In [ ]:
```
import sympy as sp
from scipy import *
from sympy import *
init_printing()
from IPython.display import display, Latex, HTML, Math
import numpy as np
import pandas as pd
import math
from numpy import inf as Infinite
from sympy import Rational as R
```

Assignment 1

In [ ]:
```
# a)
# i. False. A pivot in the last column would result in inconsistency.
# ii. False. A may not be a square matrix. The invertible matrix theorem does no
# iii. False. det(3A) = 2*3^n, where n is the number of rows/columns in A.
# iv. True. Eigenspaces corresponding to distinct eigenvalues are always disjoin
# v. False. The null space of A, which coincide with the eigenspace correspondin
#    dimension 1 or 2. Thus rank(A) could be 4 or 5.
# vi. True. One can always obtain an orthogonal basis by the Gram-Schmidt proces
```

a. Determine whether each statement is true or false.

i. If the augmented matrix of a system of linear equations has a pivot in every column, then the system has a unique solution.	○ True	◉ False ✔
ii. If a matrix A has a pivot in every row, then A is invertible.	○ True	◉ False ✔
iii. If det(A) = 2, then det(3A) = 6.	○ True	◉ False ✔
iv. Eigenvectors corresponding to distinct eigenvalues are linearly independent.	◉ True ✔	○ False
v. If the characteristic polynomial of A is $p(\lambda) = \lambda^2(\lambda - 3)^4$, then the rank(A) = 4.	○ True	◉ False ✔
vi. Every non-trivial subspace of $\mathbb{R}^n$ has an orthogonal basis.	◉ True ✔	○ False

In [ ]:
```
# b) We can do row operations: subtract b times r1 from r2 and and b^2 times r1
# and you will get two rows of zeros. That means free variables which means det
# You could also just try to find the determinant using Python.
```

b. Find the determinant of matrix A below (where b is any real number). State your answer as a positive integer.

$$A = \begin{bmatrix} 1 & b & b^2 \\ b & b^2 & b^3 \\ b^2 & b^3 & b^4 \end{bmatrix}$$

$\det(A) = \boxed{0}$ ✔

In [ ]:
```
# c)
# Use the known eigval to solve for b
b = Symbol('b')
b = solve((-1)**2 + b*(-1) + 2)
l1 = -1
b
```

Out[ ]: $[3]$

In [ ]:
```
# l^2 +3l + 2 = -(l+1)(l+2)(l+1)
l1 = -2
l2 = -1
l3 = -1
l1, l2, l3
```

Out[ ]: $(-2,\ -1,\ -1)$

c. The characteristic polynomial of a matrix is $-(\lambda+1)\left(\lambda^2+b\lambda+2\right)$. Find the value of b and then state the eigenvalues s.t. $\lambda_3 \geq \lambda_2 \geq \lambda_1$. State all answers as positive integers. Please note that all eigenvalues are negative.

$b =$ 3 ✔

$\lambda_1 = -$ 2 ✔

$\lambda_2 = -$ 1 ✔

$\lambda_3 = -$ 1 ✔

In [ ]:
```
# d) Trace is sum of eigenvalues.
l2 = Symbol('l2')
l1 = 0
l2 = solve(l1 + l2 + 3*l2 - 16, l2)[0]
l2, 3*l2
```

Out[ ]: $(4,\ 12)$

d. Consider matrix *A* below

$$A = \begin{bmatrix} 4 & a & b \\ a & 4 & c \\ b & c & 8 \end{bmatrix}$$

It is known that $\lambda_1 = 0$ and that $\lambda_3 = 3\lambda_2$. What are the eigenvalues of A? State your answers as positive integers such that $\lambda_3 \geq \lambda_2 \geq \lambda_1$, and determine whether it is possible to determine whether A is diagonalizable or not.

$\lambda_1 = 0$

$\lambda_2 =$ 4 ✔

$\lambda_3 =$ 12 ✔

A	A is not diagonalizable	
B	A is diagonalizable	✔
C	There is insufficient information to determine whether A is diagonalizable or not.	

Assignment 2

$$
\begin{aligned}
A^3 - 4A^2 + 3A - 5I_n &= 0 \\
5I_n &= A^3 - 4A^2 + 3A \\
5 \cdot I_n A^{-1} &= \left(A^3 - 4A^2 - 3A\right) A^{-1} \\
A^{-1} &= \frac{1}{5}\left(A^2 - 4A + 3 \cdot I_n\right)
\end{aligned}
$$

a. Stating all answers as positive integers, if A is an n x n matrix and it satisfies the equation $A^3 - 4A^2 + 3A - 5I_n = 0$, then A is nonsingular and an expression for its inverse is

$A^{-1} = \frac{1}{5}\left(A^{2} - 4A + 3I_n\right)$ ✔

Assignment 3

In [ ]:
```
# a)
p, q = symbols('p q')
Matrix([[1,0,1,0, q],[0,1,0,2, 0],[1,0,2,3, 0],[0,2,3,p, 3]]).echelon_form()
```

