Визуализация функций распределий отказов на наработке t.

Ведущая функция потока

Согласно определению, ведущая функция потока Ω это ряд, с вообще говоря бесконечным числом членов последовательности, вида $$ \Omega(t) = \sum_{i = 1}^{\infty} F_i(t)$$

где $F_i(t)$ вероятность i-ого отказа на наработке t. Также, параметр потока отказов будет представляться в виде: $$ \omega(t) = \frac{d\Omega(t)}{dt}$$В свою очередь, для систем с схожим распределением всех отказов на наработке t будет справедлив принцип композиции функция распределения, что выражается сверткой вида $$ F_i(t) = \int_{0}^{t}F_{i-1}(t - \tau)dF_1(\tau)$$

откуда следует, что зная закон распределния 1 отказа на заданной наработке, можно итеративно восстановить все законы, вплоть до i-ого.

Очевидно, что $ dF(t) = f(t)dt$, и тогда подставляя в предыдущую формулу получим $$ F_i(t) = \int_{0}^{t}F_{i-1}(t - \tau)f_1(\tau)d\tau$$

Экспонента. Закон распределения вида $1 - e^{-\lambda t}$

Пусть известно, что все плотности распределения i-ых отказов распределены экспоненциально и справделиво $$f_1(t) = f_2(t) = f_3(t) = f_4(t) =... = f_i(t)$$

Поставим задачу - визуализировать законы распределния i-ых отказов, а также получившиеся $\Omega(t)$ и $\omega(t)$.

1.

Получим все распределния и визуализируем их. Я делал циклическое вычисление свертки, ограничился F_6(t), если нижний индекс распределения распределения принять в качестве порядка свертки, то начиная с 7-8 процесс вычисления занимает приличное время. Ниже код

In [ ]:

```
syms F t F_sum tau arr;

lambda = 0.05;

F(t) = 1 - exp(-lambda*.5*t);
%F_sum = F;
n = 6 ;
%input('Enter n: ');
for i = 1:n
     arr(i) = F(t);
     F(t) = int(F(t-tau)*lambda*.5*exp(-lambda*.5*tau), tau, 0,t);
end
```

по итогу выполнения имею массив символьных функций. Визуализировав его получу

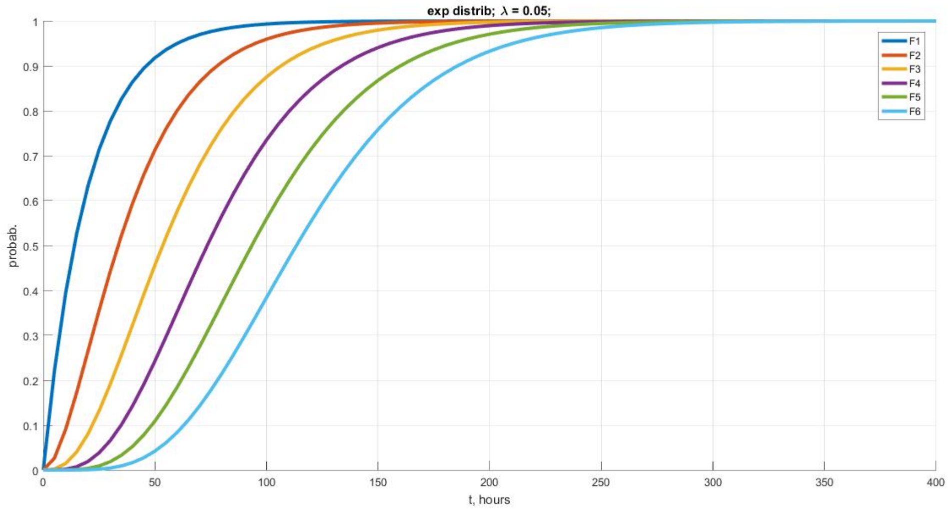
In [ ]:

```
F1(t) = arr(1);
F2(t) =arr(2);
F3(t) =arr(3);
F4(t) =arr(4);
F5(t) =arr(5);
F6(t) =arr(6);
% %
hold on
plot([0:5:400], F1([0:5:400]), 'LineWidth', 3);
title('exp distrib; \lambda = 0.05;');
grid on
plot([0:5:400], F2([0:5:400]), 'LineWidth', 3);
plot([0:5:400], F3([0:5:400]), 'LineWidth', 3);
plot([0:5:400], F4([0:5:400]), 'LineWidth', 3);
plot([0:5:400], F5([0:5:400]), 'LineWidth', 3);
plot([0:5:400], F6([0:5:400]), 'LineWidth', 3);
xlabel('t, hours')
ylabel('probab.')
legend('F1','F2','F3','F4','F5','F6');
```

