Московский авиационный институт
(национальный исследовательский университет)

Институт №8 «Информационные технологии и прикладная математика»

Кафедра 806 «Вычислительная математика и программирование»

Курсовая работа по дисциплине «Численные методы»

Студент: Молчанов Владислав
Группа: М8О-408Б-20
Преподаватель: Пивоваров Д.Е.

Москва, 2023

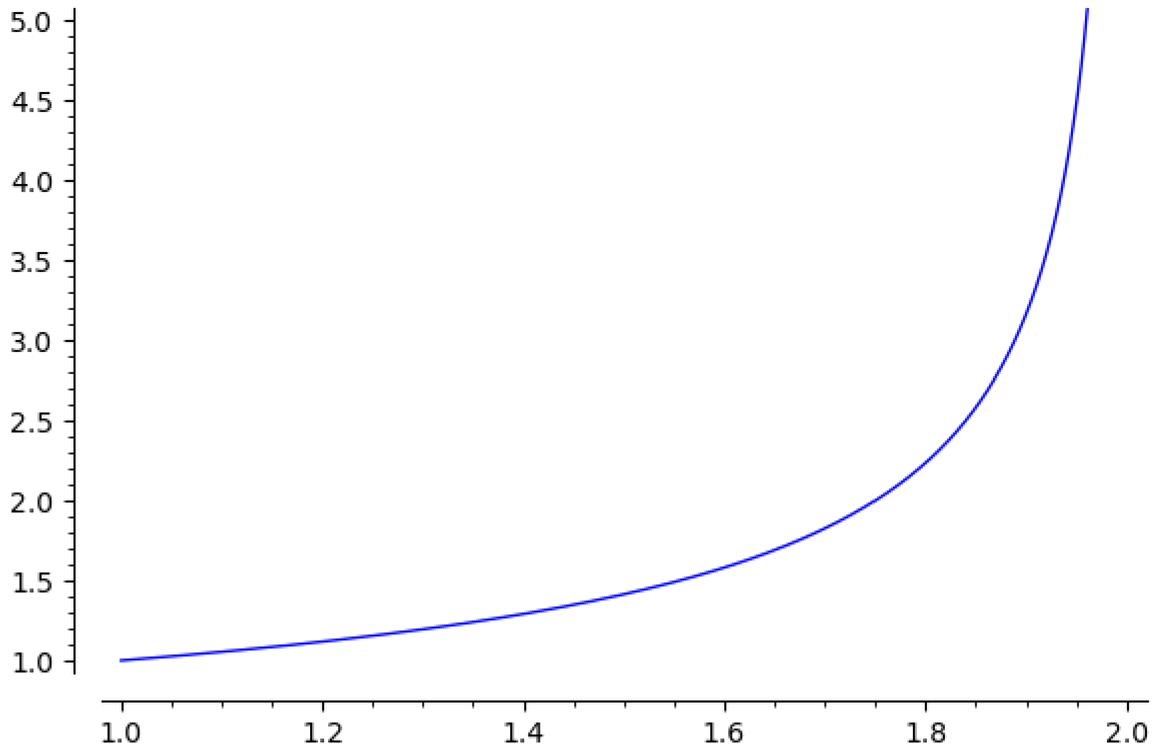
Задание: Вычисление несобственных интегралов численными методами
Тема: 10

Проблемы, возникающие при попытке применить классические методы численного интегрирования к несобственным интегралам

Попадание сетки в точку вблизи бесконечности:

In [ ]:
```
func = 1/(sqrt(2-x))
show(func)
show(plot(func,xmin = 1,xmax = 2,ymax = 5))
#интеграл,вычисленный непосредственно
show("Аналитически выисленное значение интеграла: {}".format(integral(func,x
show("Вычисленное немодифицированным методом трапеций значение: {}".format(i
```

$$\frac{1}{\sqrt{-x+2}}$$



Аналитически выисленное значение интеграла: 2.00000000000000

Вычисленное немодифицированным методом трапеций значение: 152.6492437

Пытаемся интегрировать сетодом трапеций:

In [ ]:
```
integrate(func,1,2,0.0001,"trap",trace = True)
```

2

Out[ ]:
```
152.649243796903
```

In [ ]:
```
h = 0.001
```

In [ ]:
```
show("Метод прямоугольников")
show("Результат: {}\n\t Шаг интегрирования: {}\n\t Ошибка (Рунге Ромберг | 0
     format(integrate(func,X0,X1,h,"trap",trace = true),
            h.n(digits = 2),
            abs(runge_rumbert_error(func,X0,X1,h,0.01,"trap")),
            abs(integrate(func,X0,X1,h,"trap") - integral(func,x,X0,X1).n(di
```

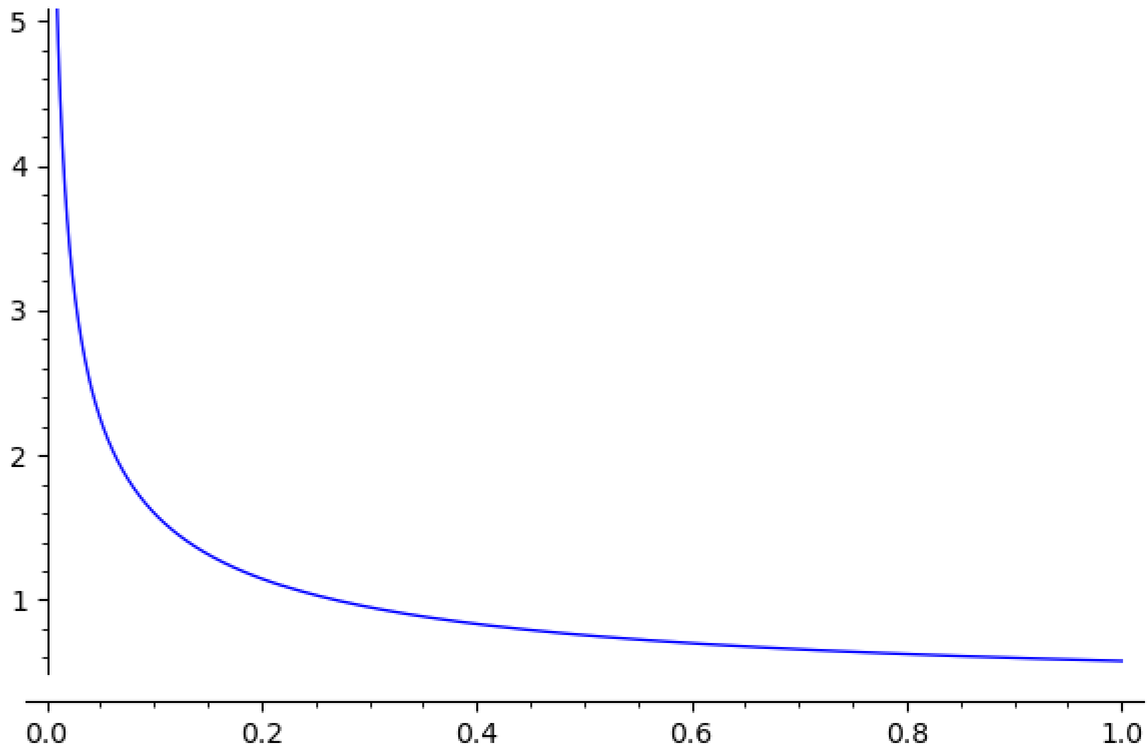
Метод прямоугольников

Результат: 1508.59229291816
Шаг интегрирования: 0.0010
Ошибка (Рунге Ромберг | 0.015): 1521.81634083067
Ошибка (Истинная) 1506.59229291816

Попадание сетки в бесконечность

In [ ]:
```
func = 1/(sqrt(-x**2+4*x))
show(func)
show(plot(func,xmin = 0,xmax = 1,ymax = 5))
#интеграл,вычисленный непосредственно
show(integral(func,x,0,1).n())
```

$$\frac{1}{\sqrt{-x^2 + 4\,x}}$$



1.04719755119660

3

Невозможность численно интегрировать до бесконечности

In [ ]:
```
func = 1/(x**2 + 1)
show(func)
```

```
integral(func,x,1,infinity).n()
```

$$\frac{1}{x^2+1}$$

Out[ ]:
```
0.785398163397448
```

Непонятно, какое число брать в качестве правого предела интегрирования
Если взять слишком малый предел, точность не будет возрастать при сколь угодно большом увеличении мелкости сетки

In [ ]:
```
integrate(func,1,10,0.001,"rect")
```

Out[ ]:
```
0.685729490154680
```

Попытаемся оценить ошибку методом Рунге Ромберга

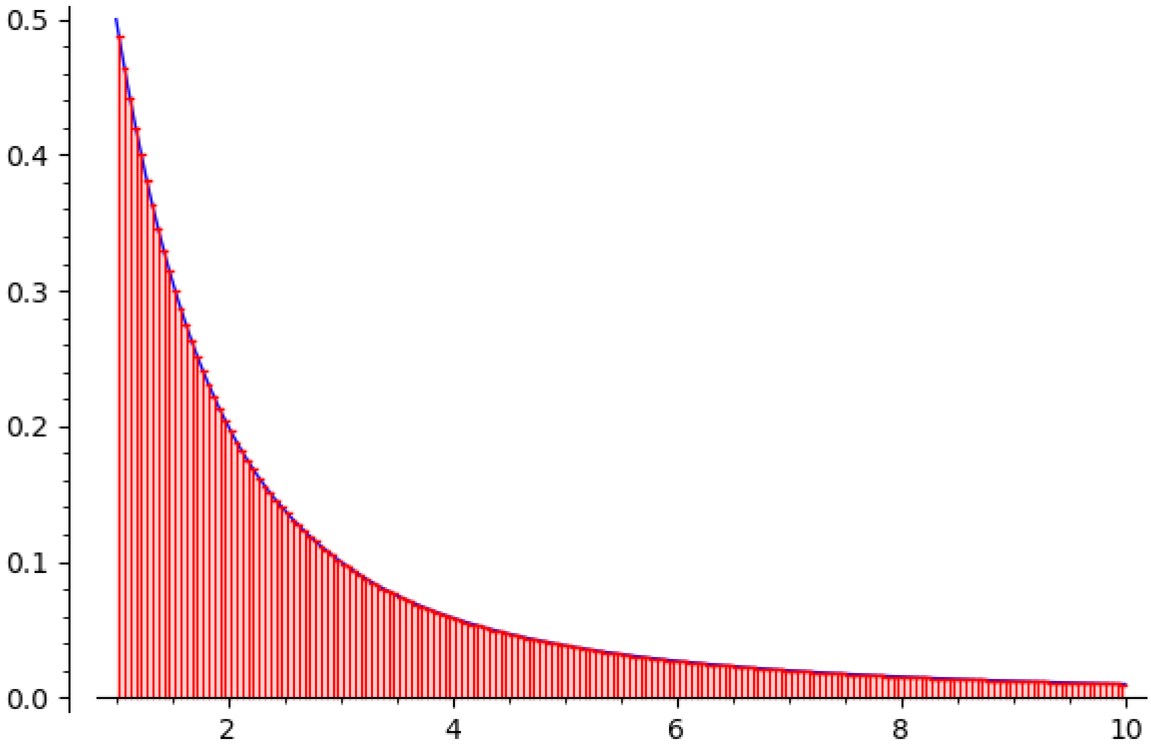
In [ ]:
```
func = 1/(1+x**2)
X0 = 1
X1 = 10
h = 0.05
f = func
show("f(x) =")
show((f(x)))

show("Метод прямоугольников")
show("Результат: {}\n\t Шаг интегрирования: {}\n\t Ошибка (Рунге Ромберг | (
     format(integrate(func,X0,X1,h,"rect",trace = true),
            h.n(digits = 2),
            abs(runge_rumbert_error(func,X0,X1,h,0.015,"rect")),
            abs(integrate(func,X0,X1,h,"rect") - integral(func,x,X0,infinity
```

f(x) =

$$\frac{1}{x^2+1}$$

Метод прямоугольников

4



Результат: 0.685677631807072
Шаг интегрирования: 0.050
Ошибка (Рунге Ромберг | 0.015): 0.0000526572856806114
Ошибка (Истинная) 0.0997205315903765