Out[ ]:
$$\begin{bmatrix} 1 & 0 & 1 & 0 & q \\ 0 & 1 & 0 & 2 & 0 \\ 0 & 0 & 1 & 3 & -q \\ 0 & 0 & 0 & p-13 & 3q+3 \end{bmatrix}$$

In [ ]:
```
# unique: p != 13, q whatever
# no solution: p = 13, q != -1
# inf. solutions: p = 13, q = -1
```

a unique solution:

$p \neq 13$ ✔

no solution

$p = 13$ ✔ and $q \neq -1$ ✔

an infinite number of solutions

$p = 13$ ✔ and $q = -1$ ✔

In [ ]:
```
# b)
# Just insert p = 13 in the coefficient matrix and find nullspace
A = Matrix([[1,0,1,0],[0,1,0,2],[1,0,2,3],[0,2,3,13]])
A.nullspace()
```

Out[ ]:
$$\left[\begin{bmatrix} 3 \\ -2 \\ -3 \\ 1 \end{bmatrix}\right]$$

b. Using the value for *p* found for the infinite case in part (a), what is a basis for the nullspace of the equivalent homogeneous system (i.e. the coefficient matrix A for the above system, s.t. $A\vec{x} = \vec{0}$

$$\left\{\begin{bmatrix} \boxed{3} \\ -2 \\ -3 \\ \boxed{1} \end{bmatrix}\right\}$$

Assignment 4

In [ ]:
```
# a)
W1 = Matrix([[1,1,0], [2,1,0]]).T
W2 = Matrix([[1,2,4,0], [2,2,3,1]]).T
W3 = Matrix([1, -1, 1])

# We use "the trick" where just find the nullspace of the transpose since this w
# the orthogonal complement.
W1C = W1.T.nullspace()[0]
W2C = W2.T.nullspace()
W3C = W3.T.nullspace()
W1C, (2*W2C[0], 2*W2C[1]), W3C
```

Out[ ]:
$$\left(\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \left(\begin{bmatrix} 2 \\ -5 \\ 2 \\ 0 \end{bmatrix}, \begin{bmatrix} -2 \\ 1 \\ 0 \\ 2 \end{bmatrix}\right), \left[\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}\right]\right)$$

In [ ]:
```
def pivots(M, n):
    p = len(M.rref()[1])
    return p == n
```

In [ ]:
```
# b)

# We must check 3 things (technically 2):
    # 1) Make sure that the number entries (i.e. number of rows) match the dimen
    # 2) Make sure that we have as many pivots as the dimension from (1)
    # 3) Make sure we have no more columns than pivots, since then it would not

V1 = Matrix.hstack(W1, W1C)
V2 = Matrix.hstack(W2, W2C[0], W2C[1])
V3 = Matrix.hstack(W2, W2C[0], W2C[1])
V4 = Matrix.hstack(W1, W3C[0], W3C[1])
V5 = Matrix.hstack(W3, W3C[0], W3C[1])

display(
```

```
    V1.shape[0] == 3 and pivots(V1, 3) and V1.shape[1] == 3,
    V2.shape[0] == 2 and pivots(V2, 2) and V2.shape[1] == 2,
    V3.shape[0] == 4 and pivots(V3, 4) and V3.shape[1] == 4,
    V4.shape[0] == 3 and pivots(V4, 3) and V4.shape[1] == 3,
    V5.shape[0] == 3 and pivots(V5, 3) and V5.shape[1] == 3
)
```

```
True
False
True
False
True
```

In [ ]:
```
# Many thought V4 was a basis since it spans all of R^3, but it consists of 4 ve
# so it fails the definition of a basis which is the MINIMUM number of vectors i
# The MIN here would be 3 vectors.
```

Assignment 5

In [ ]:
```
# a)
Z = Matrix([[1,0,-1,5],[0,1,3,2],[0,0,0,0]])
a1 = Matrix([1,2,3])
a2 = Matrix([4, -1, 2])

# The rowspace is just the two pivot rows in R
# The colspace is just the two pivot columns in A
# The nullspace:
Z.nullspace()
```

Out[ ]:
$$\left[\begin{bmatrix}1\\-3\\1\\0\end{bmatrix}, \begin{bmatrix}-5\\-2\\0\\1\end{bmatrix}\right]$$

All answers in this exercise must be stated as positive integers.

A portion of the matrix A and the reduced row echelon form (Z) of A are shown below:

$$A = \begin{bmatrix}1 & 4 & * & *\\ 2 & -1 & * & *\\ 3 & 2 & * & *\end{bmatrix} \text{ and } Z = \begin{bmatrix}1 & 0 & -1 & 5\\ 0 & 1 & 3 & 2\\ 0 & 0 & 0 & 0\end{bmatrix}$$

a. Find the following vector spaces with regard to A.