In [2]:

```
from IPython.display import Image, display
Image('Image/exp_lam0.05_n6_axis.jpg')
```

Out[2]:



Соответственно, просуммровав все функции распределения получу ведущую функцию потока

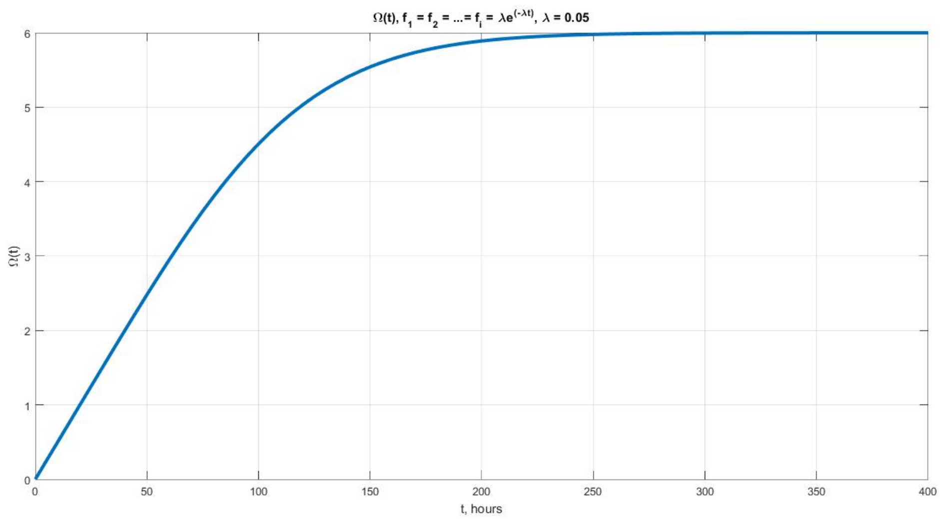
In [ ]:

```
Omega(t) = sum(arr);
figure();
plot([0:5:400], Omega([0:5:400]),'LineWidth', 3);
title('\Omega(t), f_1 = f_2 = ...= f_i = \lambdae^{(-\lambda\t)}, \lambda = 0.05');
xlabel('t, hours');
ylabel('\Omega(t)');
grid on
```

In [5]:

```
from IPython.display import Image, display
Image('Image/exp_lam0.05_n6_Omega.jpg')
```

Out[5]:



Затем, вычисляю $ \omega(t) = \frac{d\Omega(t)}{dt}$

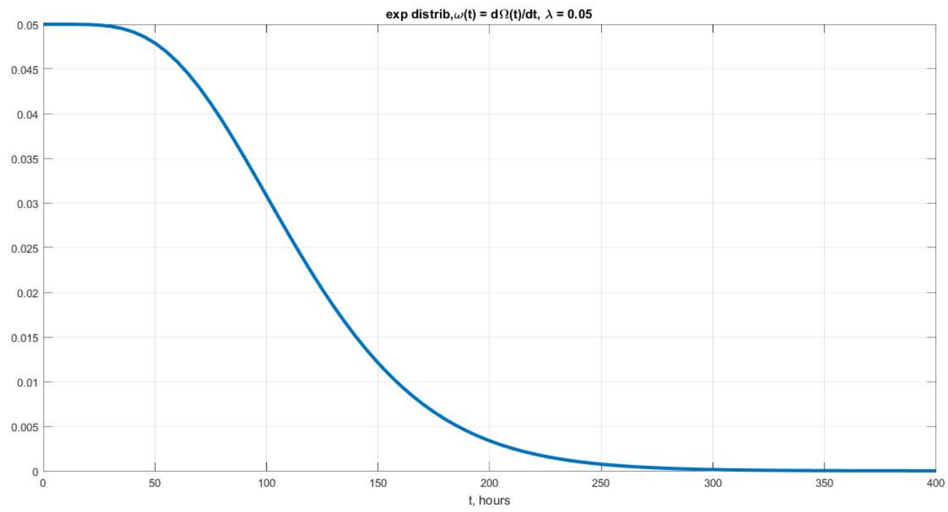
In [ ]:

```
figure();
omega(t) = diff(Omega(t),t);
plot([0:5:400], omega([0:5:400]),'LineWidth', 3);
title('exp distrib,\omega(t) = d\Omega(t)/dt, \lambda = 0.05');
xlabel('t, hours');
grid on
```