Видим, что ошибка, определяемая методом Рунге Ромберга сильно меньше истинной

Справедливости ради, стоит отметить, что в большинстве случаев классические квадратурные методы дают неплохую точность и для несобственных интегралов, однако их непредсказуемое поведение в определенных ситуациях не позволяет говорить об их применимости в решении задачи вычисления несобственных интегралов

In [ ]:
```
def runge_romberg_error_h(r1,r2,k,a): #счет ошибки методом Рунге-Ромберга

    err = (r1 - r2)/(k**a - 1)

    return err

def runge_romberg_error(f,xmin,xmax,h,k,method): #счет ошибки методом Рунге-
    r1 = integrate(f,X0,X1,h,method)
    r2 = integrate(f,X0,X1,h*k,method)

    a = 1
    if method == "simpson":
        a = 2

    err = (r1 - r2)/(k**a - 1)   5

    return err

def plot_sq_int(pts):
    if(len(pts) != 3):
```

```
        return false
    a = var("a0,a1,a2")
    fx = a[0]*x**2 + a[1]*x**1 + a[2]
    sols = solve([fx(x = pts[0][0]) == pts[0][1], fx(x = pts[1][0]) == pts[1
    A = []
    for k in sols[0]:
        A.append(k.rhs())
    func = A[0]*x**2 + A[1]*x + A[2]
    return plot(func(x = x),xmin = pts[0][0],xmax = pts[2][0],color = "red")


def integrate(func,xmin,xmax,h,method,trace = false, eps = 1e-5):
    if(h <= 0): #чтоб не упал в бесконечный цикл
        return false

    step = h
    xcur = xmin
    res = 0

    traceplt = 0

    if(h < 0.01):#если шаг очень мал, то трэйсить не надо
        trace = False

    if(trace):
        traceplt = plot(func,xmin = xmin, xmax = xmax)



    if(method == "simpson"):
        while (xcur + 2*step <= xmax + eps):
            if(trace):
                fi1 = func(x = xcur)
                fi2 = func(x = xcur + step)
                fi3 = func(x = xcur + 2*step)

                traceplt += line([(xcur,0),(xcur,fi1)],color = "red")
                traceplt += line([(xcur + step,0),(xcur + step,fi2)],color =
                traceplt += line([(xcur + 2*step,0),(xcur + 2*step,fi3)],col

                traceplt += plot_sq_int([(xcur,fi1),(xcur + step,fi2),(xcur
            res += func(x = xcur) + 4*func(x = xcur + step) + func(x = xcur
            xcur += step*2
        res *= step/3
    else:
        if(method == "trap"):
            while (xcur + step <= xmax + eps):
                if(trace):
                    fi1 = func(x = xcur)
                    fi2 = func(x = xcur + step)
                    traceplt += line([(xcur,0),(xcur,fi1)],color = "red")
                    traceplt += line([(xcur + step,0),(xcur + step,fi2)],col
                    traceplt += line([(xcur, fi1),(xcur + step, fi2)],color

                res += func(x = xcur) + func(x = xcur + step)
                xcur += step
            res *= step/2
        else:
            if(method == "rect")6
                while (xcur + step <= xmax + eps):
                    if(trace):
                        fi = func(x = xcur + step/2)
                        traceplt += line([(xcur + step/2,0),(xcur + step/2,f
                        traceplt += line([(xcur, fi),(xcur + step, fi)],colo
```

```
                    res += func(x = xcur + step/2)
                    xcur += step
                res*=step
            else:
                print("ERROR: wrong method")
                return false
    if(trace):
        show(traceplt)
    return res.n()
```

Несобственные интегралы второго рода

Метод выделения особой точки

Данный метод работает для функций вида:
((x-a)^alpha) * ksi(x)

In [ ]:
```
(x,a,alpha,ksi,f) = var('x,a,alpha,ksi,f')
func = ((x-a)**alpha)*f
show(func)
```

$$(-a+x)^{\alpha} f$$

где a - особая точка
a -1 < alpha < 0

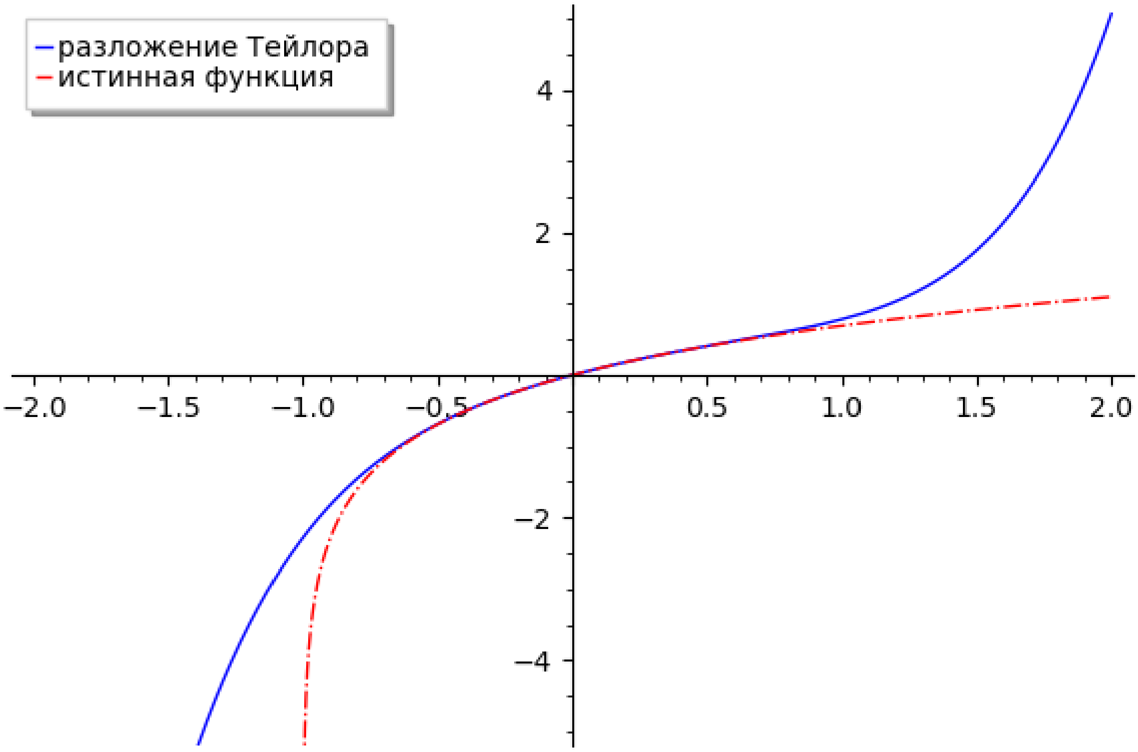
In [ ]:
```
def pairTaylor_series(f,a,n,sep = 0.5):
    df = f.derivative()
    scur = f(x = a)
    ssec = 0
    for i in range(1,n + 1):
        #print(df)
        if( i <= n*sep):
            scur += df(x = a)/(factorial(i)) * (x - a)**(i)
        else:
            ssec += df(x = a)/(factorial(i)) * (x - a)**(i)
        df = df.derivative()
    return (scur,ssec)
```

In [ ]:
```
f = ln(x + 1)
```

In [ ]:
```
Tf,Ts = pairTaylor_series(f,0,5)
Tf += Ts
Tf
```

Out[ ]:
```
1/5*x^5 - 1/4*x^4 + 1/3*x^3 - 1/2*x^2 + x
```

In [ ]:
```
show(plot(Tf,xmin = -2,xmax = 2,legend_label = "разложение Тейлора") +
     plot(f,xmin = -1.0001,xmax = 2, color = "red",legend_label = "истинная
```



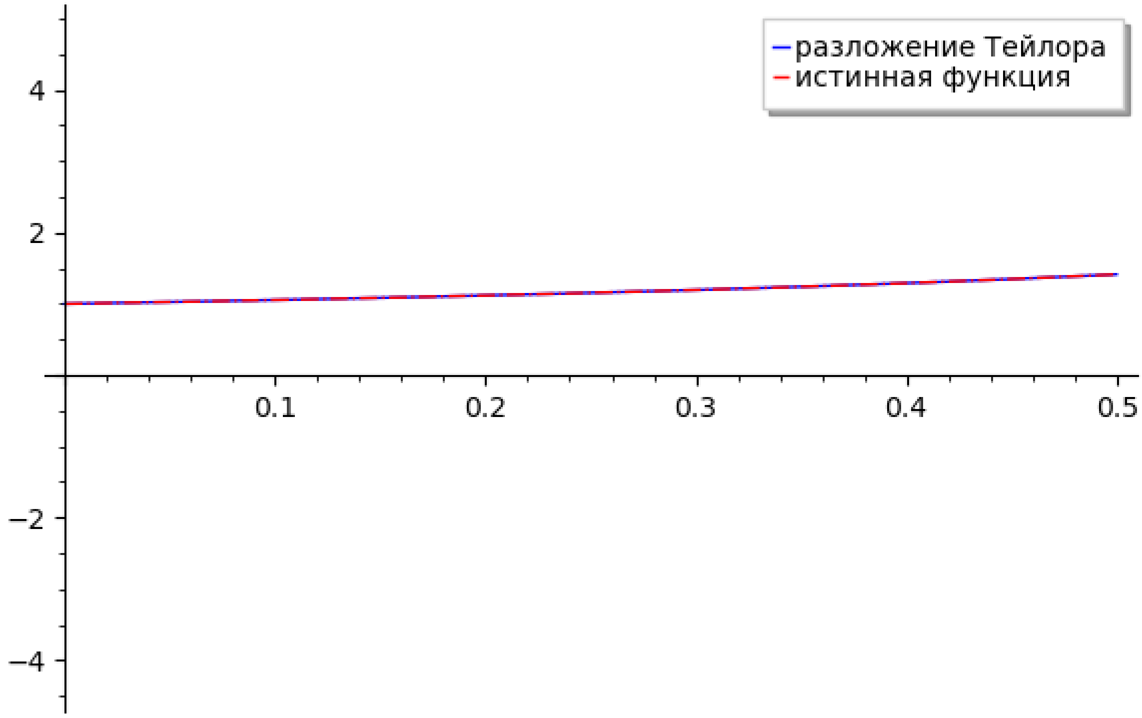
In [ ]:
```
a = 0
al = (-1/2)
csi =  (1-x)**(-1/2)

X0 = 0
X1 = 0.5

f = ((x - a)**al)*csi(x)

show(f)
```

$$\frac{1}{\sqrt{x}\sqrt{-x+1}}$$

In [ ]:
```
Tf,Ts = pairTaylor_series(csi,0,10)
T = Tf + Ts
show(T)

show(plot(T,xmin = X0,xmax = X1,legend_label = "разложение Тейлора") +
     plot(csi,xmin = X0,xmax = X1, color = "red",legend_label = "истинная фу
```

$$\frac{46189}{262144}x^{10}+\frac{12155}{65536}x^{9}+\frac{6435}{32768}x^{8}+\frac{429}{2048}x^{7}+\frac{231}{1024}x^{6}+\frac{63}{256}x^{5}+\frac{35}{128}x^{4}+\frac{5}{16}$$

