Вычисление одномерных, двумерных и интегралов с переменным верхним пределом

q = **quad** *(f, a, b)*
q = **quad** *(f, a, b, tol)*
q = **quad** *(f, a, b, tol, sing)*
[q, ier, nfun, err] = **quad** *(…)*

***quad* – низкая точность с разрывной функцией**
***quadv* – средняя точность с плавной функцией *quadl* – средняя точность**
***quadgk* – колебательные функции и бесконечные границы**
***quadcc* – высокая точность с функциями с особенностями**

Вычислить интеграл:

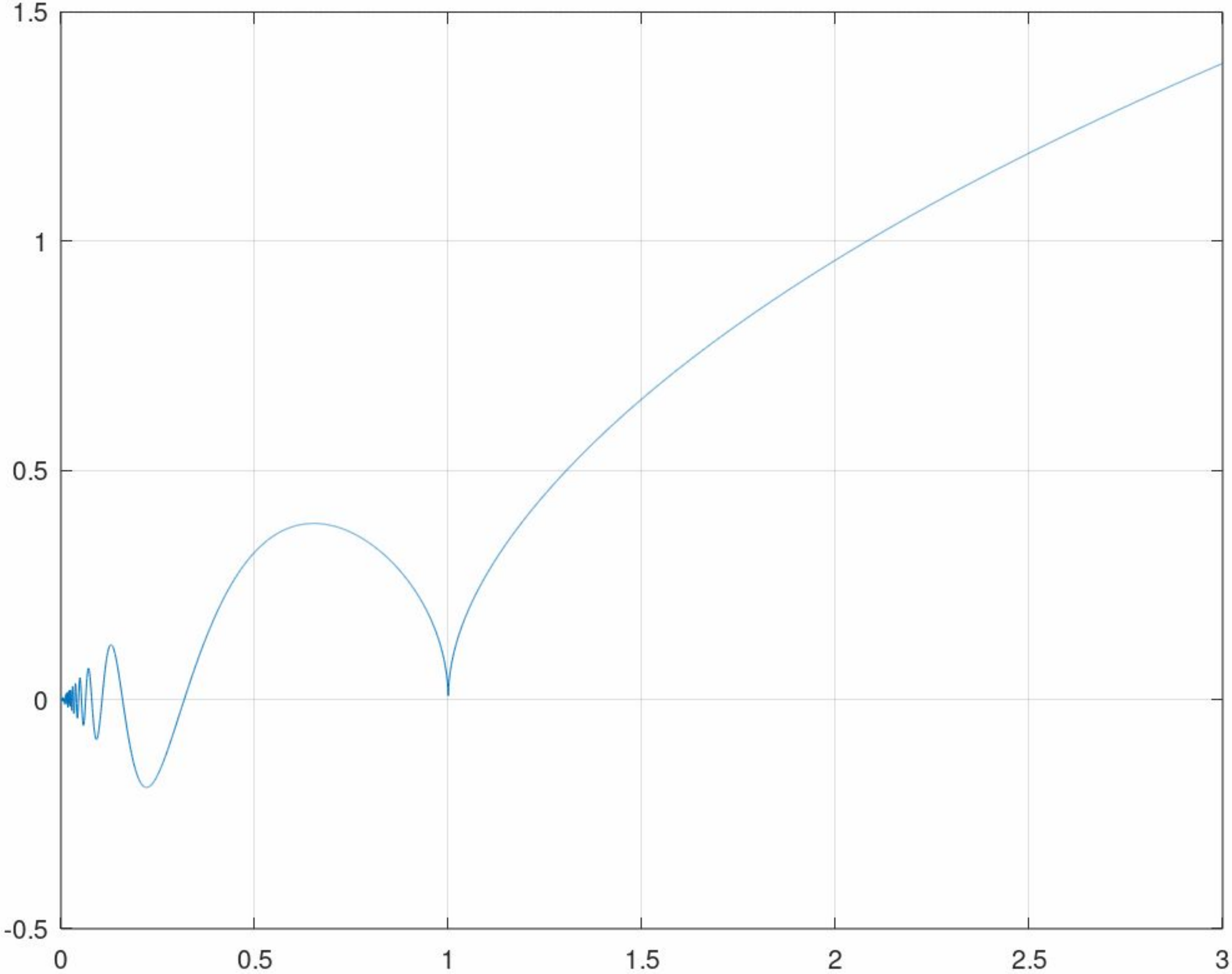
$$y = \int_0^3 x \cdot \sin(1/x)\sqrt{|1-x|}\,dx$$