A basis of the row space of A. Use rows from Z:

$$\left\{\begin{bmatrix}1 & \boxed{0} & -1 & \boxed{5}\end{bmatrix}, \begin{bmatrix}\boxed{0} & 1 & \boxed{3} & \boxed{2}\end{bmatrix}\right\}$$

A basis for the column space of A. Use columns from A.

$$\left\{\begin{bmatrix}1\\ \boxed{2}\\ \boxed{3}\end{bmatrix}, \begin{bmatrix}\boxed{4}\\ -1\\ \boxed{2}\end{bmatrix}\right\}$$

A basis of the nullspace.

$$\left\{\begin{bmatrix}\boxed{1}\\-3\\\boxed{1}\\\boxed{0}\end{bmatrix},\begin{bmatrix}-5\\-2\\\boxed{0}\\\boxed{1}\end{bmatrix}\right\}$$

✔

In [ ]:
```
# b)
# If it is to be consistent, then [9, 0, a] must be a linear combination of a1 a
# That means we put all there in a matrix and reduce to echelon form.
# We now define a such that we get a row of zeros
a = symbols('a')
solve(Matrix.hstack(a1, a2, Matrix([9, 0, a])).echelon_form()[-1,-1])
```

Out[ ]: $[7]$

b. Let $\vec{b} = [9, 0, a]^{\mathrm{T}}$ for some $a \in \mathbb{R}$. Find the value of a for which the system of equations $A\vec{x} = \vec{b}$ is consistent.

$a = \boxed{7}$ ✔

In [ ]:
```
# c)
# Since a only has two pivot columns (a1 and a2), which we know from looking at
# a3 and a4 can be derived from the scales stated in Z (the reduced echelon form
# To get a3, I need -1 of a1 and 3 of a3
# To get a4, I need 5 of a1 and 2 of a4
a3 = Z[0,2]*a1 + Z[1,2]*a2
a4 = Z[0,3]*a1 + Z[1,3]*a2
a3, a4
```

Out[ ]: $\left(\begin{bmatrix}11\\-5\\3\end{bmatrix}, \begin{bmatrix}13\\8\\19\end{bmatrix}\right)$

c. Find the missing columns of A, i.e. find $\mathbf{a}_3$ and $\mathbf{a}_4$.

$$a_3 = \begin{bmatrix}\boxed{11}\\-5\\\boxed{3}\end{bmatrix}, \quad a_4 = \begin{bmatrix}13\\\boxed{8}\\\boxed{19}\end{bmatrix}$$ ✔

In [ ]:
```
# We can check our result by rref'ing our new A and comparing it to Z
A = Matrix.hstack(a1, a2, a3, a4)
A.rref()[0]==Z
```

Out[ ]: True

Assignment 6

In [ ]:
```
# a)
```

```
x = pd.DataFrame([
    math.sin(-math.pi / 2),
    math.sin(0),
    math.sin(math.pi / 2)
])
y = pd.DataFrame([
    1,
    1,
    -1,
])

X = x.astype(int)
X1 = Matrix([ones(len(x), 1)]).row_join(Matrix((X)))

display(Math(latex(X1) + r'\left[\begin{array}{l}\alpha \\ \beta \end{array}\rig
```

$$\begin{bmatrix}1 & -1\\1 & 0\\1 & 1\end{bmatrix}\begin{bmatrix}\alpha\\\beta\end{bmatrix}=\begin{bmatrix}1\\1\\-1\end{bmatrix}$$

In [ ]:
```
# b)
X1tX1 = X1.T*X1
X1ty = X1.T*Matrix(y)
Mat, _ = X1tX1.row_join(X1ty).rref()
B1 = Mat[:,-1]
display(Latex("$$y = {}{}sin x$$".format(latex(B1[0]), B1[1])))
```

$$y=\frac{1}{3}-1sinx$$

In [ ]:
```
# c)
Y = Matrix(y)
display(Latex("$$\epsilon = {}$$".format(latex(Y-X1*B1))))
display(Latex("$$e = {}$$".format(latex((Y-X1*B1).norm()))))
```

$$\epsilon=\begin{bmatrix}-\frac{1}{3}\\\frac{2}{3}\\-\frac{1}{3}\end{bmatrix}$$

$$e=\frac{\sqrt{6}}{3}$$

Assignment 7

In [ ]:
```
# a)
AtA = Matrix([[81,-27],[-27,9]])
AAt = Matrix([[10,-20,-20],[-20,40,40],[-20,40,40]])
vecs = AAt.eigenvects()
display(AtA)
display(AAt)
```

```
display(vecs)
display(AtA.eigenvects())
```

$$\begin{bmatrix} 81 & -27 \\ -27 & 9 \end{bmatrix}$$

$$\begin{bmatrix} 10 & -20 & -20 \\ -20 & 40 & 40 \\ -20 & 40 & 40 \end{bmatrix}$$

$$\left[\left(0,\ 2,\ \left[\begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix},\ \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix}\right]\right),\ \left(90,\ 1,\ \left[\begin{bmatrix} -\frac{1}{2} \\ 1 \\ 1 \end{bmatrix}\right]\right)\right]$$

$$\left[\left(0,\ 1,\ \left[\begin{bmatrix} \frac{1}{3} \\ 1 \end{bmatrix}\right]\right),\ \left(90,\ 1,\ \left[\begin{bmatrix} -3 \\ 1 \end{bmatrix}\right]\right)\right]$$

