            data.x[i] = i * h;
            data.y1[i]=(N - listM[i]);
        }
    }
}

```



```

        data.y2[i]=(N - listM[i])+Math.sqrt( listD[i]);
        data.h = Math.max(data.h , data.y2[i]);
    }

    data.w = T*h;
    return data;
}

package pkg;

public class Analitic {
    private double lambda;
    private double mu;
    private int N;
    private int T;
    private double h;
    private int i;

    public Analitic(double lambda, double mu, int n, int t, double h, int k, int
i) {
        super();
        this.lambda = lambda;
        this.mu = mu;
        N = n;
        T = t;
        this.h = h;
        this.i = i;
    }

    public Data start() {
        Data data = new Data(T);
        for (int t=0; t<T; t++) {
            data.x[t]= t*h;
            data.y1[t] = ((lambda*N)/(lambda+mu)) +
                ((mu*i-lambda*(N-i))/(lambda+mu))*
                Math.exp(-1.0*(lambda+mu)*data.x[t] );
            double d = ((N*lambda*mu)/((lambda+mu)*(lambda+mu))) +
                ((lambda*lambda*(N-i)+mu*(i*mu-
lambda*N))/((lambda+mu)*(lambda+mu)))*
                Math.exp(-(lambda+mu)* data.x[t] ) -
                ((lambda*lambda*(N-
i)+i*mu*mu)/((lambda+mu)*(lambda+mu)))*
                Math.exp(-2.0*(lambda+mu)* data.x[t] );
            data.y2[t]=data.y1[t]+Math.sqrt(d);
            //data.y2[t]=data.y1[t]+d;
            data.h=Math.max(data.h, data.y2[t]);
            //System.out.println();
        }
        return data;
    }
}

package pkg;

public class Data{
    public Data(int T) {
        super();
        this.T=T;
        this.x = new double[T];
        this.y1 =new double[T];
        this.y2 =new double[T];
    }
}

```

```
public int T;  
public double x[];  
  
public double w;  
public double h;  
public double y1[];  
public double y2[];
```

```
}
```