Напишем анонимную функцию:

f=@(x)x .* sin (1./x) .* sqrt (abs (1 - x))

Построим ее график в отрезке (0,3)

```
>> x=0.0001:0.001:3;
>> plot(x,f(x))
```


1.5
1
0.5
0
-0.5
0
0.5
1
1.5
2
2.5
3

```
>> f=@(x)x .* sin (1./x) .* sqrt (abs (1 - x))
 y=quad(f,0,3)
 ABNORMAL RETURN FROM DQAGP
y =  1.981941224557954
>> y=quadv(f,0,3)
warning: division by zero
y =  NaN
>> y=quadl(f,0,3)
error: max_recursion_depth exceeded
>> y=quadgk(f,0,3)
y =  1.981941211175243
>> y=quadcc(f,0,3)
y =  1.981941198153926
```

Разные способы вычисления

Особенности интегрирования

```
>> f=@(x)x.^-0.9
>> y=quad(f,0,1)
y =  9.99999999999792
>> y=quadv(f,0,1)
warning: quadv: minimum step size reached -- possible singular integral
y =  9.999992801849245
>> y=quadl(f,0,1)
error: max_recursion_depth exceeded
>> y=quadgk(f,0,1)
warning: quadgk: non finite integrand encountered
y =  9.789371547302718
>> y=quadcc(f,0,1)
y =  9.999999260395999
```

Вычислить интеграл:

$$y = \int_{0}^{\infty} \frac{1}{\sqrt{x} \cdot (x+1)} dx$$

```
>> f=(@(x) 1 ./ (sqrt (x) .* (x + 1))
>> quadgk(f, 0, Inf)
ans =  3.14159265358979
```

Интегрирование таблично заданных функций

```
x=0:0.1:pi/2;
y=sin(x);
z=trapz (x, y) % Интегрирование методом
          %  трапеций
z=0.9966636
```

Рассмотрим другой способ интегрирования

```
p=polyfit(x,y,5); %Интерполяция полиномом
p1=polyint(p);   % Производная полинома
```

Вычисление интеграла с переменным верхним пределом

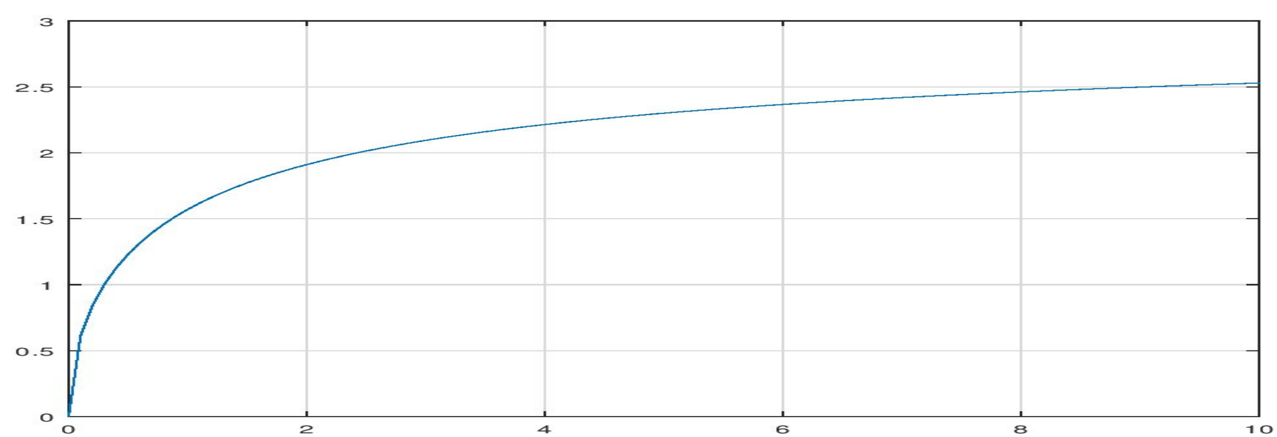
Для вычисления такого интеграла надо составить файл-функцию вида:

```
function y=Fax(f,a,x)
  % Вычисление интеграла с переменным
   % верхним пределом
  n=length(x);
  y=zeros(1,n); %Выделение памяти
  for i=1:n       % Заполнение массива в цикле
    y(i)=quadgk(f,a,x(i));
  endfor
endfunction
```

