In [4]:
```python
def harmonic_forward(n):
    """Calcule la série harmonique en ordre croissant"""
    somme = 0.0
    for k in range(1, n + 1):
        somme += 1 / k
    return somme
def harmonic_backward(n):
    """Calcule la série harmonique en ordre décroissant"""
    somme = 0.0
    for k in range(n, 0, -1):  # commence à n et descend jusqu'à 1
        somme += 1 / k
    return somme

# Programme principal
n_values = [10, 100, 500, 1000, 10000]
for n in n_values:
    H_forward = harmonic_forward(n)
    H_backward = harmonic_backward(n)
    diff = abs(H_forward - H_backward)
    print(f"n = {n}")
    print(f"  H_{n} (croissant)   = {H_forward}")
    print(f"  H_{n} (décroissant) = {H_backward}")
    print(f"  Différence          = {diff}")
    print("-" * 50)
```

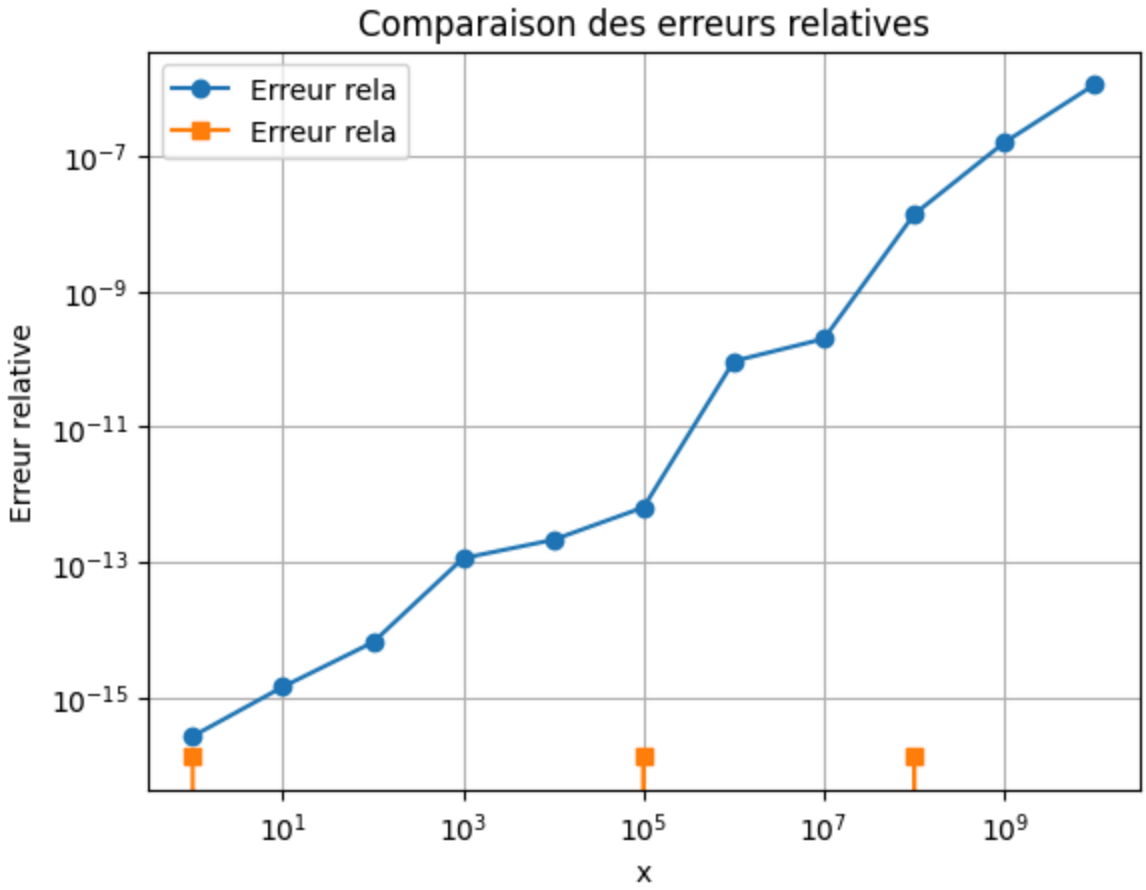
```
n = 10
  H_10 (croissant)   = 2.9289682539682538
  H_10 (décroissant) = 2.9289682539682538
  Différence         = 0.0
--------------------------------------------------
n = 100
  H_100 (croissant)   = 5.187377517639621
  H_100 (décroissant) = 5.1873775176396215
  Différence          = 8.881784197001252e-16
--------------------------------------------------
n = 500
  H_500 (croissant)   = 6.79282342999052
  H_500 (décroissant) = 6.7928234299905235
  Différence          = 3.552713678800501e-15
--------------------------------------------------
n = 1000
  H_1000 (croissant)   = 7.485470860550343
  H_1000 (décroissant) = 7.485470860550341
  Différence          = 2.6645352591003757e-15
--------------------------------------------------
n = 10000
  H_10000 (croissant)   = 9.787606036044348
  H_10000 (décroissant) = 9.787606036044386
  Différence          = 3.730349362740526e-14
--------------------------------------------------
```