8



In [ ]:
```
g(x) = ((x - a)**al)*Tf
phi(x) = ((x - a)**al)*Ts

g(x) = g(x).expand().simplify()
phi(x) = phi(x).expand().simplify()

show("g(x) = ")
show(g(x).expand().simplify())
show("phi(x) = ")
show(phi(x).expand().simplify())
```

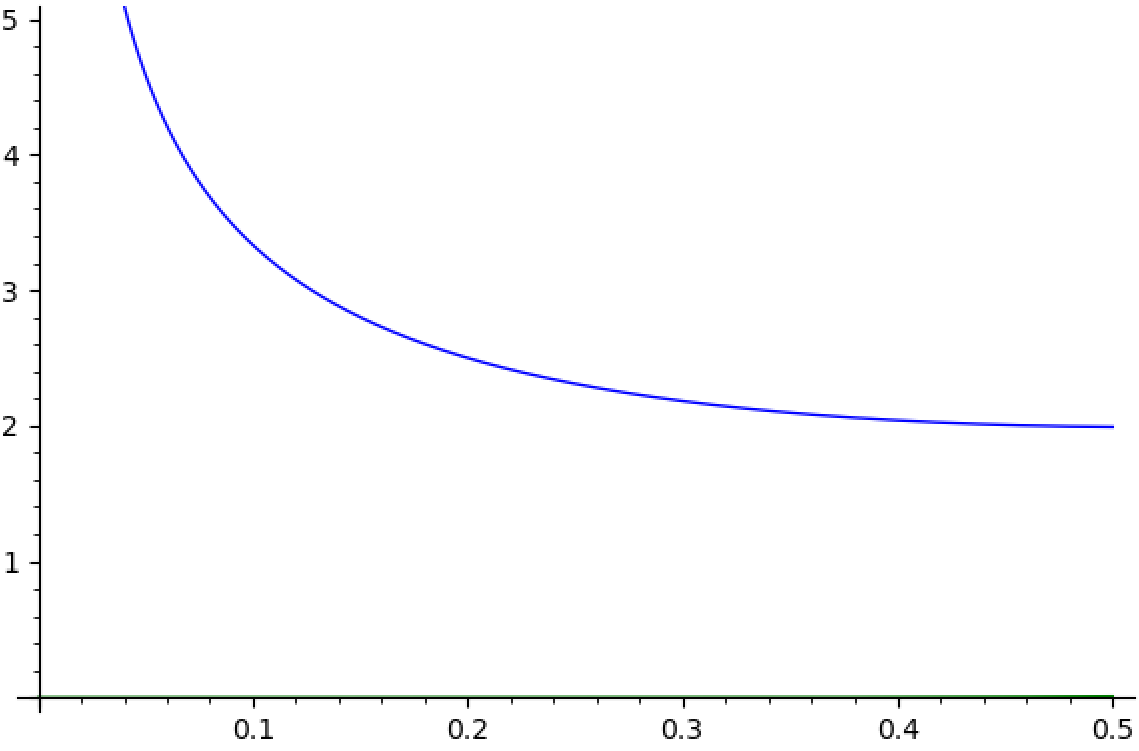
g(x) =

$$\frac{63}{256}x^{\frac{9}{2}} + \frac{35}{128}x^{\frac{7}{2}} + \frac{5}{16}x^{\frac{5}{2}} + \frac{3}{8}x^{\frac{3}{2}} + \frac{1}{2}\sqrt{x} + \frac{1}{\sqrt{x}}$$

phi(x) =

$$\frac{46189}{262144}x^{\frac{19}{2}} + \frac{12155}{65536}x^{\frac{17}{2}} + \frac{6435}{32768}x^{\frac{15}{2}} + \frac{429}{2048}x^{\frac{13}{2}} + \frac{231}{1024}x^{\frac{11}{2}}$$

In [ ]:
```
show(plot(g(x),xmin = X0,xmax = X1) + plot(phi(x),xmin = X0,xmax = X1,color
```



In [ ]:
```
result = integral(g(x),x,X0,X1) + integrate(phi(x),X0,X1,0.01,"rect")
show(result)
```

$$1.5707867680588379$$

In [ ]:
```
 integrate(phi(x),X0,X1,0.001,"simpson")
```

Out[ ]: 0.000640102529309632

In [ ]:
```
integral(g(x),x,X0,X1)
```

Out[ ]: 1.5701471425751357

In [ ]:
```
def pairTaylor_series(f,a,n,sep = 0.5):
    df = f.derivative()
    scur = f(x = a)
    ssec = 0
    for i in range(1,n + 1):
        if( i < n*sep):
            scur += df(x = a)/(factorial(i)) * (x - a)**(i)
        else:
            ssec += df(x = a)/(factorial(i)) * (x - a)**(i)
        df = df.derivative()
    return (scur,ssec)

def Improper_integral_SP_isolation(h = 0.01, #решение несобственного интегра
                                   trace = false, #применимо лишь для функци
                                   sep = 0.5,
                                   method = "rect"):
    Tf,Ts = pairTaylor_series(csi,0,10,sep = sep)
    T = Tf + Ts
    if(trace):
        show(T)

        show(plot(T,xmin = X0,xmax = X1,legend_label = "разложение Тейлора")
             plot(csi,xmin = X0,xmax = X1, color = "red",legend_label = "ист
             ymin = -5,ymax = 5)
```

```
    g(x) = ((x - a)**al)*Tf
    phi(x) = ((x - a)**al)*Ts

    g(x) = g(x).expand().simplify()
    phi(x) = phi(x).expand().simplify()

    if(trace):
        show("g(x) = ")
        show(g(x).expand().simplify())
        show("phi(x) = ")
        show(phi(x).expand().simplify())
        show(plot(g(x),xmin = X0,xmax = X1,legend_label = "g(x)") +
             plot(phi(x),xmin = X0,xmax = X1,color = "green",legend_label =
             ymin = 0,ymax =5)

    result = integral(g(x),x,X0,X1) + integrate(phi(x),X0,X1,h,method)
    return result
```

In [ ]:
```
a = 0
al = (-1/2)
csi =  (1-x)**(-1/2)

X0 = 0
X1 = 0.5

f = ((x - a)**al)*csi(x)

show("Необходимо рассчитать интеграл этой функции:")
show(f)
show("От {} до {}".format(X0.n(digits = 5),X1.n(digits = 5)))

show("Ее график:")
show(plot(f,xmin = X0, xmax = X1,ymin = 0,ymax = 5))
```

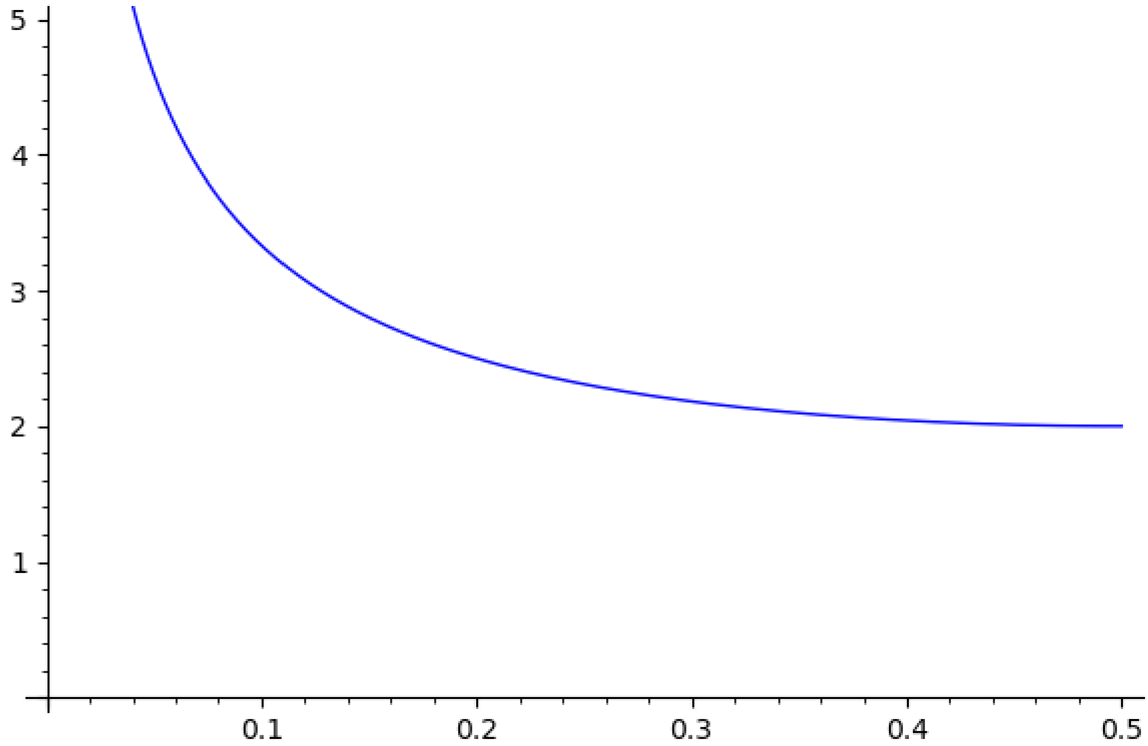
Необходимо рассчитать интеграл этой функции:

$$\frac{1}{\sqrt{x}\sqrt{-x+1}}$$

От 0.00000 до 0.50000

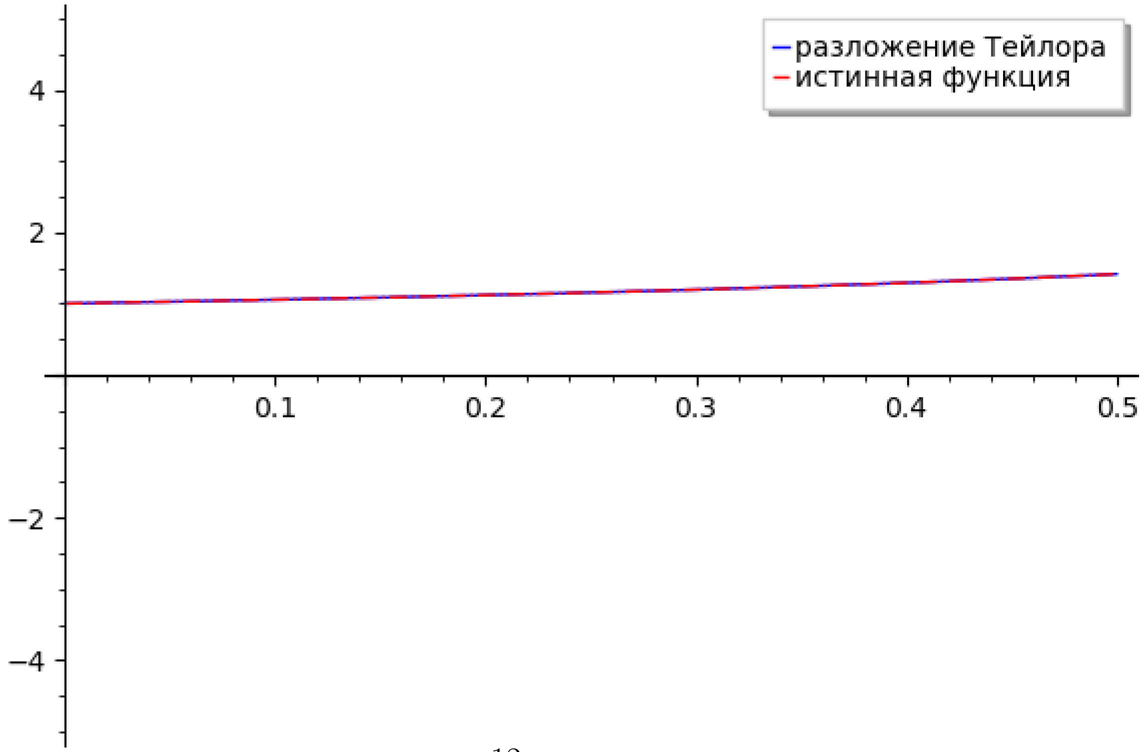
Ее график:

11



```
In [ ]: print("\n\n")
res = Improper_integral_SP_isolation(h = 0.1,sep = 0.1,trace = true)
show(res)
```

$$\frac{46189}{262144}x^{10}+\frac{12155}{65536}x^{9}+\frac{6435}{32768}x^{8}+\frac{429}{2048}x^{7}+\frac{231}{1024}x^{6}+\frac{63}{256}x^{5}+\frac{35}{128}x^{4}+\frac{5}{16}$$



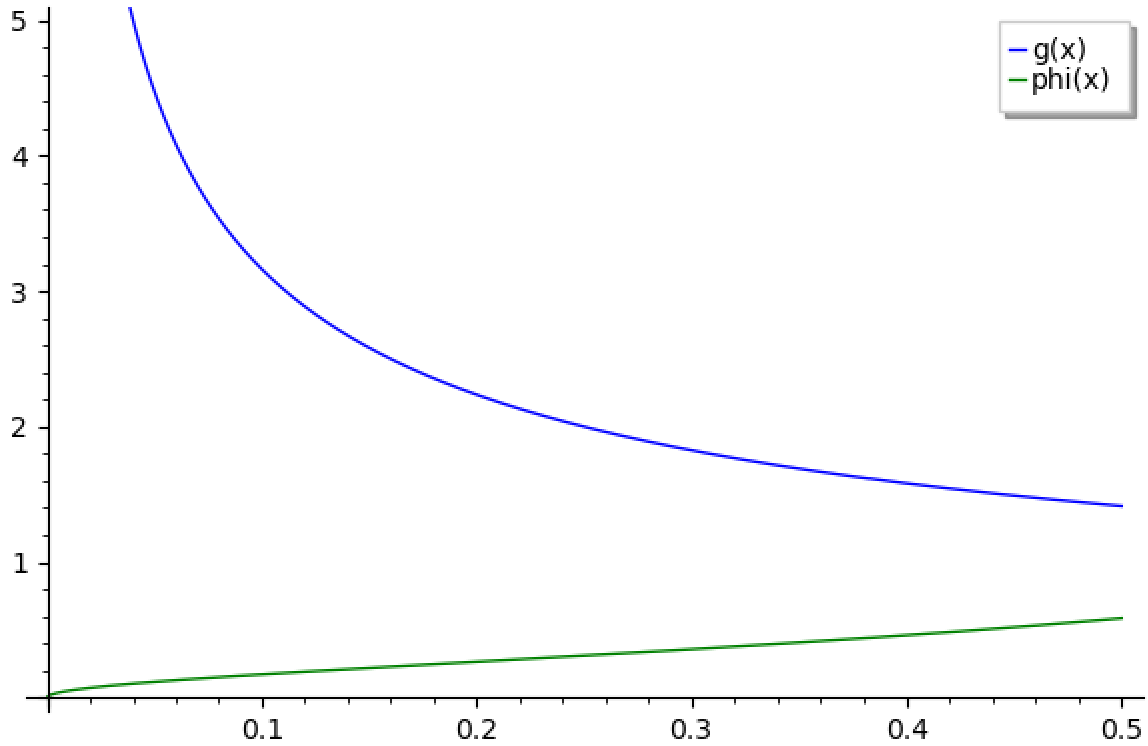
12

g(x) =

$$\frac{1}{\sqrt{x}}$$

phi(x) =

$$\frac{46189}{262144}x^{\frac{19}{2}}+\frac{12155}{65536}x^{\frac{17}{2}}+\frac{6435}{32768}x^{\frac{15}{2}}+\frac{429}{2048}x^{\frac{13}{2}}+\frac{231}{1024}x^{\frac{11}{2}}+\frac{63}{256}x^{\frac{9}{2}}+\frac{35}{128}x^{\frac{7}{2}}$$



In [ ]:
```
h = 0.1
```

In [ ]:
```
show("Метод выделения особенности")
show("Результат: {}\n\t Шаг интегрирования: {}\n\t Ошибка (Рунге Ромберг | (
     format(Improper_integral_SP_isolation(h,trace = false),
            h.n(digits = 2),
            abs(runge_romberg_error_h(Improper_integral_SP_isolation(h,sep =
            abs(Improper_integral_SP_isolation(h,sep = 0.1) - integral(f,x,X
```

Метод выделения особенности

Результат: 1.5707005541572556

Шаг интегрирования: 0.10

Ошибка (Рунге Ромберг | 0.01): 0.00039980858781429873

Ошибка (Истинная) 0.00038763302527100585

13

In [ ]:
```
show("Метод прямоугольников")
show("Результат: {}\n\t Шаг интегрирования: {}\n\t Ошибка (Рунге Ромберг | (
     format(integrate(f,X0,X1,h,"rect",trace = false),
            h.n(digits = 2),
```

```
            abs(runge_rumbert_error(f,X0,X1,h,0.01,"rect")),
            abs(integrate(f,X0,X1,h,"rect") - integral(f,x,X0,X1).n(digits =
```

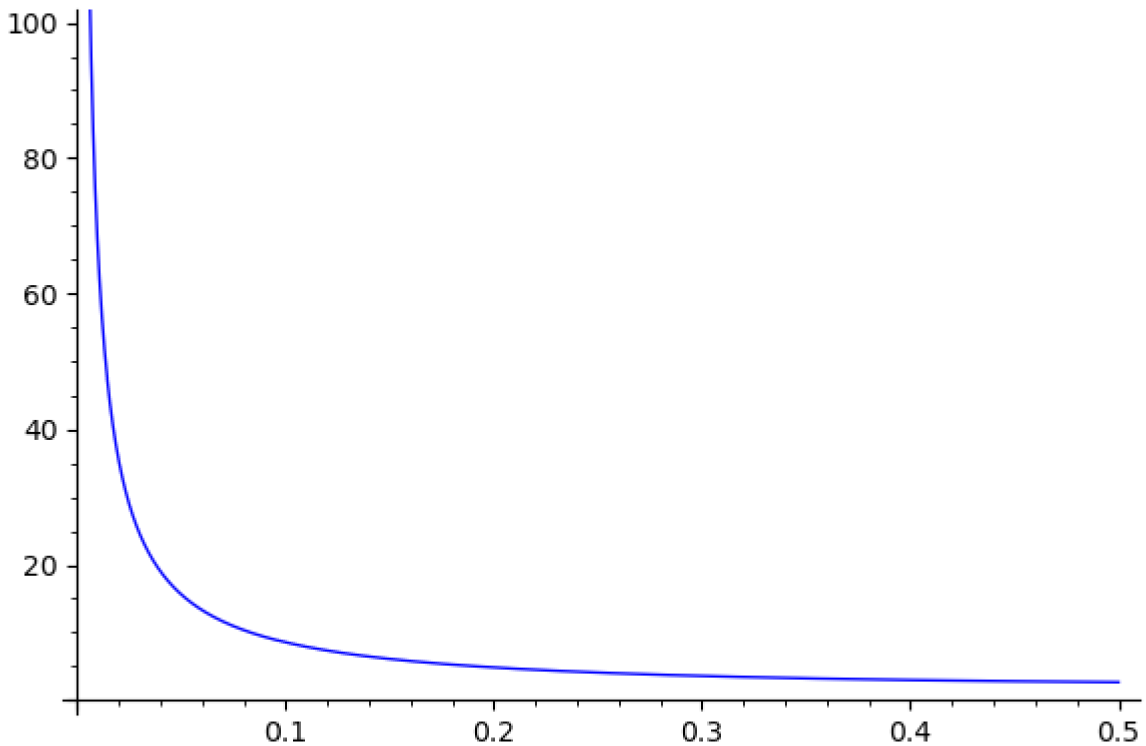
In [ ]:
```
a = 0
al = (-1/1.1)
csi =  (1-x)**(-1/2)

X0 = 0
X1 = 0.5

f = ((x - a)**al)*csi(x)

show("Необходимо рассчитать интеграл этой функции:")
show(f)
show("От {} до {}".format(X0.n(digits = 5),X1.n(digits = 5)))

show("Ее график:")
show(plot(f,xmin = X0, xmax = X1,ymin = 0,ymax = 100))
```

Необходимо рассчитать интеграл этой функции:

$$\frac{1}{x^{0.909090909090909}\sqrt{-x+1}}$$

От 0.00000 до 0.50000

Ее график:



Основной минус этого метода заключается в том, что он требует непосредственного вычисления интеграла, и оттого не может считаться на 100% численным.