Вычислим интеграл вида: $\displaystyle\int_0^x \frac{1}{\sqrt{y}\cdot(y+1)}dy$

В командной строке вводим анонимную подынтегральную функцию:

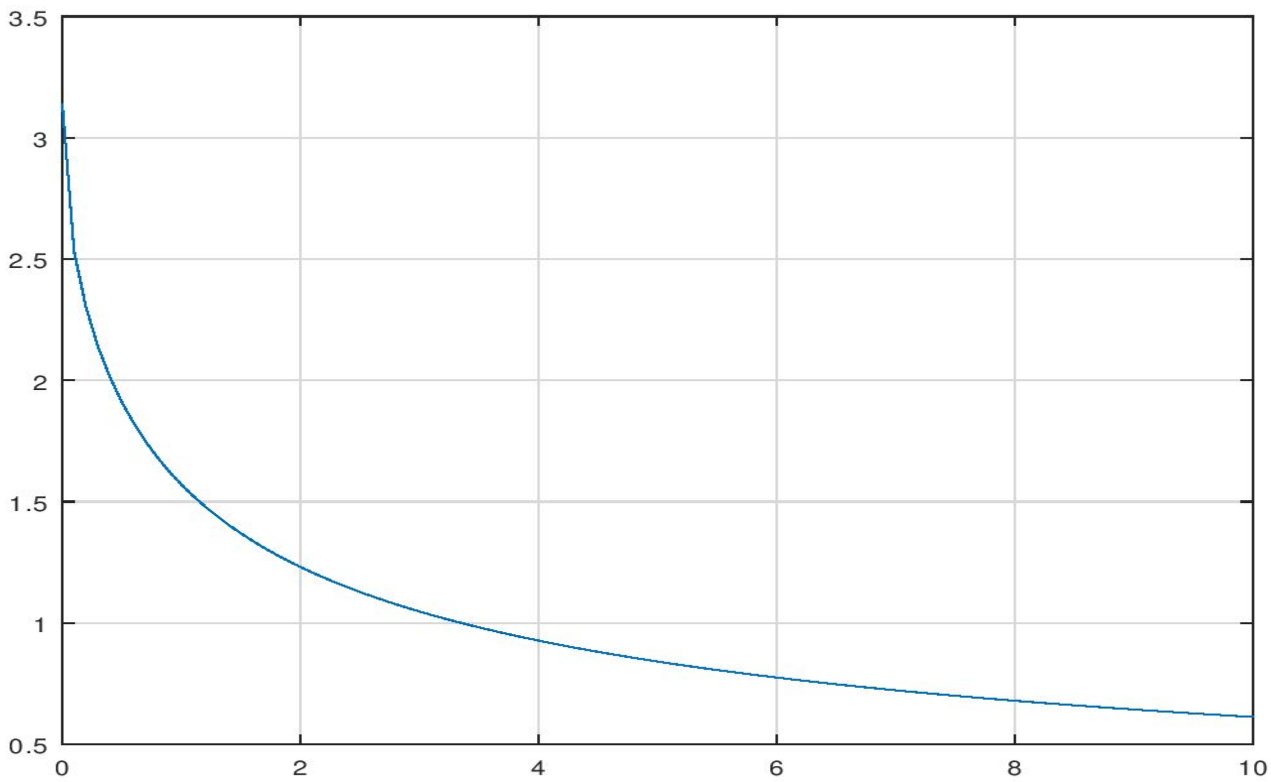
```
>> f=@(z) 1 ./ (sqrt (z) .* (z + 1));
>> x=0:0.1:10;
>> y=Fax(f,0,x);
>> plot(x,y)
```



Вычислить интеграл с переменным нижним пределом:

$$\int_x^{\infty} \frac{1}{\sqrt{y}\cdot(y+1)}dy$$

Самостоятельно



Двумерные интегралы

Двумерные интегралы можно вычислять приведением двумерного интеграла к повторному (двумя способами). Особенностями двумерных интегралов является то, что область интегрирования может не являться прямоугольной. Рассмотрим вначале интегрирование в прямоугольной области.