In [ ]:
```
# b)
# The Gram matrix, as mentioned, is AtA. Some thought it had something to do wit
# The Gram Matrix is the matrix you set up to test whether a set of vectors are
# In class we did U.T * U:
x1 = 2*vecs[1][2][0]
x2 = vecs[0][2][0]
x3 = vecs[0][2][1]
X = Matrix.hstack(x1, x2, x3)
X, (X.T * X)
```

Out[ ]:
$$\left(\begin{bmatrix} -1 & 2 & 2 \\ 2 & 1 & 0 \\ 2 & 0 & 1 \end{bmatrix},\ \begin{bmatrix} 9 & 0 & 0 \\ 0 & 5 & 4 \\ 0 & 4 & 5 \end{bmatrix}\right)$$

In [ ]:
```
# c)
# From above we see that the vectors are not orthogonal, more specifically
# x2 and x3 are not ortogonal to each other
# We use Gram Schmidt to fix this
u1 = vecs[1][2][0].normalized()
u2 = GramSchmidt([x2, x3], True)[0]
u3 = GramSchmidt([x2, x3], True)[1]
U = Matrix.hstack(u1, u2, u3)
u1, u2, u3
```

Out[ ]:
$$\left(\begin{bmatrix} -\frac{1}{3} \\ \frac{2}{3} \\ \frac{2}{3} \end{bmatrix},\ \begin{bmatrix} \frac{2\sqrt{5}}{5} \\ \frac{\sqrt{5}}{5} \\ 0 \end{bmatrix},\ \begin{bmatrix} \frac{2\sqrt{5}}{15} \\ -\frac{4\sqrt{5}}{15} \\ \frac{\sqrt{5}}{3} \end{bmatrix}\right)$$

In [ ]:
```
# d)
# Since AAt is 3 x 3 and AtA is 2 x 2, A must be 3 x 2
# Since only 1 non-zero singular value, rank = 1
# m (number of columns) = dim nul A + dim col A (Rank) so dim nul A = 1
# dim Row is the same as Rank, so 1
# dim Nul A.T is 2 since it is n - rank = 2
```

In [ ]:
```
# In case you don't believe me, here is A:
A = Matrix([[-3,1],[6,-2],[6,-2]])
A, A.rank(),A.nullspace(),A.rowspace(),A.T.nullspace()
```

Out[ ]:
$$\left(\begin{bmatrix} -3 & 1 \\ 6 & -2 \\ 6 & -2 \end{bmatrix}, 1, \left[\begin{bmatrix} \frac{1}{3} \\ 1 \end{bmatrix}\right], [[-3 \quad 1]], \left[\begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix}\right]\right)$$

Assignment 8

In total, 5 + 3 litres enter A, which means 8 must leave A from pipe c. If 8 enters B, 8 must leave, 3 from b and 5 from d. We get:

In [ ]:
```
# We translate the problem to a matrix problem

A = Matrix([[-R(8,200), R(3,100)],[R(8,200), -R(8,100)]])
display(Math(r'A = ' + latex(A)))
```

$$A = \begin{bmatrix} -\frac{1}{25} & \frac{3}{100} \\ \frac{1}{25} & -\frac{2}{25} \end{bmatrix}$$

In [ ]:
```
# We find the eigenvalues and the corresponding eigenspaces.
l = symbols('l')
l1, l2 = solve(det(A-l*eye(np.shape(A)[0])))
display(Math(r'\lambda_0 =' +latex(l1) + r'\approx' + latex(round(l1, 2))))
display(Math(r'\lambda_1 =' +latex(l2) + r'\approx' + latex(round(l2, 2))))

v1 = (A-l1*eye(np.shape(A)[0])).nullspace()[0]
v2 = (A-l2*eye(np.shape(A)[0])).nullspace()[0]
display(Math(r'v_0 = '+ latex(v1) + r'=' + latex(v1.evalf(4))))
display(Math(r'v_1 = '+ latex(v2) + r'=' + latex(v2.evalf(4))))
```

$$\lambda_0 = -\frac{1}{10} \approx -0.1$$

$$\lambda_1 = -\frac{1}{50} \approx -0.02$$

$$v_0 = \begin{bmatrix} -\frac{1}{2} \\ 1 \end{bmatrix} = \begin{bmatrix} -0.5 \\ 1.0 \end{bmatrix}$$

$$v_1 = \begin{bmatrix} \frac{3}{2} \\ 1 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 1.0 \end{bmatrix}$$

In [ ]:
```
y0 = Matrix([100, 50])

# To solve the system we form the following augmented matrix and solve. The c's
C = v1.row_join(v2).row_join(y0).rref()[0][:, -1]
C
```