Метод исключения особой точки

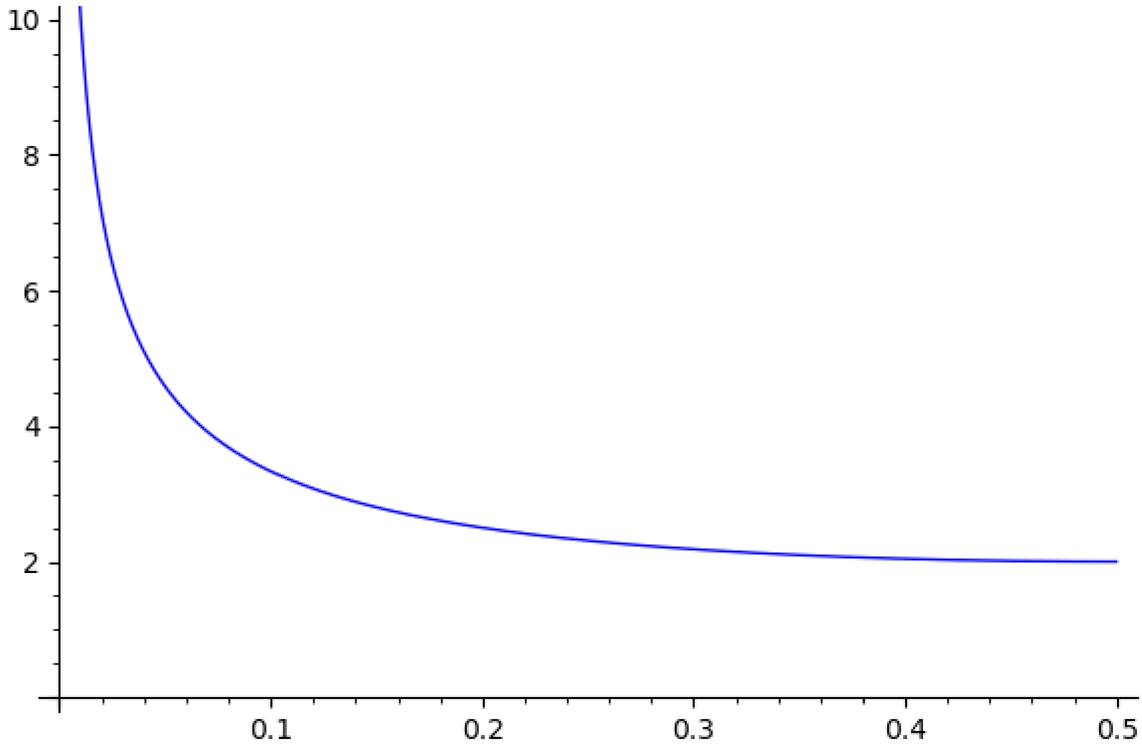
С пердварительной обработкой

In [ ]:
```
def Improper_integral_SP_exclusion(#функция обнаруживает точечных особенност
                                   f,X0,X1,h,
                                   trace = false,
                                   method = "rect",
                                   eps = 1e-4):
                                   #без сдвига сетки
    hc = h #с шагом h будем вычислять производную
    h /= 2 #проверять надо точки, из которых будем брать занчения f
    infpos = 1/h
    infneg = -1/h
    xcur = X0
    if(trace):
            print("{} -- {} | checking...".format(X0,X1))

    i = 0
    repmark = (X1-X0)/(hc*10)
    result = 0
    while (xcur < X1 + h*eps):

        try:
            f(x = xcur)
            if(trace and (( i % (repmark) == 0) or abs(f(x = xcur)) > infpos
                print("{} -- {} ({}) -- {} \t|{}|".format(X0,xcur,f(x = xcur
        except(BaseException): #проверка на деление на 0
            if(trace):
                print(" |0|")
                print("  !{} -- {} (NAN) -- {} \t|{}|".format(X0,xcur,X1,h))
            if(xcur - eps < X0):
                #это начальная точка
                #сдвигаем начало на шаг
                X0 += hc
                xcur += h
                #проверяем следующую точку
                continue
            else:
                if(xcur + eps > X1):
                    #это последняя точка
                    #после инетгрировать не нужно
                    if(trace):
                        print("integrating ps...")
                    return integrate(f,X0,X1 - hc, h = hc,method = method,ep
                else:
                    #это точка в середине
                    if(trace):
                        print("integrating fu ps...")
                    #интегрироуем до последнего проверенного узла
                    result += integrate(f,X0,xcur - hc, h = hc,method = meth
                    #продолжаем проверку со следующего узла
                    X0 = xcur + hc
                    continue

        if (f(x = xcur) > infpos or f(x = xcur) < infneg): #проверка на свер
            if(trace):
                print(" |inf|")
                print("  !{} -- {} ({}) -- {} \t|{}|".format(X0,xcur,f(x = x
            if(xcur - eps < X0):
                #это начальная точка
                #сдвигаем начало на шаг
                X0 += hc
                xcur += h
```

```
                #проверяем следующую точку
                continue
            else:
                if(xcur + eps > X1):
                    #это последняя точка
                    #после инетгрировать не нужно
                    if(trace):
                        print("integrating ps...")
                    return integrate(f,X0,X1 - hc, h = hc,method = method,ep
                else:
                    #это точка в середине
                    if(trace):
                        print("integrating fu ps...")
                    #интегрироуем до последнего проверенного узла
                    result += integrate(f,X0,xcur - hc, h = hc,method = meth
                    #продолжаем проверку со следующего узла
                    X0 = xcur + hc
                    continue

        if (complex(f(x = xcur)).imag != 0):
            if(trace):
                print(" |complex|")
                print("  !{} -- {} ({}) -- {} \t|{}|".format(X0,xcur,f(x = x
            if(xcur - eps < X0):
                #это начальная точка
                #сдвигаем начало на шаг
                X0 += hc
                xcur += h
                #проверяем следующую точку
                continue
            else:
                if(xcur + eps > X1):
                    #это последняя точка
                    #после инетгрировать не нужно
                    if(trace):
                        print("integrating ps...")
                    return integrate(f,X0,X1 - hc, h = hc,method = method,ep
                else:
                    #это точка в середине
                    if(trace):
                        print("integrating fu ps...")
                    #интегрироуем до последнего проверенного узла
                    result += integrate(f,X0,xcur - hc, h = hc,method = meth
                    #продолжаем проверку со следующего узла
                    X0 = xcur + hc
                    continue

        xcur += h
        i += 1

    if(trace):
            print("integrating...")

    return result + integrate(f,X0,X1,hc,method = method)
```

In [ ]:

```
show(plot(f,xmin = X0, xmax = X1,ymin = 0,ymax = 10))
```



In [ ]:
```
integral(f,x,X0,X1)
```

Out[ ]:
```
pi - 1.5707963267948966
```

In [ ]:
```
Improper_integral_SP_exclusion(f,X0,X1,h = 0.00001,trace = true,method = "tr
```

```
0 -- 0.500000000000000 | checking...
 |0|
  !0 -- 0 (NAN) -- 0.500000000000000    |5.00000000000000e-6|
0.0000100000000000000 -- 5.00000000000000e-6 (447.214713538139) -- 0.500000
000000000        |5.00000000000000e-6|
0.0000100000000000000 -- 0.0250050000000023 (6.40450215747031) -- 0.5000000
00000000         |5.00000000000000e-6|
0.0000100000000000000 -- 0.0500049999999970 (4.58809735283853) -- 0.5000000
00000000         |5.00000000000000e-6|
0.0000100000000000000 -- 0.0750050000000046 (3.79651569580492) -- 0.5000000
00000000         |5.00000000000000e-6|
0.0000100000000000000 -- 0.100005000000030 (3.33325926219080) -- 0.50000000
0000000          |5.00000000000000e-6|
0.0000100000000000000 -- 0.125005000000055 (3.02366395062394) -- 0.50000000
0000000          |5.00000000000000e-6|
0.0000100000000000000 -- 0.150005000000080 (2.80052173006020) -- 0.50000000
0000000          |5.00000000000000e-6|
0.0000100000000000000 -- 0.175005000000105 (2.63177715849876) -- 0.50000000
0000000          |5.00000000000000e-6|
0.0000100000000000000 -- 0.200005000000130 (2.49997656302428) -- 0.50000000
0000000          |5.00000000000000e-6|
0.0000100000000000000 -- 0.225005000000155 (2.39471847751776) -- 0.50000000
0000000          |5.00000000000000e-6|
0.0000100000000000000 -- 0.250005000000180 (2.30938568105869) -- 0.50000000
0000000          |5.00000000000000e-6|
0.0000100000000000000 -- 0.275005000000066 (2.23955740699827) -- 0.50000000
0000000          |5.00000000000000e-6|
0.0000100000000000000 -- 0.300004999999952 (2.18216851123603) -- 0.50000000
0000000          |5.00000000000000e-6|
0.0000100000000000000 -- 0.325004999999838 (2.13503353507013) -- 0.50000000
0000000          |5.00000000000000e-6|
0.0000100000000000000 -- 0.350004999999725 (2.09656276182544) -- 0.50000000
0000000          |5.00000000000000e-6|
0.0000100000000000000 -- 0.375004999999611 (2.06558560986693) -- 0.50000000
0000000          |5.00000000000000e-6|
0.0000100000000000000 -- 0.400004999999497 (2.04123719985299) -- 0.50000000
0000000          |5.00000000000000e-6|
0.0000100000000000000 -- 0.425004999999383 (2.02288384563335) -- 0.50000000
0000000          |5.00000000000000e-6|
0.0000100000000000000 -- 0.450004999999269 (2.01007360024389) -- 0.50000000
0000000          |5.00000000000000e-6|
0.0000100000000000000 -- 0.475004999999156 (2.00250369362658) -- 0.50000000
0000000          |5.00000000000000e-6|
integrating...
```

Out[ ]:

```
1.56459713033287
```

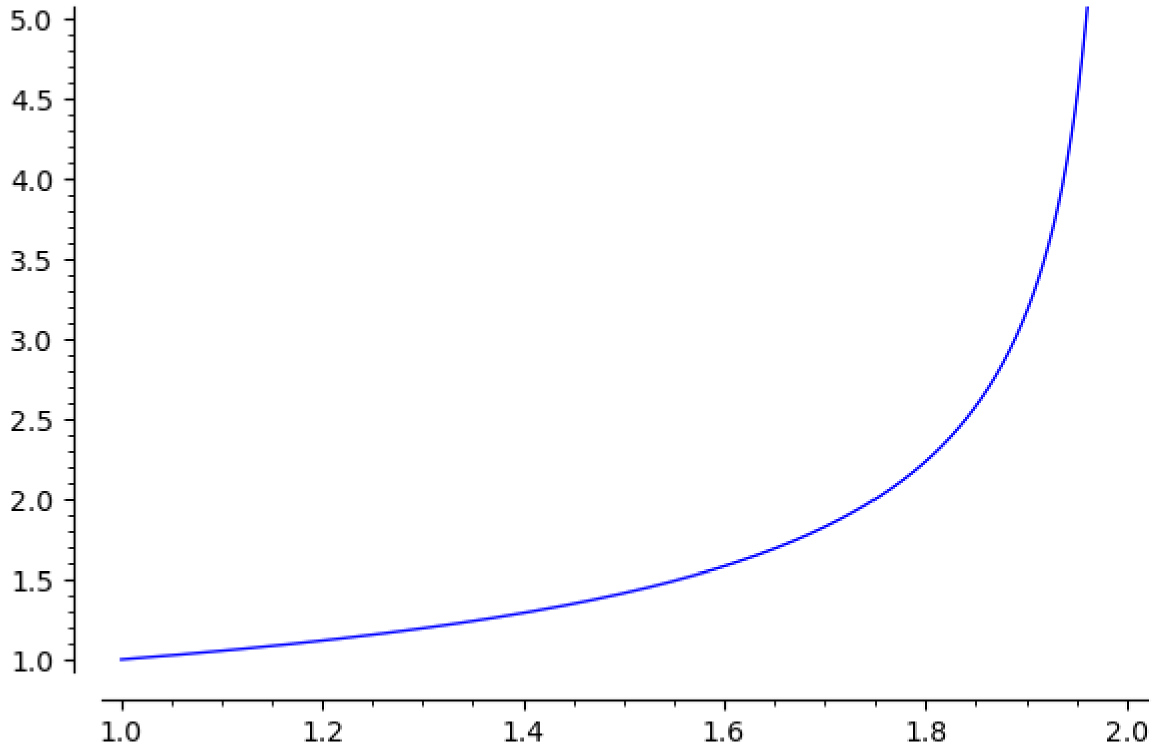
Этот метод дает довольно большую ошибку, но при этом исключает возможность возникновения экстримальных значений и ошибок деления на ноль