In [6]:

```
from IPython.display import Image, display
Image('Image/exp_lam0.05_n6_omega_small.jpg')
```

Out[6]:

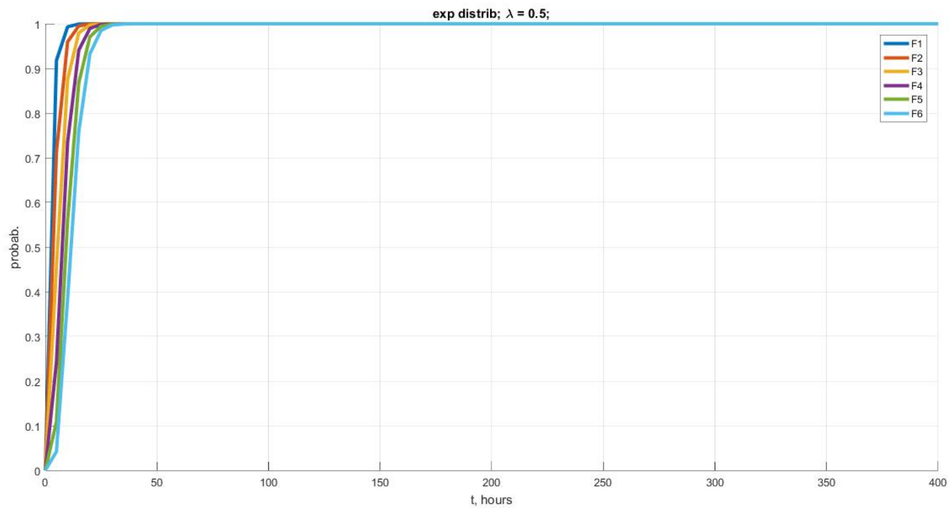


Повторив все для другого параметра λ, к примеру $\lambda $ = 0.5

In [7]:

```
Image('Image/exp_lam0.5_n6_axis.jpg')
```

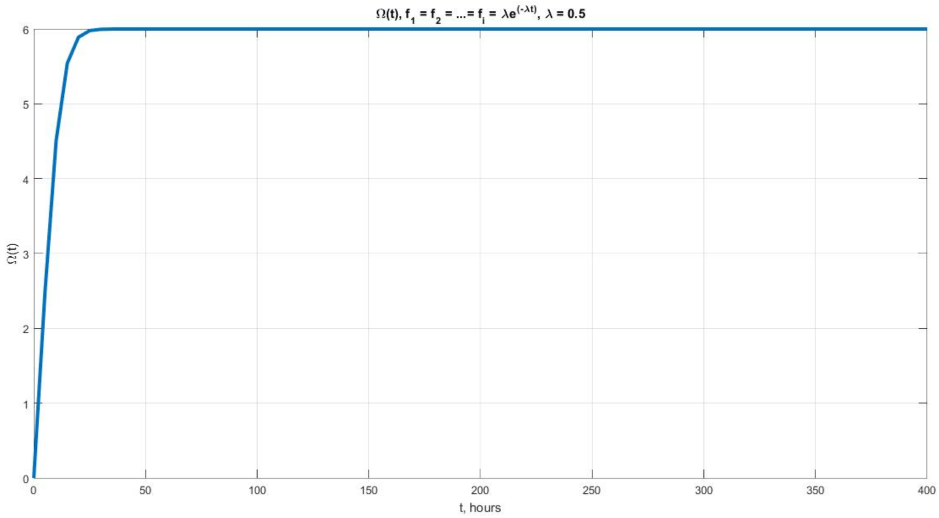
Out[7]:



In [8]:

```
Image('Image/exp_lam0.5_n6_Omega.jpg')
```

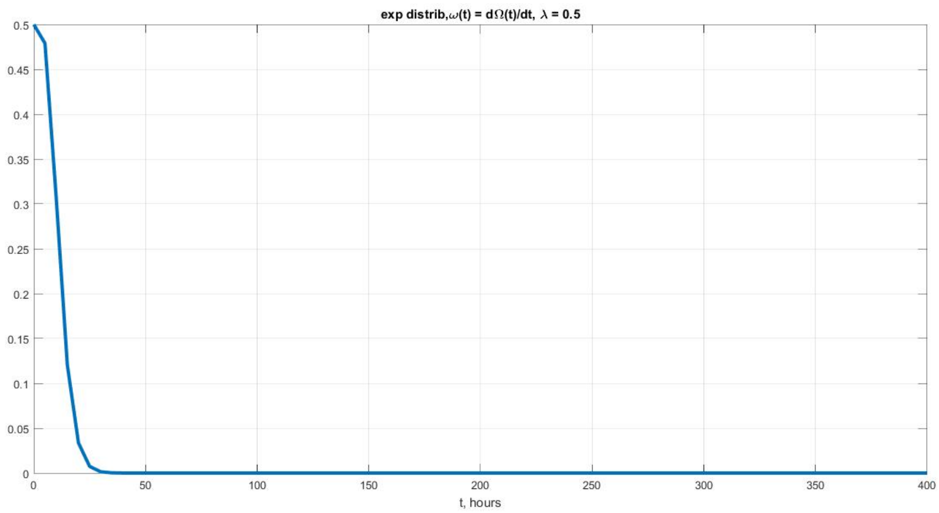
Out[8]:



In [9]:

```
Image('Image/exp_lam0.5_n6_omega_small.jpg')
```

Out[9]:



Полученные зависимости наглядно иллюстрируют тот факт, что при увеличени параметра экс.распределения λ наблюдается "скучкованность" ненулевых значений функциий распределения в окрестности нуля.

2. Нормальное распределение. ПНФ $f(t) = \frac{1}{\sigma\sqrt{(2\pi)}} e^{-\frac{(t - t_{av})^2}{2\sigma^2}}$

Решим этот же ряд задач с сохранением всех условий, указанных выше. За исключением того, что распределение будет теперь описываться нормальным законом.

Делал уже не для 6 , а для 4 отказов. Для 4 вычисление сверток + отрисовка = 1.5 часа. Для 6 я не дождался.

Пусть σ = 100, а мат.ожидание $M[t] = t_{av}$ = 50

In [ ]:

```
sigma = 100;
average = 50;

syms F_n t_n F_sum_n tau_n arr_n;



n = 4;

F_n(t_n) = (14373410442865825*2^(1/2)*pi^(1/2)*(erf(20000^(1/2)/400) + erf((20000^(1/2)
*(t_n - 50))/20000)))/72057594037927936;

%int(1/(sqrt(2*pi*sigma^2))*exp(-((t_n -average)^2)/(2*sigma^2)), t_n,0,t_n);
%F_sum = F;
%n = input('Enter n: ');
for i = 1:n
     arr_n(i) = F_n(t_n);
     F_n(t_n) = int(F_n(t_n-tau_n)*1/(sqrt(2*pi*sigma^2))*exp(-(((t_n-tau_n) -average)^
2)/(2*sigma^2)), tau_n, 0,t_n);
end
```

Первоначальный вид зависимости был вычислен в командной строке матлаба, как $F_n(t_n)$ = int(f(t_n),t_n,0,t_n)