In [ ]:
```python
pip install pandas
```

In [3]:
```
import math
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from decimal import Decimal, getcontext
def formula_direct(x):
    "Calcul direct de sqrt(x+1) - sqrt(x) en double précision (float)"
    return math.sqrt(x + 1.0) - math.sqrt(x)
def formula_stable(x):
    """Formule mathématiquement équivalente mais numériquement stable
       sqrt(x+1) - sqrt(x) = 1 / (sqrt(x+1) + sqrt(x))
    """
    return 1.0 / (math.sqrt(x + 1.0) + math.sqrt(x))
def reference_high_prec(x, prec=80):
    """Calcul de référence en haute précision avec Decimal."""
    getcontext().prec = prec
    X = Decimal(x)
    return (X + Decimal(1)).sqrt() - X.sqrt()
def analyze_subtraction_catastrophe():
    """Analyse comparative pour x variant de 1 à 1e10."""
    xs = np.logspace(0, 10, num=11, base=10)  # 1, 10, 100, ..., 1e10
    results = []
    for x in xs:
        direct = formula_direct(x)
        stable = formula_stable(x)
        ref = float(reference_high_prec(x))
        err_direct = abs(direct - ref)
        err_stable = abs(stable - ref)
        rel_err_direct = err_direct / abs(ref)
        rel_err_stable = err_stable / abs(ref)
        results.append({
            "x": x,
            "direct": direct,
            "stable": stable,
            "reference": ref,
            "err_direct": err_direct,
            "err_stable": err_stable,
            "rel_err_direct": rel_err_direct,
            "rel_err_stable": rel_err_stable
        })
    df = pd.DataFrame(results)
    print(df)
    # Tracer les erreurs relatives
    plt.figure()
    plt.loglog(df["x"], df["rel_err_direct"], 'o-', label="Erreur rela")
    plt.loglog(df["x"], df["rel_err_stable"], 's-', label="Erreur rela")
    plt.xlabel("x")
    plt.ylabel("Erreur relative")
    plt.title("Comparaison des erreurs relatives")
    plt.legend()
    plt.grid(True, which="both")
    plt.show()
    return df
analyze_subtraction_catastrophe()
```

	x	direct	stable	reference	err_direct	err_stable
0	1.000000e+00	0.414214	0.414214	0.414214	1.110223e-16	5.551115e-17
1	1.000000e+01	0.154347	0.154347	0.154347	2.220446e-16	0.000000e+00
2	1.000000e+02	0.049876	0.049876	0.049876	3.261280e-16	0.000000e+00
3	1.000000e+03	0.015807	0.015807	0.015807	1.804112e-15	0.000000e+00
4	1.000000e+04	0.005000	0.005000	0.005000	1.065120e-15	0.000000e+00
5	1.000000e+05	0.001581	0.001581	0.001581	1.015464e-15	2.168404e-19
6	1.000000e+06	0.000500	0.000500	0.000500	4.627939e-14	0.000000e+00
7	1.000000e+07	0.000158	0.000158	0.000158	3.125901e-14	0.000000e+00
8	1.000000e+08	0.000050	0.000050	0.000050	6.808832e-13	6.776264e-21
9	1.000000e+09	0.000016	0.000016	0.000016	2.470451e-12	0.000000e+00
10	1.000000e+10	0.000005	0.000005	0.000005	5.583154e-12	0.000000e+00

	rel_err_direct	rel_err_stable
0	2.680315e-16	1.340158e-16
1	1.438605e-15	0.000000e+00
2	6.538826e-15	0.000000e+00
3	1.141306e-13	0.000000e+00
4	2.130294e-13	0.000000e+00
5	6.422373e-13	1.371423e-16
6	9.255880e-11	0.000000e+00
7	1.976994e-10	0.000000e+00
8	1.361766e-08	1.355253e-16
9	1.562450e-07	0.000000e+00
10	1.116631e-06	0.000000e+00



Out[3]:

	x	direct	stable	reference	err_direct	err_stable	rel_err_direct	r
0	1.000000e+00	0.414214	0.414214	0.414214	1.110223e-16	5.551115e-17	2.680315e-16	
1	1.000000e+01	0.154347	0.154347	0.154347	2.220446e-16	0.000000e+00	1.438605e-15	0
2	1.000000e+02	0.049876	0.049876	0.049876	3.261280e-16	0.000000e+00	6.538826e-15	0
3	1.000000e+03	0.015807	0.015807	0.015807	1.804112e-15	0.000000e+00	1.141306e-13	0
4	1.000000e+04	0.005000	0.005000	0.005000	1.065120e-15	0.000000e+00	2.130294e-13	0
5	1.000000e+05	0.001581	0.001581	0.001581	1.015464e-15	2.168404e-19	6.422373e-13	
6	1.000000e+06	0.000500	0.000500	0.000500	4.627939e-14	0.000000e+00	9.255880e-11	0
7	1.000000e+07	0.000158	0.000158	0.000158	3.125901e-14	0.000000e+00	1.976994e-10	0
8	1.000000e+08	0.000050	0.000050	0.000050	6.808832e-13	6.776264e-21	1.361766e-08	
9	1.000000e+09	0.000016	0.000016	0.000016	2.470451e-12	0.000000e+00	1.562450e-07	0
10	1.000000e+10	0.000005	0.000005	0.000005	5.583154e-12	0.000000e+00	1.116631e-06	0