Попробуем решить ту задачу, что дала неадекватный результат методом трапеций

In [ ]:

```
func = 1/(sqrt(2-x))
X0 = 1
X1 = 2
h = 0.01
show(func)
show(plot(func,xmin = 1,xmax = 2,ymax = 5))
#интеграл,вычисленный непосредственно
show(integral(func,x,1,2).n())
```

$$\frac{1}{\sqrt{-x+2}}$$



2.00000000000000

In [ ]:
```
Improper_integral_SP_exclusion(func,1,2,0.0001,method = "trap",trace = true)
```

```
1 -- 2 | checking...
1 -- 1 (1) -- 2         |0.000050000000000000|
1 -- 1.05000000000011 (1.02597835208521) -- 2   |0.000050000000000000|
1 -- 1.10000000000021 (1.05409255338958) -- 2   |0.000050000000000000|
1 -- 1.15000000000032 (1.08465228909348) -- 2   |0.000050000000000000|
1 -- 1.20000000000042 (1.11803398875019) -- 2   |0.000050000000000000|
1 -- 1.25000000000053 (1.15470053837966) -- 2   |0.000050000000000000|
1 -- 1.30000000000063 (1.19522860933493) -- 2   |0.000050000000000000|
1 -- 1.35000000000074 (1.24034734589279) -- 2   |0.000050000000000000|
1 -- 1.40000000000084 (1.29099444873671) -- 2   |0.000050000000000000|
1 -- 1.45000000000095 (1.34839972492765) -- 2   |0.000050000000000000|
1 -- 1.50000000000106 (1.41421356237459) -- 2   |0.000050000000000000|
1 -- 1.55000000000116 (1.49071198500178) -- 2   |0.000050000000000000|
1 -- 1.60000000000127 (1.58113883008669) -- 2   |0.000050000000000000|
1 -- 1.65000000000137 (1.69030850946035) -- 2   |0.000050000000000000|
1 -- 1.70000000000148 (1.82574185835505) -- 2   |0.000050000000000000|
1 -- 1.75000000000158 (2.00000000000633) -- 2   |0.000050000000000000|
1 -- 1.80000000000169 (2.23606797750923) -- 2   |0.000050000000000000|
1 -- 1.85000000000179 (2.58198889748705) -- 2   |0.000050000000000000|
1 -- 1.90000000000190 (3.16227766019841) -- 2   |0.000050000000000000|
1 -- 1.95000000000200 (4.47213595508924) -- 2   |0.000050000000000000|
1 -- 2.00000000000211 (4.21509657505623e-11 - 688377.510640772*I) -- 2  |0.
000050000000000000|
 |complex|
  !1 -- 2.00000000000211 (4.21509657505623e-11 - 688377.510640772*I) -- 2
|0.000050000000000000|
integrating ps...
```

Out[ ]: 1.98039645448383

In [ ]:
```
h = 0.01
show("Метод исключения особенности")
show("Результат: {}\n\t Шаг интегрирования: {}\n\t Ошибка (Рунге Ромберг | 0
     format(Improper_integral_SP_exclusion(f,X0,X1,h = h,method = "trap"),
            h.n(digits = 2),
            abs(runge_rumbert_error_h(Improper_integral_SP_exclusion(f,X0,X1
            abs(Improper_integral_SP_exclusion(f,X0,X1,h = h,method = "trap"
```

Метод исключения особенности

Результат: 1.37440494673634
Шаг интегрирования: 0.010
Ошибка (Рунге Ромберг | 0.01): 0.177979732200823
Ошибка (Истинная) 0.196391380058554

Модифицированный метод исключения особенности

In [ ]:
```
def Improper_integral_SP_exclusion_mod(#функция обнаруживает точечнык особен
                                   f,X0,X1,h,
                                   trace = false,
                                   method = "rect",
                                   eps = 1e-4):
                                   #без сдвига сетки
    hc = h #с шагом h будем вычислять производную
    h /= 2 #проверять надо точки, из которых будем брать занчения f
    hsub = hc*eps #шаг для вспомогательного интегрирования
    infpos = 1/(hc*eps)
    infneg = -1/(hc*eps)
    xcur = X0
    repmark = (X1-X0)/(hc*10)
    if(trace):
            print("{} -- {} | checking...".format(X0,X1))

    i = 0
    while (xcur < X1 + h*eps):

        try:
            f(x = xcur)
            if(trace and i % (repmark) == 0):
                print("{} -- {} ({}) -- {} \t|{}|".format(X0,xcur,f(x = xcur
        except(BaseException): #проверка на деление на 0
            if(trace):
                print(" |0|")
                print("  !{} -- {} (NAN) -- {} \t|{}|".format(X0,xcur,X1,h))
            if(xcur - h*eps < X0):
                return (Improper_integral_SP_exclusion(f,X0 + hc,X1, h = hc,
                        Improper_integral_SP_exclusion_mod(f,X0 + hsub,xcur
            if(xcur + h*eps > X1):
                return (integrate(f,X0,X1 - hc, h = hc,method = method,eps =
                        Improper_integral_SP_exclusion_mod(f,X1 - hc,xcur -

            return (integrate(f,X0,xcur - hc, h = hc,method = method,eps = h
                    Improper_integral_SP_exclusion(f,xcur + hc,X1, h = hc,me
                        Improper_integral_SP_exclusion_mod(f,xcur - hc,xcur
                        Improper_integral_SP_exclusion_mod(f,xcur + hsub,xcu

        if (f(x = xcur) > infpos or f(x = xcur) < infneg): #проверка на свер
            if(trace):
                print(" |inf|")
```

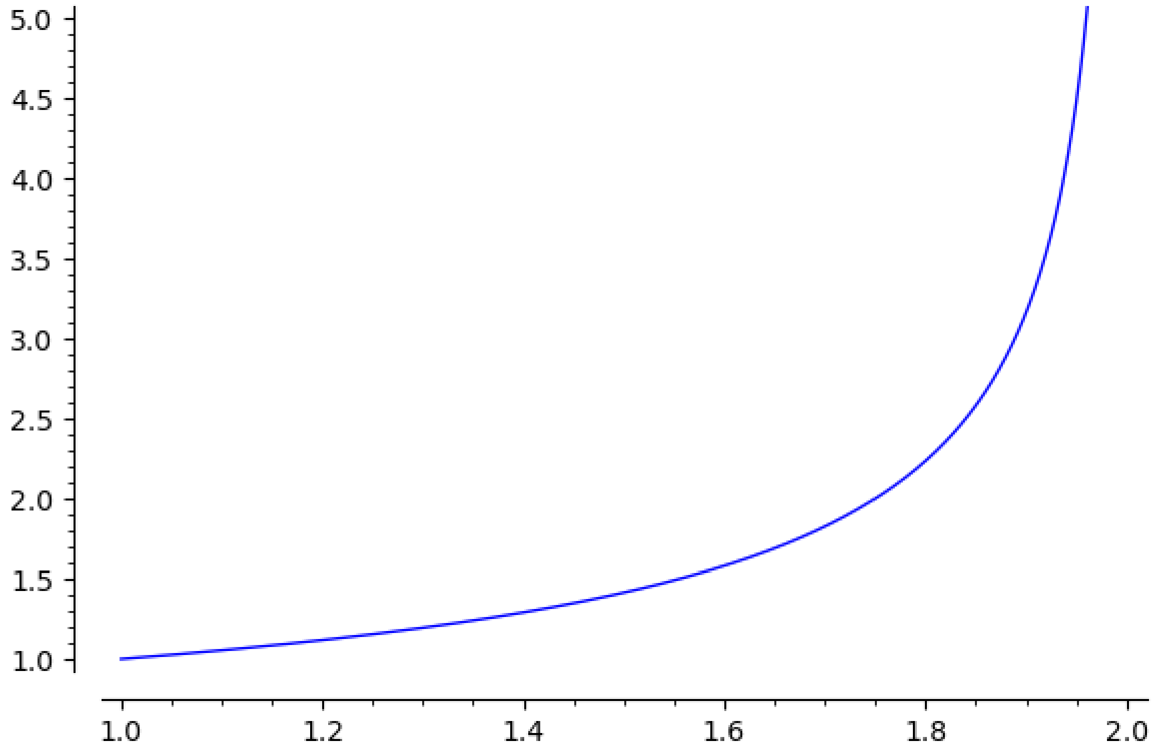
20

```
                print("  !{} -- {} ({}) -- {} \t|{}|".format(X0,xcur,f(x = x
            if(xcur - h*eps < X0):
                return (Improper_integral_SP_exclusion(f,X0 + hc,X1, h = hc,
                        Improper_integral_SP_exclusion_mod(f,X0 + hsub,xcur
            if(xcur + h*eps > X1):
                return (integrate(f,X0,X1 - hc, h = hc,method = method,eps =
                        Improper_integral_SP_exclusion_mod(f,X1 - hc,xcur -

            return (integrate(f,X0,xcur - hc, h = hc,method = method,eps = h
                    Improper_integral_SP_exclusion(f,xcur + hc,X1, h = hc,met
                        Improper_integral_SP_exclusion_mod(f,xcur + hsub,xcu
                        Improper_integral_SP_exclusion_mod(f,xcur - hc,xcur
        if (complex(f(x = xcur)).imag != 0):
            if(trace):
                print(" |complex|")
                print("  !{} -- {} ({}) -- {} \t|{}|".format(X0,xcur,f(x = x

            if(xcur - h*eps < X0):
                return (Improper_integral_SP_exclusion(f,X0 + hc,X1, h = hc,
                        Improper_integral_SP_exclusion_mod(f,X0 + hsub,xcur
            if(xcur + h*eps > X1):
                return (integrate(f,X0,X1 - hc, h = hc,method = method,eps =
                        Improper_integral_SP_exclusion_mod(f,X1 - hc,xcur -

            return (integrate(f,X0,xcur - hc, h = hc,method = method,eps = h
                    Improper_integral_SP_exclusion(f,xcur + hc,X1, h = hc,met
                        Improper_integral_SP_exclusion_mod(f,xcur + hsub,xcu
                        Improper_integral_SP_exclusion_mod(f,xcur - hc,xcur

        xcur += h
        i += 1

    if(trace):
            print("integrating...")

    return integrate(f,X0,X1,hc,method = method,eps = eps*hc)
```

In [ ]:

```
func = 1/(sqrt(2-x))
X0 = 1
X1 = 2
h = 0.1
show(func)
show(plot(func,xmin = 1,xmax = 2,ymax = 5))
#интеграл,вычисленный непосредственно
show(integral(func,x,1,2).n())
```

$$\frac{1}{\sqrt{-x+2}}$$



2.00000000000000

In [ ]:
```
show("Метод исключения особенности с дополнительным интегрированием")
show("Результат: {}\n\t Шаг интегрирования: {}\n\t Ошибка (Рунге Ромберг | 0
     format(Improper_integral_SP_exclusion_mod(func,X0,X1,h = h,method = "re
            h.n(digits = 2),
            abs(runge_rumbert_error_h(Improper_integral_SP_exclusion_mod(fur
            abs(Improper_integral_SP_exclusion_mod(func,X0,X1,h = h,method =
```

Метод исключения особенности с дополнительным интегрированием

Результат: 1.98777930310592
Шаг интегрирования: 0.10
Ошибка (Рунге Ромберг | 0.01): 0.00697763328108115
Ошибка (Истинная) 0.0122206968940812