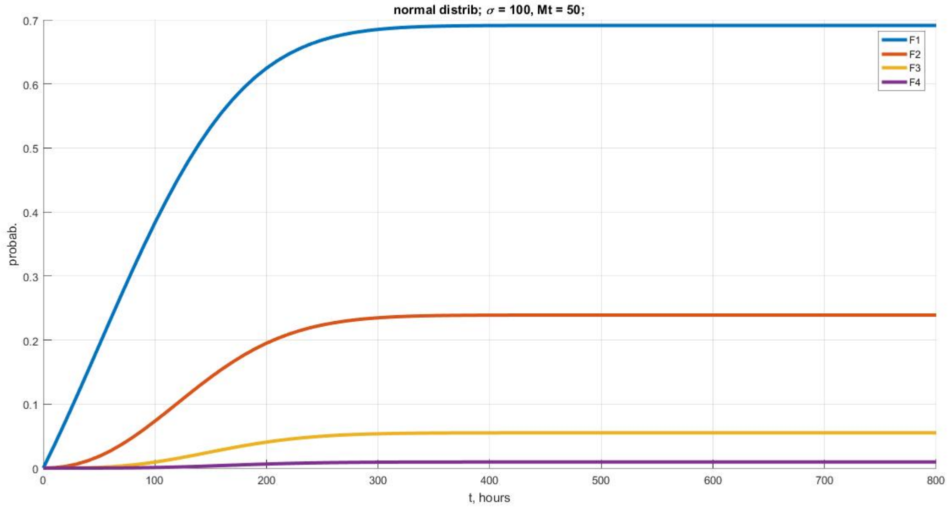
In [ ]:

```
F1(t_n) =arr_n(1);
F2(t_n) =arr_n(2);
F3(t_n) =arr_n(3);
F4(t_n) =arr_n(4);
%F5(t_n) =arr_n(5);
%F6(t_n) =arr_n(6);


hold on
plot([0:10:800], F1([0:10:800]), 'LineWidth', 3);
title('normal distrib; \sigma = 100, Mt = 50;');
grid on
plot([0:10:800], F2([0:10:800]), 'LineWidth', 3);
plot([0:10:800], F3([0:10:800]), 'LineWidth', 3);
plot([0:10:800], F4([0:10:800]), 'LineWidth', 3);
%plot([0:10:800], F5([0:10:800]), 'LineWidth', 3);
%plot([0:10:800], F6([0:10:800]), 'LineWidth', 3);
xlabel('t, hours')
ylabel('probab.')
legend('F1','F2','F3','F4');%,'F5','F6');
```

In [10]:

```
Image('Image/norm_sig100_Mt50_n6_axis.jpg')
```

Out[10]:



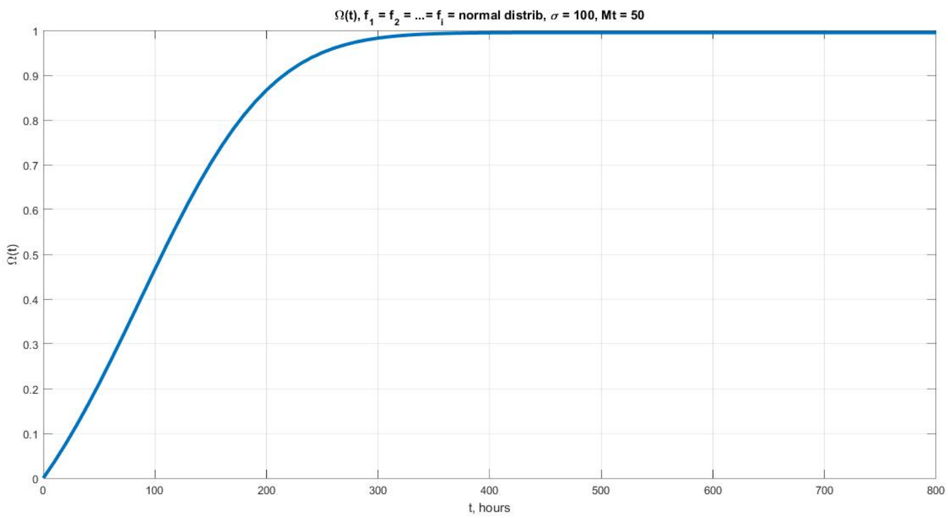
In [ ]:

```
%
Omega(t_n) = sum(arr_n);
figure();
plot([0:10:800], Omega([0:10:800]),'LineWidth', 3);
title('\Omega(t), f_1 = f_2 = ...= f_i = normal distrib, \sigma = 100, Mt = 50');
xlabel('t, hours');
ylabel('\Omega(t)');
grid on
%
```

In [11]:

```
Image('Image/norm_sig100_Mt50_n6_Omega.jpg')
```

Out[11]:



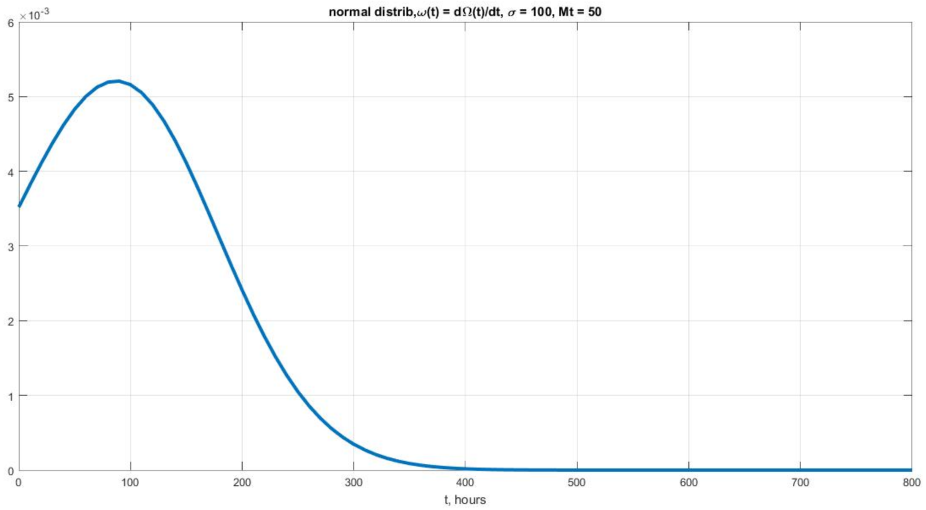
In [ ]:

```
figure();
omega(t_n) = diff(Omega(t_n),t_n);
plot([0:10:800], omega([0:10:800]),'LineWidth', 3);
title('normal distrib,\omega(t) = d\Omega(t)/dt, \sigma = 100, Mt = 50');
xlabel('t, hours');
grid on
```

In [12]:

```
Image('Image/norm_sig100_Mt50_n6_omega_small.jpg')
```

Out[12]:



Без условия равенства плотностей распределения

Цель - решить эту же задачу, только с условием того, что $$f_1(t) \neq f_2(t) \neq f_3(t) \neq f_4(t) \neq... \neq f_i(t)$$

Предложение - взять в основу экспоненциальный закон распределения (на нем все быстро работает), также циклически вычислять свертки, но на каждой итерации случайно зашумлять параметр распределения и изменять амплитуды.

Для определенности, возьму параметр $\epsilon = [0,1]$ (в коде использую rand(1)). Тогда с его помощью буду случайно изменять параметры распределний в свертках, начиная со 2-ой. В итоге

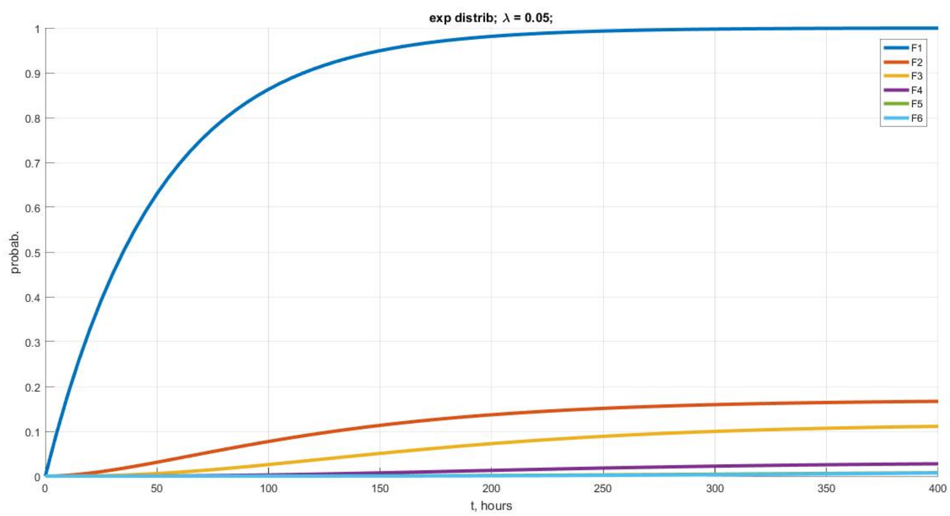
In [ ]:

```
syms F t F_sum tau arr;

lambda = 0.05;
eps = input('Enter epsilon for disequality:');
F(t) = 1 - exp(-lambda*rand(1)*.5*t);
%F_sum = F;
n = 6 ;
%input('Enter n: ');
for i = 1:n
     arr(i) = F(t);
     F(t) = int(F(t-tau)*lambda*rand(1)*.5*exp(-lambda*rand(1)*.5*tau)*rand(1), tau, 0,
t);
end

    F1(t) = arr(1);
    F2(t) =arr(2);
    F3(t) =arr(3);
    F4(t) =arr(4);
    F5(t) =arr(5);
    F6(t) =arr(6);
    % %
    hold on
    plot([0:5:400], F1([0:5:400]), 'LineWidth', 3);
    title('exp distrib; \lambda = 0.05;');
    grid on
    plot([0:5:400], F2([0:5:400]), 'LineWidth', 3);
    plot([0:5:400], F3([0:5:400]), 'LineWidth', 3);
    plot([0:5:400], F4([0:5:400]), 'LineWidth', 3);
    plot([0:5:400], F5([0:5:400]), 'LineWidth', 3);
    plot([0:5:400], F6([0:5:400]), 'LineWidth', 3);
    xlabel('t, hours')
    ylabel('probab.')
    legend('F1','F2','F3','F4','F5','F6');
```