$$\int_0^1 \int_2^3 \sin(\pi \cdot x \cdot y)\sqrt{x \cdot y}\ dxdy$$

Вначале напишем функцию, вычисляющую внутренний интеграл по ***x*** при различных значениях *y*:

```
function q = gint(y,a,b)
  q = ones (size (y)); %Распределяем память
  ff=@(x,y)sin (pi*x.*y) .* sqrt (x.*y);
  for i = 1:length (y)
    f = @(x)ff(x,y(i));
    q(i) = quadgk (f, a, b);
  endfor
endfunction
```

Анонимную функцию объявляем здесь

В командной строке найдем двумерный интеграл:

```
>> ggint=@(x)gint(x,2,3);In=quadgk(ggint,0,1)
In =  0.045377
```

Этот алгоритм осуществлен в функции dblquad для интегралов двух переменных. Эквивалент этого процесса осуществлен в **triplequad** для интегралов трех переменных. Для примера, вызовем **dblquad** как показано ниже:

```
>> f=@(x,y)sin(pi*x.*y).*sqrt(x.*y);
>> In=dblquad (f, 2, 3, 0, 1)
In =  0.045377
```

dblquad *(f, xa, xb, ya, yb)*
dblquad *(f, xa, xb, ya, yb, tol)*
dblquad *(f, xa, xb, ya, yb, tol, quadf)*
dblquad *(f, xa, xb, ya, yb, tol, quadf, …)*
f – указатель на функцию, встроенная функция, или строка, содержащие имя функции. Функция **f** должна иметь форму **z = f(x,y)** где **x** - вектор и **y** - скаляр. Она должна возвращать вектор той же самой длины и ориентации как **x**. **xa**, **ya** и **xb**, **yb** - нижний и верхний пределы интегрирования для **x** и **y** соответственно.
Дополнительный аргумент **tol** определяет абсолютную точность вычисления каждого подинтеграла. Значение по умолчанию –
 1e-6.
Дополнительный аргумент **quadf** определяет какую основную функцию интегрирования использовать. Любой выбор **quad** доступен и по умолчанию - **quadcc**.
Дополнительные аргументы функции **f** записываются после

Примеры различных вызовов функции dblquad:

```
>> f=@(x,y)sin(pi*x.*y).*sqrt(x.*y);
>> format long;
>> In=dblquad (f, 2, 3, 0, 1)
In =    4.537731171544740e-02
In=dblquad (f, 2, 3, 0, 1,1e-8) % Погрешность уменьшили
In =    4.537731134206942e-02
>> f=@(x,y,a)sin(pi*x.*y+a).*sqrt(x.*y+a); % a - параметр
% функции
>> In=dblquad (f, 2, 3, 0, 1,1e-8,[],0) % a=0
In =    4.537731134206942e-02
>> In=dblquad (f, 2, 3, 0, 1,1e-8,[],2*pi) % a=2*pi
In =    3.012549854177141e-01
```

Рекурсивный алгоритм для интегрирования, представленный выше, называется повторным интегрированием. Существует отдельный двумерный метод интегрирования, который осуществлен в функции **quad2d**.

Эта функция выполняет интегрирование, подразделяя область интегрирования в прямоугольные области и выполняет отдельное интегрирование над этими областями. Области дальше подразделяются в еще меньшие области, чтобы достигать желаемой точности. Этот метод функционирует быстрее, чем **dblquad**.

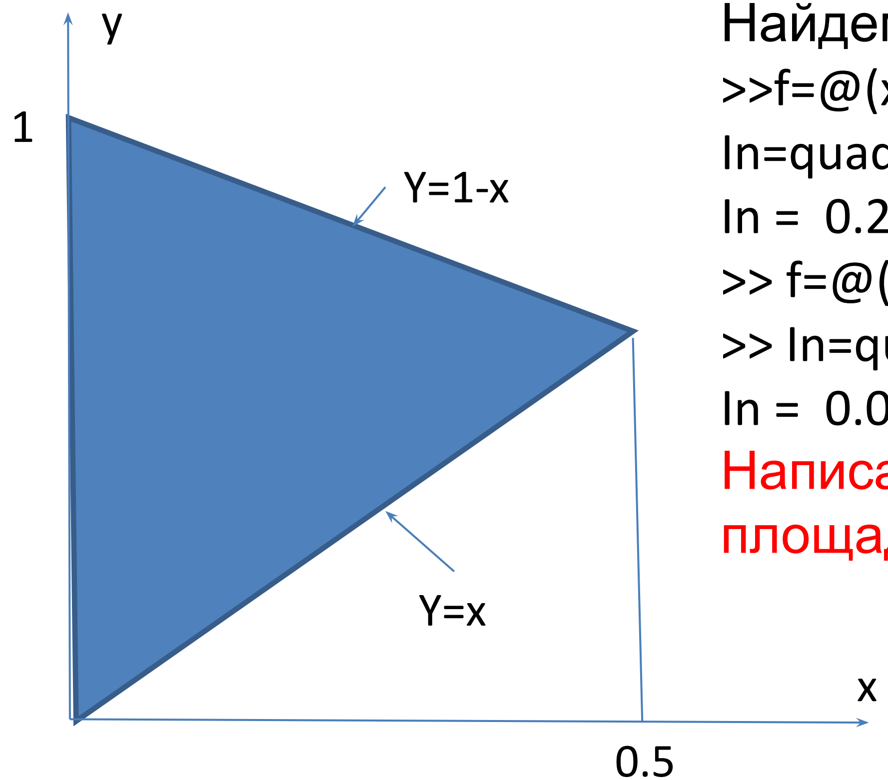
q = **quad2d** *(f, xa, xb, ya, yb)*
q = **quad2d** *(f, xa, xb, ya, yb, prop, val, …)*
[q, err, iter] = **quad2d** *(…)*