Out[ ]:
$$\begin{bmatrix} -\frac{25}{2} \\ \frac{125}{2} \end{bmatrix}$$

In [ ]:
```
# a) Wasn't able to use solve, so I just made a for loop to check
for t in range (0,100):
    y1 = float(C[0]*v1[0]*math.e**(l1*t) + C[1]*v2[0]*math.e**(l2*t))
    if y1 <= 50:
        print(t)
```

```
        print(y1)
        break
```

```
32
49.688428528900594
```

In [ ]:

```
# b)
# Many took my code from the "Do this first" exercise. The problem is that you s
# NEVER use code that you do not know what does! In my code the limit was set to
# but in this problem the limit is inf
# Learning outcome: Do not copy code that you don't understand yourself
# (I did give you 1 point out of 2.5, though :-))
t = symbols('t')
y1 = C[0]*v1[0]*math.e**(l1*t) + C[1]*v2[0]*math.e**(l2*t)
y2 = C[0]*v1[1]*math.e**(l1*t) + C[1]*v2[1]*math.e**(l2*t)
z=limit(y1/y2,t,Infinite)
z
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(log((54365636569181/20000000000000)**(1/(50*log(2**(7/25)*5**(1
3/50)*54365636569181**(49/50)/54365636569181)))) + 1) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
```

```
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(log((54365636569181/20000000000000)**(1/(50*log(2**(7/25)*5**(1
3/50)*54365636569181**(49/50)/54365636569181)))) + 1) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
```

```
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
```

```
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated
```

```
During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:
```

```
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
```

```
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
```

```
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))
```

```
ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
```

```
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
```

```
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated
```

```
During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:
```

```
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
```

```
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
```

```
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))
```

```
ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
```

```
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
```

```
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated
```

```
During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:
```

```
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
```

```
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
```

```
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))
```

```
ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
```

```
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
```

```
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated
```

```
During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:
```

```
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
```

```
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
```

```
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))
```

```
ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
```

```
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
```

```
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated
```

```
During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:
```

```
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
```

```
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
```

```
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))
```

```
ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
```

```
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
```

```
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated
```

```
During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:
```

```
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
```

```
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
```

```
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))
```

```
ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
```

```
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
```

```
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated
```

```
During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     71 try:
---> 72     retval = cfunc(*args, **kwargs)
     73 except TypeError as e:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:469, in limitinf(e, x)
    468 else:
--> 469     raise ValueError("{} could not be evaluated".format(sig))

ValueError: sign(-1 + log(20000000000000**(1/(50*log(54365636569181*2**(7/25)*543
65636569181**(1/10))))*54365636569181**(11/(10*log(54365636569181*2**(7/25)*54365
636569181**(1/10))))/5**(13/(50*log(54365636569181*2**(7/25)*54365636569181**(1/1
0)))))) could not be evaluated

During handling of the above exception, another exception occurred:

KeyboardInterrupt                         Traceback (most recent call last)
Cell In[137], line 10
      8 y1 = C[0]*v1[0]*math.e**(l1*t) + C[1]*v2[0]*math.e**(l2*t)
      9 y2 = C[0]*v1[1]*math.e**(l1*t) + C[1]*v2[1]*math.e**(l2*t)
---> 10 z=limit(y1/y2,t,Infinite)
     11 z

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:64, in limit(e, z, z0, dir)
     13 def limit(e, z, z0, dir="+"):
     14     """Computes the limit of ``e(z)`` at the point ``z0``.
     15
     16     Parameters
   (...)
     61      limit_seq : returns the limit of a sequence.
     62     """
---> 64     return Limit(e, z, z0, dir).doit(deep=False)

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:381, in Limit.doit(self, **hints)
    379 if l is not None:
    380     raise