In [ ]:
```
Improper_integral_SP_exclusion_mod(func,1,2,0.0001,method = "trap",trace = t
```

```
1 -- 2 | checking...
1 -- 1 (1) -- 2         |0.0000500000000000000|
1 -- 1.05000000000011 (1.02597835208521) -- 2   |0.0000500000000000000|
1 -- 1.10000000000021 (1.05409255338958) -- 2   |0.0000500000000000000|
1 -- 1.15000000000032 (1.08465228909348) -- 2   |0.0000500000000000000|
1 -- 1.20000000000042 (1.11803398875019) -- 2   |0.0000500000000000000|
1 -- 1.25000000000053 (1.15470053837966) -- 2   |0.0000500000000000000|
1 -- 1.30000000000063 (1.19522860933493) -- 2   |0.0000500000000000000|
1 -- 1.35000000000074 (1.24034734589279) -- 2   |0.0000500000000000000|
1 -- 1.40000000000084 (1.29099444873671) -- 2   |0.0000500000000000000|
1 -- 1.45000000000095 (1.34839972492765) -- 2   |0.0000500000000000000|
1 -- 1.50000000000106 (1.41421356237459) -- 2   |0.0000500000000000000|
1 -- 1.55000000000116 (1.49071198500178) -- 2   |0.0000500000000000000|
1 -- 1.60000000000127 (1.58113883008669) -- 2   |0.0000500000000000000|
1 -- 1.65000000000137 (1.69030850946035) -- 2   |0.0000500000000000000|
1 -- 1.70000000000148 (1.82574185835505) -- 2   |0.0000500000000000000|
1 -- 1.75000000000158 (2.00000000000633) -- 2   |0.0000500000000000000|
1 -- 1.80000000000169 (2.23606797750923) -- 2   |0.0000500000000000000|
1 -- 1.85000000000179 (2.58198889748705) -- 2   |0.0000500000000000000|
1 -- 1.90000000000190 (3.16227766019841) -- 2   |0.0000500000000000000|
1 -- 1.95000000000200 (4.47213595508924) -- 2   |0.0000500000000000000|
1 -- 2.00000000000211 (4.21509657505623e-11 - 688377.510640772*I) -- 2  |0.
0000500000000000000|
 |complex|
  !1 -- 2.00000000000211 (4.21509657505623e-11 - 688377.510640772*I) -- 2
|0.0000500000000000000|
1.99990000000000 -- 1.99999999000211 | checking...
1.99990000000000 -- 1.99990000000000 (100.000000000006) -- 1.99999999000211
|5.00000000000000e-9|
1.99990000000000 -- 1.99990499999997 (102.597835192113) -- 1.99999999000211
|5.00000000000000e-9|
1.99990000000000 -- 1.99990999999994 (105.409255303362) -- 1.99999999000211
|5.00000000000000e-9|
1.99990000000000 -- 1.99991499999991 (108.465228851171) -- 1.99999999000211
|5.00000000000000e-9|
1.99990000000000 -- 1.99991999999988 (111.803398790062) -- 1.99999999000211
|5.00000000000000e-9|
1.99990000000000 -- 1.99992499999985 (115.470053720973) -- 1.99999999000211
|5.00000000000000e-9|
1.99990000000000 -- 1.99992999999982 (119.522860777792) -- 1.99999999000211
|5.00000000000000e-9|
1.99990000000000 -- 1.99993499999979 (124.034734386268) -- 1.99999999000211
|5.00000000000000e-9|
1.99990000000000 -- 1.99993999999976 (129.099444612060) -- 1.99999999000211
|5.00000000000000e-9|
1.99990000000000 -- 1.99994499999973 (134.839972157418) -- 1.99999999000211
|5.00000000000000e-9|
1.99990000000000 -- 1.99994999999970 (141.421355807583) -- 1.99999999000211
|5.00000000000000e-9|
1.99990000000000 -- 1.99995499999967 (149.071197946352) -- 1.99999999000211
|5.00000000000000e-9|
1.99990000000000 -- 1.99995999999964 (158.113882287741) -- 1.99999999000211
|5.00000000000000e-9|
1.99990000000000 -- 1.99996499999960 (169.030849991827) -- 1.99999999000211
|5.00000000000000e-9|
1.99990000000000 -- 1.99996999999957 (182.574184540568) -- 1.99999999000211
|5.00000000000000e-9|
1.99990000000000 -- 1.99997499999954 (199.999998176803) -- 1.99999999000211
|5.00000000000000e-9|
```



```
1.99990000000000 -- 1.99997999999951 (223.606795032113) -- 1.99999999000211
|5.00000000000000e-9|
1.99990000000000 -- 1.99998499999948 (258.198885301200) -- 1.99999999000211
|5.00000000000000e-9|
1.99990000000000 -- 1.99998999999945 (316.227757368620) -- 1.99999999000211
```

```
|5.00000000000000e-9|
1.99990000000000 -- 1.99999499999942 (447.213569680140) -- 1.99999999000211
|5.00000000000000e-9|
integrating...
```

Out[ ]:
```
2.00020041273101
```

In [ ]:
```
Improper_integral_SP_exclusion_mod(f,X0,X1,0.00001,method = "trap",trace = t
```

```
0 -- 0.500000000000000 | checking...
 |0|
  !0 -- 0 (NAN) -- 0.500000000000000    |5.00000000000000e-6|
0.0000100000000000000 -- 0.500000000000000 | checking...
0.0000100000000000000 -- 0.0000100000000000000 (316.229347167527) -- 0.5000
00000000000     |5.00000000000000e-6|
integrating...
1.00000000000000e-9 -- 0.0000100000000000000 | checking...
1.00000000000000e-9 -- 1.00000000000000e-9 (31622.7766174952) -- 0.00001000
00000000000     |5.00000000000000e-10|
integrating...
```

Out[ ]:
```
1.57085970434016
```

In [ ]:
```
show(integral(f,x,X0,X1).n())
```

$$1.57079632679490$$

In [ ]:
```
a = 0
al = (-1/2)
csi =  (1-x)**(-1/2)

X0 = 0
X1 = 0.5

f = ((x - a)**al)*csi(x)

show("Необходимо рассчитать интеграл этой функции:")
show(f)
show("От {} до {}".format(X0.n(digits = 5),X1.n(digits = 5)))

show("Ее график:")
show(plot(f,xmin = X0, xmax = X1,ymin = 0,ymax = 5))

show("Ее производная")
show(integral(f,x,X0,X1).n())

Improper_integral_SP_exclusion_mod(f,X0,X1,0.0001,method = "trap",trace = tr
```

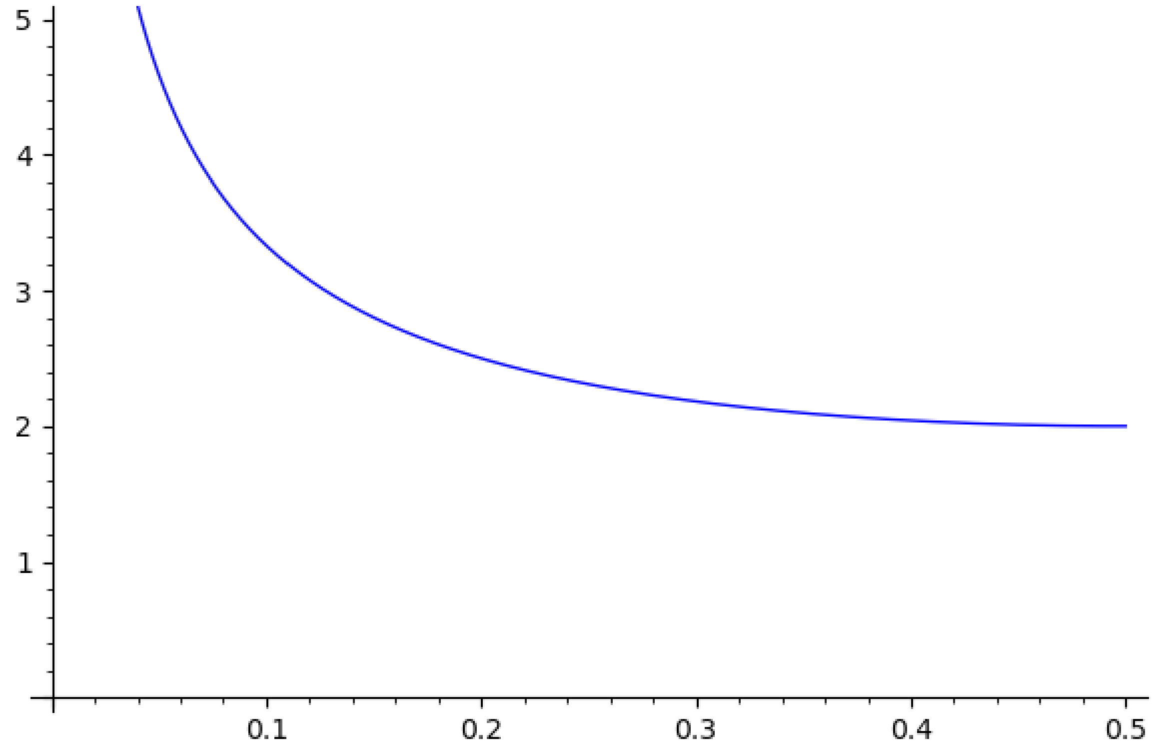
Необходимо рассчитать интеграл этой функции:

$$\frac{1}{\sqrt{x}\sqrt{-x+1}}$$

От 0.00000 до 0.50000

24

Ее график:



Ее производная

1.57079632679490

```
0 -- 0.500000000000000 | checking...
 |0|
  !0 -- 0 (NAN) -- 0.500000000000000    |0.0000500000000000000|
0.000100000000000000 -- 0.500000000000000 | checking...
0.000100000000000000 -- 0.000100000000000000 (100.005000375031) -- 0.500000
000000000       |0.0000500000000000000|
0.000100000000000000 -- 0.0501000000000009 (4.58397450192027) -- 0.50000000
0000000         |0.0000500000000000000|
0.000100000000000000 -- 0.100099999999997 (3.33185302371375) -- 0.500000000
000000  |0.0000500000000000000|
0.000100000000000000 -- 0.150099999999992 (2.79979181298121) -- 0.500000000
000000  |0.0000500000000000000|
0.000100000000000000 -- 0.200099999999986 (2.49953145987590) -- 0.500000000
000000  |0.0000500000000000000|
0.000100000000000000 -- 0.250099999999981 (2.30909327974475) -- 0.500000000
000000  |0.0000500000000000000|
0.000100000000000000 -- 0.300099999999975 (2.18197115742444) -- 0.500000000
000000  |0.0000500000000000000|
0.000100000000000000 -- 0.350099999999970 (2.09643149782026) -- 0.500000000
000000  |0.0000500000000000000|
0.000100000000000000 -- 0.400099999999964 (2.04115644842808) -- 0.500000000
000000  |0.0000500000000000000|
0.000100000000000000 -- 0.450099999999959 (2.01003506476557) -- 0.500000000
000000  |0.0000500000000000000|
0.000100000000000000 -- 0.500099999999953 (2.00000004000000) -- 0.500000000
000000  |0.0000500000000000000|
integrating...
1.00000000000000e-8 -- 0.000100000000000000 | checking...
1.00000000000000e-8 -- 1.00000000000000e-8 (10000.0000500000) -- 0.00010000
0000000000      |5.00000000000000e-9|
1.00000000000000e-8 -- 0.0000500099999999935 (141.410752242731) -- 0.000100
000000000000    |5.00000000000000e-9|
integrating...
```

Out[ ]: 1.57099672563692

In [ ]:
```
a = 0
al = (-1/2)
csi =  (1-x)**(-1/2)
h = 0.01

X0 = 0
X1 = 0.5

f = ((x - a)**al)*csi(x)

show("Метод исключения особенности с дополнительным интегрированием")
show("Результат: {}\n\t Шаг интегрирования: {}\n\t Ошибка (Рунге Ромберг | 0
     format(Improper_integral_SP_exclusion_mod(f,X0,X1,h = h,method = "rect"
            h.n(digits = 2),
            abs(runge_rumbert_error_h(Improper_integral_SP_exclusion_mod(f,X
            abs(Improper_integral_SP_exclusion_mod(f,X0,X1,h = h,method = "r
```

Метод исключения особенности с дополнительным интегрированием

Результат: 1.56687645705918
Шаг интегрирования: 0.010
Ошибка (Рунге Ромберг | 0.01): 0.00356247139949031
Ошибка (Истинная) 0.00391986973571989

Несобственные интегралы первого рода

In [ ]:
```
def Improper_integral_1(f,X0,X1,h,
                        method = "rect",
                        trace = false,
                        eps = 1e-5):

    if( (X0 > 1/eps and X1 > 1/eps) or (X0 < 1/eps and X1 < 1/eps) ): #при д
        return false

    a = var("a")
    if(X0 < -1/eps):#первый \ предел в -бесконечности
        if(trace):
            print("searching for a")

        criteria = f(x = a)/a
        step = h*10

        if(X1 > 1/eps): #если и второй предел в бесконечности, то за базу бе
            a = 0
        else:
            a = X1

        while(a > -1/eps):
            if(trace):
                print(" a: {} | {}".format(a,step))
            try:
                criteria(a = a)
                if(trace):
                    print(" a: {} ({})  | {}".format(a, criteria(a = a),step
```

```
                break
            except(BaseException):
                a -= step
                step*=2

        if(a <= -1/eps): #на данной сетке интеграл не существует
            return false

        while(abs(criteria(a = a)) > eps):
            a -= step
            step*=2

            if(trace):
                print(" a: {} ({})  | {}".format(a, criteria(a = a),step))

            if(a <= -1/eps): #при данной погрешности интеграл не сходится
                return false
    else:
        a = X0

    b = var("b")
    if(X1 > 1/eps):#второй предел в бесконечности
        if(trace):
            print("searching for b")

        criteria = f(x = b)/b
        step = h*10

        b = a

        while(b < 1/eps):
            if(trace):
                print(" b: {} | {}".format(b,step))

            try:
                criteria(b = b)
                if(trace):
                        print(" b: {} ({}) | {}".format(b,criteria(b = b),st
                break
            except(BaseException):
                b += step
                step*=2

        if(b >= 1/eps): #на данной сетке интеграл не существует
            return false

        while(abs(criteria(b = b)) > eps):
            b += step
            step*=2

            if(trace):
                print(" b: {} ({}) | {}".format(b,criteria(b = b),step))

            if(b >= 1/eps): #при данной погрешности интеграл не сходится
                return false
    else:
        b = X1

    if(trace):
        print("integrating from {} to {}".format(a,b))
        print("w step {}".format(h*(b-a)/200))

    return integrate(f,a,b,h*(b-a)/1000,method = method)
```

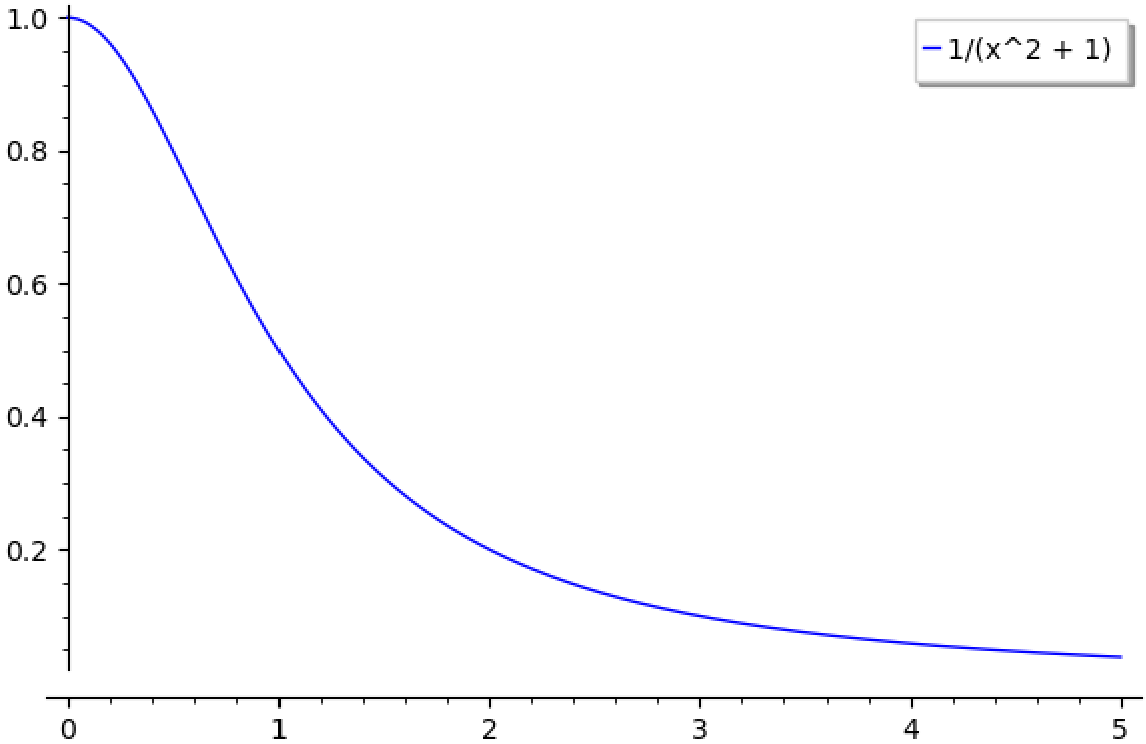
In [ ]:
```
f = exp(-x**2)
X0 = 0
X1 = infinity

Improper_integral_1(f,X0,X1,0.01,eps = 1e-20,trace = true)
```

```
searching for b
 b: 0 | 0.100000000000000
 b: 0.100000000000000 | 0.200000000000000
 b: 0.100000000000000 (9.90049833749168) | 0.200000000000000
 b: 0.300000000000000 (3.04643728423743) | 0.400000000000000
 b: 0.700000000000000 (0.875180563120594) | 0.800000000000000
 b: 1.50000000000000 (0.0702661497079096) | 1.60000000000000
 b: 3.10000000000000 (0.0000216305884847777) | 3.20000000000000
 b: 6.30000000000000 (9.19414743713458e-19) | 6.40000000000000
 b: 12.7000000000000 (7.06056235564279e-72) | 12.8000000000000
integrating from 0 to 12.7000000000000
w step 0.000635000000000000
```

Out[ ]:
```
0.886226925452794
```

In [ ]:
```
plot(f,xmin = 0, xmax = 5, legend_label = "{}".format(f(x)))
```

Out[ ]:



In [ ]:
```
show(integral(f,x,X0,X1).n())
```

0.886226925452758

In [ ]:
```
h = 0.01
```

In [ ]:
```
f = exp(-x**2)
X0 = 0
X1 = infinity
h = 0.01
e = 1e-4
m = 1e-4
show(f)
```

28

```
show("Метод отсечения бесконечности")
show("Результат: {}\n\t Шаг интегрирования: {}\n\t Ошибка (Рунге Ромберг | 0
     format(Improper_integral_1(f,X0,X1,h,eps = e),
            h.n(digits = 2),
            abs(runge_rumbert_error_h(Improper_integral_1(f,X0,X1,h,eps = e)
            abs(Improper_integral_1(f,X0,X1,h,eps = e) - integral(f,x,X0,X1)
```

$$e^{\left(-x^2\right)}$$

Метод отсечения бесконечности

Результат: 0.886216602099357

Шаг интегрирования: 0.010

Ошибка (Рунге Ромберг | 0.1): 0.0000103243858473657

Ошибка (Истинная) 0.0000103233534008984

Задача, которую пытались решить в начале

In [ ]:
```
f = 1/(x**2 + 1)
show(integral(f,x,X0,X1).n())
```

$$0.785398163397448$$

In [ ]:
```
f = 1/(x**2 + 1)
show(f)
integral(f,x,1,infinity).n()

Improper_integral_1(f,X0,X1,0.01,eps = 1e-10,trace = true)
```

$$\frac{1}{x^2+1}$$

```
searching for b
 b: 1 | 0.100000000000000
 b: 1 (1/2) | 0.100000000000000
 b: 1.10000000000000 (0.411353352529823) | 0.200000000000000
 b: 1.30000000000000 (0.285959393766085) | 0.400000000000000
 b: 1.70000000000000 (0.151217299259035) | 0.800000000000000
 b: 2.50000000000000 (0.0551724137931034) | 1.60000000000000
 b: 4.10000000000000 (0.0136946905684666) | 3.20000000000000
 b: 7.30000000000000 (0.00252323266476078) | 6.40000000000000
 b: 13.7000000000000 (0.000386839264030564) | 12.8000000000000
 b: 26.5000000000000 (0.0000536592236851813) | 25.6000000000000
 b: 52.1000000000000 (7.06849351127493e-6) | 51.2000000000000
 b: 103.300000000000 (9.07106624071104e-7) | 102.400000000000
 b: 205.700000000000 (1.14891225273928e-7) | 204.800000000000
 b: 410.500000000000 (1.44563261691967e-8) | 409.600000000000
 b: 820.100000000000 (1.81300465252726e-9) | 819.200000000000
 b: 1639.30000000000 (2.26999301982858e-10) | 1638.40000000000
 b: 3277.70000000000 (2.83983008785592e-11) | 3276.80000000000
integrating from 1 to 3277.70000000000
w step 0.163835000000000
```

29

Out[ ]: 0.785070703166263

In [ ]:
```
f = 1/(1+x**2)
X0 = 1
```

```
X1 = infinity
```

In [ ]:
```
f = 1/(1+x**2)
X0 = 1
X1 = infinity
h = 0.01
e = 1e-10
m = 1e-4
show(f)

show("Метод отсечения бесконечности")
show("Результат: {}\n\t Шаг интегрирования: {}\n\t Ошибка (Рунге Ромберг | 0
     format(Improper_integral_1(f,X0,X1,h,eps = e),
            h.n(digits = 2),
            abs(runge_rumbert_error_h(Improper_integral_1(f,X0,X1,h,eps = e)
            abs(Improper_integral_1(f,X0,X1,h,eps = e) - integral(f,x,X0,X1)
```

$$\frac{1}{x^2+1}$$

Метод отсечения бесконечности

Результат: 0.785070703166263
Шаг интегрирования: 0.010
Ошибка (Рунге Ромберг | 0.1): 0.00540909369114177
Ошибка (Истинная) 0.000327460231185572

In [ ]:
```
f = 1/(x**2 + 1)
show(integral(f,x,X0,X1).n())
```

0.785398163397448

Вывод В ходе выполнения этой работы я провел поверхностное исследование довольно интересной и глубокой темы численного интегрирования несобственных интегралов. Все методы, рассмотренные мной в этой работе, являются надстройками над обычными квадратурными методами, которые, проведя те или иные действия над функцией или сеткой интегрирования в итоге сводятся к запускам обычных методов над измененными функциями или сетками. Это означает, что они применимы с любыми квадратурными методами и их реализациями, а не только с теми, что рассматривал я. С другой стороны, существует целый класс методов с переменным шагом интегрирования, которые также применимы для вычисления несобственных интегралов. Эти методы также стоят рассмотрения. Также может иметь смысл более подробное рассмотрение модифицированного метода исключения особой точки, например, поиск более оптимального метода определения применимости или неприменимости квадратурных методов на данной сетке с данными значениями функции в узлах. В целом, рассмотренные методы позволяют с достаточно высокой точностью вычислять несобственные интегралы различных видов.