Вычисление двумерного интеграла функции f, используя адаптивный квадратурный метод над двумерной областью определенной **xa**, **xb**, **ya**, **yb**, используя разбиение на прямоугольные области. К тому же, **ya** и **yb** могут быть скалярными функциями переменной **x**, учитывающими интегрирование над непрямоугольными областями.

f – указатель на функцию, встроенная функция, или строка, содержащие имя функции. Функция **f** должна быть в виде **z = f(x,y),** где **x** - вектор и **y** - скаляр. Она должна возвращать вектор той же

```
>> f=@(x,y)sin(pi*x.*y).*sqrt(x.*y);
>> In=quad2d(f,2,3,0,1)
In =    4.537731177584814e-02
```

Рассмотрим интегрирование по области в двумерном пространстве:



Найдем площадь треугольника

```
>>f=@(x,y) 1;
In=quad2d(f,0,0.5,@(x)x,@(x)1-x)
In =  0.25000
>> f=@(x,y)sin(pi*x.*y).*sqrt(x.*y);
>> In=quad2d(f,0,0.5,@(x)x,@(x)1-x)
In =  0.022185
```

Написать функцию вычисления площади круга радиуса R

Задание для самостоятельной работы

Пусть необходимо построить график функции, которая задается следующей формулой:

$$d(f_1, f_2) = \frac{\left| \sum_{k=1}^{N} \frac{1}{\mu_k} \sum_{i=1}^{N} z_i \psi^*_{i,k} \sum_{j=1}^{N} s^*_j \psi_{j,k} \right|^2}{\sum_{k=1}^{N} \sum_{m=1}^{N} \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{l=1}^{N} \sum_{n=1}^{N} \frac{1}{\mu_k \mu_m} R_{i,l} \psi_{i,k} \psi^*_{l,m} s^*_j s_n \psi_{j,k} \psi^*_{n,m}}$$

$$z_i = \frac{1}{\sqrt{N}} e^{j2\pi \cdot f_1 \cdot T \cdot i}, \ s_i = \frac{1}{\sqrt{N}} e^{j2\pi \cdot f_2 \cdot T \cdot i} \quad R_{i,j} = Pe^{-\alpha_2((i-j)T)^2}, \ \alpha_2 = \frac{(\pi \cdot df_2)^2}{4\ln(2)}$$

$\mu_k, \psi_{i,k}$ Собственные значения и вектора матрицы $K_{i,j} = e^{-\alpha_1((i-j)T)^2}, \ \alpha_1 = \frac{(\pi \cdot df_1)^2}{4\ln(2)}$

Исходные данные

- N=16
- f1=0:1023
- f2=0:7
- T=0.002
- df1=1
- df2=3
- P=1000

Выдать промежуточные данные:
Построить график N/2 –й строки матрицы К.
Построить график первых 4 собственных векторов матрицы К, отвечающих максимальным по величине собственным значениям.

Ввод данных реализовать в режиме диалога inputdlg

Выполнение домашнего задания является необходимым, но недостаточным условием для получения зачета.
Отчет по домашнему заданию посылать мне на почту в виде документа Word, оформленного в соответствии с требованиями к курсовой работе.