--> 381 r = heuristics(e, z, z0, dir)
```

```
    382 if r is None:
    383     return self

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:77, in heuristics(e, z, z0, dir)
     75 rv = None
     76 if z0 is S.Infinity:
---> 77     rv = limit(e.subs(z, 1/z), z, S.Zero, "+")
     78     if isinstance(rv, Limit):
     79         return

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:64, in limit(e, z, z0, dir)
     13 def limit(e, z, z0, dir="+"):
     14     """Computes the limit of ``e(z)`` at the point ``z0``.
     15
     16     Parameters
   (...)
     61      limit_seq : returns the limit of a sequence.
     62     """
---> 64     return Limit(e, z, z0, dir).doit(deep=False)

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:381, in Limit.doit(self, **hints)
    379 if l is not None:
    380     raise
--> 381 r = heuristics(e, z, z0, dir)
    382 if r is None:
    383     return self

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:126, in heuristics(e, z, z0, dir)
    124             if rat_e is S.NaN or rat_e == e:
    125                 return
--> 126             return limit(rat_e, z, z0, dir)
    127 return rv

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:64, in limit(e, z, z0, dir)
     13 def limit(e, z, z0, dir="+"):
     14     """Computes the limit of ``e(z)`` at the point ``z0``.
     15
     16     Parameters
   (...)
     61      limit_seq : returns the limit of a sequence.
     62     """
---> 64     return Limit(e, z, z0, dir).doit(deep=False)

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:381, in Limit.doit(self, **hints)
    379 if l is not None:
    380     raise
--> 381 r = heuristics(e, z, z0, dir)
    382 if r is None:
    383     return self

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:126, in heuristics(e, z, z0, dir)
    124             if rat_e is S.NaN or rat_e == e:
    125                 return
```

```
--> 126             return limit(rat_e, z, z0, dir)
    127 return rv

    [... skipping similar frames: limit at line 64 (77 times), Limit.doit at line
381 (76 times), heuristics at line 126 (76 times)]

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:381, in Limit.doit(self, **hints)
    379 if l is not None:
    380     raise
--> 381 r = heuristics(e, z, z0, dir)
    382 if r is None:
    383     return self

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:126, in heuristics(e, z, z0, dir)
    124             if rat_e is S.NaN or rat_e == e:
    125                 return
--> 126             return limit(rat_e, z, z0, dir)
    127 return rv

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:64, in limit(e, z, z0, dir)
     13 def limit(e, z, z0, dir="+"):
     14     """Computes the limit of ``e(z)`` at the point ``z0``.
     15
     16     Parameters
   (...)
     61      limit_seq : returns the limit of a sequence.
     62     """
---> 64     return Limit(e, z, z0, dir).doit(deep=False)

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\limits.py:375, in Limit.doit(self, **hints)
    372 l = None
    374 try:
--> 375     r = gruntz(e, z, z0, dir)
    376     if r is S.NaN or l is S.NaN:
    377         raise PoleError()

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:732, in gruntz(e, z, z0, dir)
    729     else:
    730         raise NotImplementedError("dir must be '+' or '-'")
--> 732 r = limitinf(e0, z)
    734 # This is a bit of a heuristic for nice results... we always rewrite
    735 # tractable functions in terms of familiar intractable ones.
    736 # It might be nicer to rewrite the exactly to what they were initially,
    737 # but that would take some work to implement.
    738 return r.rewrite('intractable', deep=True)

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     69 @wraps(func)
     70 def wrapper(*args, **kwargs):
     71     try:
---> 72         retval = cfunc(*args, **kwargs)
     73     except TypeError as e:
     74         if not e.args or not e.args[0].startswith('unhashable type:'):
```

```
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:452, in limitinf(e, x)
    450     c0, e0 = mrv_leadterm(e.min, x)
    451 else:
--> 452     c0, e0 = mrv_leadterm(e, x)
    453 sig = sign(e0, x)
    454 if sig == 1:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     69 @wraps(func)
     70 def wrapper(*args, **kwargs):
     71     try:
---> 72         retval = cfunc(*args, **kwargs)
     73     except TypeError as e:
     74         if not e.args or not e.args[0].startswith('unhashable type:'):

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\series\gruntz.py:556, in mrv_leadterm(e, x)
    554 f, logw = rewrite(exps, Omega, x, w)
    555 try:
--> 556     lt = f.leadterm(w, logx=logw)
    557 except (NotImplementedError, PoleError, ValueError):
    558     n0 = 1

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\expr.py:3538, in Expr.leadterm(self, x, logx, cdir)
   3536 from .symbol import Dummy
   3537 from sympy.functions.elementary.exponential import log
-> 3538 l = self.as_leading_term(x, logx=logx, cdir=cdir)
   3539 d = Dummy('logx')
   3540 if l.has(log(x)):

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     69 @wraps(func)
     70 def wrapper(*args, **kwargs):
     71     try:
---> 72         retval = cfunc(*args, **kwargs)
     73     except TypeError as e:
     74         if not e.args or not e.args[0].startswith('unhashable type:'):

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\expr.py:3501, in Expr.as_leading_term(self, logx, cdir, *symbols)
   3499 if x not in self.free_symbols:
   3500     return self
-> 3501 obj = self._eval_as_leading_term(x, logx=logx, cdir=cdir)
   3502 if obj is not None:
   3503     from sympy.simplify.powsimp import powsimp

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\add.py:1048, in Add._eval_as_leading_term(self, x, logx, cdir)
   1045 infinite = [t for t in expr.args if t.is_infinite]
   1047 _logx = Dummy('logx') if logx is None else logx
-> 1048 leading_terms = [t.as_leading_term(x, logx=_logx, cdir=cdir) for t in exp
r.args]
   1050 min, new_expr = Order(0), 0
```

```
   1052 try:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\add.py:1048, in <listcomp>(.0)
   1045 infinite = [t for t in expr.args if t.is_infinite]
   1047 _logx = Dummy('logx') if logx is None else logx
-> 1048 leading_terms = [t.as_leading_term(x, logx=_logx, cdir=cdir) for t in exp
r.args]
   1050 min, new_expr = Order(0), 0
   1052 try:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     69 @wraps(func)
     70 def wrapper(*args, **kwargs):
     71     try:
---> 72         retval = cfunc(*args, **kwargs)
     73     except TypeError as e:
     74         if not e.args or not e.args[0].startswith('unhashable type:'):

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\expr.py:3501, in Expr.as_leading_term(self, logx, cdir, *symbols)
   3499 if x not in self.free_symbols:
   3500     return self
-> 3501 obj = self._eval_as_leading_term(x, logx=logx, cdir=cdir)
   3502 if obj is not None:
   3503     from sympy.simplify.powsimp import powsimp

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\mul.py:2035, in Mul._eval_as_leading_term(self, x, logx, cdir)
   2034 def _eval_as_leading_term(self, x, logx=None, cdir=0):
-> 2035     return self.func(*[t.as_leading_term(x, logx=logx, cdir=cdir) for t i
n self.args])

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\mul.py:2035, in <listcomp>(.0)
   2034 def _eval_as_leading_term(self, x, logx=None, cdir=0):
-> 2035     return self.func(*[t.as_leading_term(x, logx=logx, cdir=cdir) for t i
n self.args])

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     69 @wraps(func)
     70 def wrapper(*args, **kwargs):
     71     try:
---> 72         retval = cfunc(*args, **kwargs)
     73     except TypeError as e:
     74         if not e.args or not e.args[0].startswith('unhashable type:'):

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\expr.py:3501, in Expr.as_leading_term(self, logx, cdir, *symbols)
   3499 if x not in self.free_symbols:
   3500     return self
-> 3501 obj = self._eval_as_leading_term(x, logx=logx, cdir=cdir)
   3502 if obj is not None:
   3503     from sympy.simplify.powsimp import powsimp

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
```

```
\core\power.py:1825, in Pow._eval_as_leading_term(self, x, logx, cdir)
   1823 from sympy.functions.elementary.complexes import im
   1824 try:
-> 1825     f = b.as_leading_term(x, logx=logx, cdir=cdir)
   1826 except PoleError:
   1827     return self

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     69 @wraps(func)
     70 def wrapper(*args, **kwargs):
     71     try:
---> 72         retval = cfunc(*args, **kwargs)
     73     except TypeError as e:
     74         if not e.args or not e.args[0].startswith('unhashable type:'):

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\expr.py:3501, in Expr.as_leading_term(self, logx, cdir, *symbols)
   3499 if x not in self.free_symbols:
   3500     return self
-> 3501 obj = self._eval_as_leading_term(x, logx=logx, cdir=cdir)
   3502 if obj is not None:
   3503     from sympy.simplify.powsimp import powsimp

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\add.py:1048, in Add._eval_as_leading_term(self, x, logx, cdir)
   1045 infinite = [t for t in expr.args if t.is_infinite]
   1047 _logx = Dummy('logx') if logx is None else logx
-> 1048 leading_terms = [t.as_leading_term(x, logx=_logx, cdir=cdir) for t in exp
r.args]
   1050 min, new_expr = Order(0), 0
   1052 try:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\add.py:1048, in <listcomp>(.0)
   1045 infinite = [t for t in expr.args if t.is_infinite]
   1047 _logx = Dummy('logx') if logx is None else logx
-> 1048 leading_terms = [t.as_leading_term(x, logx=_logx, cdir=cdir) for t in exp
r.args]
   1050 min, new_expr = Order(0), 0
   1052 try:

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     69 @wraps(func)
     70 def wrapper(*args, **kwargs):
     71     try:
---> 72         retval = cfunc(*args, **kwargs)
     73     except TypeError as e:
     74         if not e.args or not e.args[0].startswith('unhashable type:'):

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\expr.py:3501, in Expr.as_leading_term(self, logx, cdir, *symbols)
   3499 if x not in self.free_symbols:
   3500     return self
-> 3501 obj = self._eval_as_leading_term(x, logx=logx, cdir=cdir)
   3502 if obj is not None:
   3503     from sympy.simplify.powsimp import powsimp
```

```
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\mul.py:2035, in Mul._eval_as_leading_term(self, x, logx, cdir)
   2034 def _eval_as_leading_term(self, x, logx=None, cdir=0):
-> 2035     return self.func(*[t.as_leading_term(x, logx=logx, cdir=cdir) for t i
n self.args])

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\mul.py:2035, in <listcomp>(.0)
   2034 def _eval_as_leading_term(self, x, logx=None, cdir=0):
-> 2035     return self.func(*[t.as_leading_term(x, logx=logx, cdir=cdir) for t i
n self.args])

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\cache.py:72, in __cacheit.<locals>.func_wrapper.<locals>.wrapper(*args, **k
wargs)
     69 @wraps(func)
     70 def wrapper(*args, **kwargs):
     71     try:
---> 72         retval = cfunc(*args, **kwargs)
     73     except TypeError as e:
     74         if not e.args or not e.args[0].startswith('unhashable type:'):

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\expr.py:3504, in Expr.as_leading_term(self, logx, cdir, *symbols)
   3502 if obj is not None:
   3503     from sympy.simplify.powsimp import powsimp
-> 3504     return powsimp(obj, deep=True, combine='exp')
   3505 raise NotImplementedError('as_leading_term(%s, %s)' % (self, x))

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\simplify\powsimp.py:117, in powsimp(expr, deep, combine, force, measure)
    114     return expr
    116 if deep or expr.is_Add or expr.is_Mul and _y not in expr.args:
--> 117     expr = expr.func(*[recurse(w) for w in expr.args])
    119 if expr.is_Pow:
    120     return recurse(expr*_y, deep=False)/_y

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\simplify\powsimp.py:117, in <listcomp>(.0)
    114     return expr
    116 if deep or expr.is_Add or expr.is_Mul and _y not in expr.args:
--> 117     expr = expr.func(*[recurse(w) for w in expr.args])
    119 if expr.is_Pow:
    120     return recurse(expr*_y, deep=False)/_y

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\simplify\powsimp.py:108, in powsimp.<locals>.recurse(arg, **kwargs)
    106 _force = kwargs.get('force', force)
    107 _measure = kwargs.get('measure', measure)
--> 108 return powsimp(arg, _deep, _combine, _force, _measure)

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\simplify\powsimp.py:117, in powsimp(expr, deep, combine, force, measure)
    114     return expr
    116 if deep or expr.is_Add or expr.is_Mul and _y not in expr.args:
--> 117     expr = expr.func(*[recurse(w) for w in expr.args])
    119 if expr.is_Pow:
    120     return recurse(expr*_y, deep=False)/_y
```

```
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\simplify\powsimp.py:117, in <listcomp>(.0)
    114     return expr
    116 if deep or expr.is_Add or expr.is_Mul and _y not in expr.args:
--> 117     expr = expr.func(*[recurse(w) for w in expr.args])
    119 if expr.is_Pow:
    120     return recurse(expr*_y, deep=False)/_y

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\simplify\powsimp.py:108, in powsimp.<locals>.recurse(arg, **kwargs)
    106 _force = kwargs.get('force', force)
    107 _measure = kwargs.get('measure', measure)
--> 108 return powsimp(arg, _deep, _combine, _force, _measure)

    [... skipping similar frames: <listcomp> at line 117 (1 times), powsimp at li
ne 117 (1 times), powsimp.<locals>.recurse at line 108 (1 times)]

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\simplify\powsimp.py:117, in powsimp(expr, deep, combine, force, measure)
    114     return expr
    116 if deep or expr.is_Add or expr.is_Mul and _y not in expr.args:
--> 117     expr = expr.func(*[recurse(w) for w in expr.args])
    119 if expr.is_Pow:
    120     return recurse(expr*_y, deep=False)/_y

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\simplify\powsimp.py:117, in <listcomp>(.0)
    114     return expr
    116 if deep or expr.is_Add or expr.is_Mul and _y not in expr.args:
--> 117     expr = expr.func(*[recurse(w) for w in expr.args])
    119 if expr.is_Pow:
    120     return recurse(expr*_y, deep=False)/_y

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\simplify\powsimp.py:108, in powsimp.<locals>.recurse(arg, **kwargs)
    106 _force = kwargs.get('force', force)
    107 _measure = kwargs.get('measure', measure)
--> 108 return powsimp(arg, _deep, _combine, _force, _measure)

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\simplify\powsimp.py:120, in powsimp(expr, deep, combine, force, measure)
    117     expr = expr.func(*[recurse(w) for w in expr.args])
    119 if expr.is_Pow:
--> 120     return recurse(expr*_y, deep=False)/_y
    122 if not expr.is_Mul:
    123     return expr

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\simplify\powsimp.py:108, in powsimp.<locals>.recurse(arg, **kwargs)
    106 _force = kwargs.get('force', force)
    107 _measure = kwargs.get('measure', measure)
--> 108 return powsimp(arg, _deep, _combine, _force, _measure)

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\simplify\powsimp.py:204, in powsimp(expr, deep, combine, force, measure)
    202 _n = S.NegativeOne
    203 for b, e in be:
--> 204     if (b.is_Symbol or b.is_Add) and -b in c_powers and b in c_powers:
    205         if (b.is_positive is not None or e.is_integer):
    206             if e.is_integer or b.is_negative:
```

```
File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\expr.py:189, in Expr.__neg__(self)
    184 def __neg__(self):
    185     # Mul has its own __neg__ routine, so we just
    186     # create a 2-args Mul with the -1 in the canonical
    187     # slot 0.
    188     c = self.is_commutative
--> 189     return Mul._from_args((S.NegativeOne, self), c)

File c:\Users\Adi\AppData\Local\Programs\Python\Python311\Lib\site-packages\sympy
\core\operations.py:108, in AssocOp._from_args(cls, args, is_commutative)
    105         return Order(obj, *order_symbols)
    106     return obj
--> 108 @classmethod
    109 def _from_args(cls, args, is_commutative=None):
    110     """Create new instance with already-processed args.
    111     If the args are not in canonical order, then a non-canonical
    112     result will be returned, so use with caution. The order of
    113     args may change if the sign of the args is changed."""
    114     if len(args) == 0:

KeyboardInterrupt:
```

is 3/2 infinite is kinda funny on compiler