In [13]:

```
Image('Image/exp_lam0.05_n6_axis_random.jpg')
```

Out[13]:



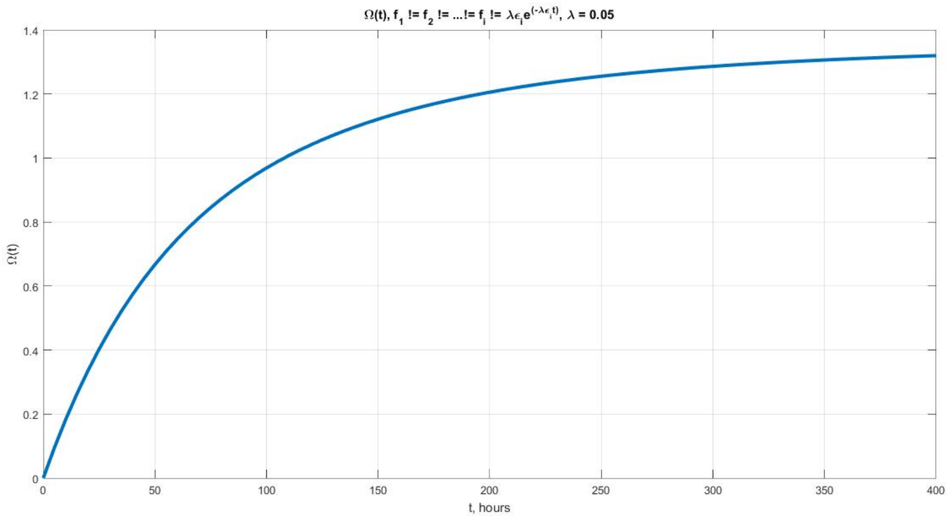
In [ ]:

```
Omega(t) = sum(arr);
figure();
plot([0:5:400], Omega([0:5:400]),'LineWidth', 3);
title('\Omega(t), f_1 != f_2 != ...!= f_i != \lambda\epsilon_ie^{(-\lambda\epsilon_i
t)}, \lambda = 0.05');
xlabel('t, hours');
ylabel('\Omega(t)');
grid on

figure();
omega(t) = diff(Omega(t),t);
plot([0:5:400], omega([0:5:400]),'LineWidth', 3);
title('exp distrib,\omega(t) = d\Omega(t)/dt, \lambda = 0.05');
xlabel('t, hours');
grid on
```

In [14]:

```
Image('Image/exp_lam0.05_n6_Omega_random.jpg')
```

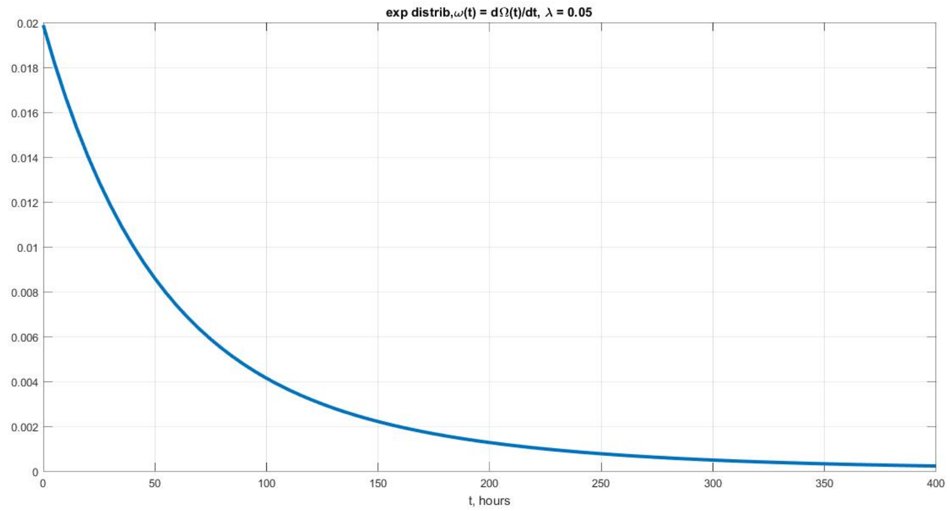
Out[14]:



In [15]:

```
Image('Image/exp_lam0.05_n6_omega_small_random.jpg')
```

Out[15]:

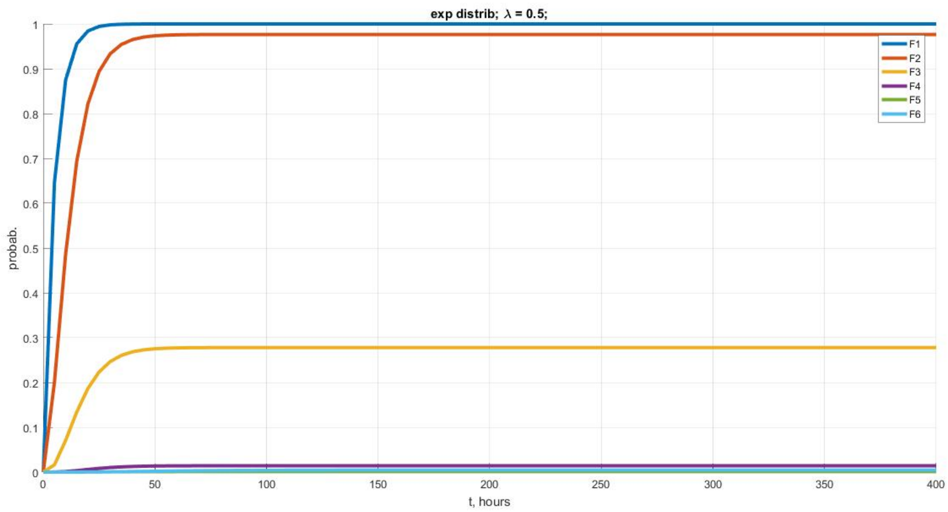


Повторив все для другого параметра λ, к примеру $\lambda $ = 0.5

In [21]:

```
Image('Image/exp_lam0.5_n6_axis_random.jpg')
```

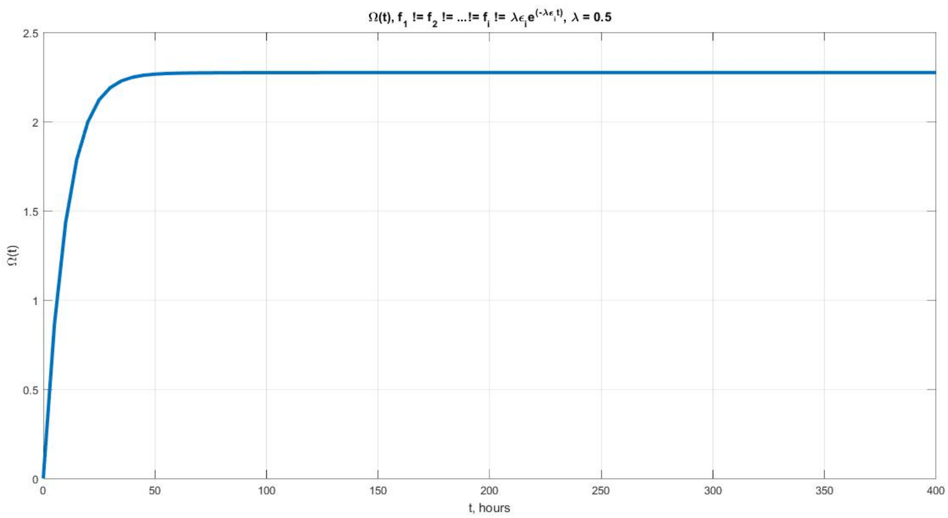
Out[21]:



In [20]:

```
Image('Image/exp_lam0.5_n6_Omega_random.jpg')
```

Out[20]:



In [19]:

```
Image('Image/exp_lam0.5_n6_omega_small_random.jpg')
```

Out[19]: