Terra Spal : Explicier, à l'aide d'un langage de programmation, les trajectoires astronomiques ou satellitaires pour tester les 2ème et 3ème lois de Kepler.

Attention des positions de Mars et tracé de sa trajectoire

```
In [1]: %load_ext matplotlib_inline
        %matplotlib inline
        import numpy as np, matplotlib.pyplot as plt, pandas as pd, datetime
```

```
In [2]: import math, sympy, sympy.plot_implicit, sympy.plotting.textplot
```

```
In [3]: from math import *
```

Obtention des données d'éphémérides :

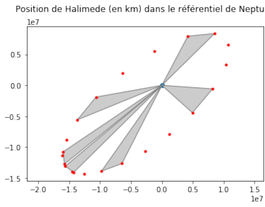
Lien de la NASA : https://ssd.jpl.nasa.gov/horizons.cgi#top

Horizons Web-Interface
Begin with "Target body"
Target body: ? help
Enter name, number or designation of the desired target body.
Enter ? for a list of options.
Enter ? for a list of options.
Enter ? for a list of options

```
        donnees = list(csv.DictReader(source, delimiter=','))
    dates = [donnee["Calendar Date (TDB)"][6:-14] for donnee in d
onnees]
    dates_Julien = [float(donnee["JDTDB"].replace(',','.')) for d
onnee in donnees]
    dt = dates_Julien[1] - dates_Julien[0]
    t = np.array([dt*i for i in range(len(dates))])
    x = np.array([float(donnee["X"].replace(',','.')) for donnee
in donnees])
    y = np.array([float(donnee["Y"].replace(',','.')) for donnee
in donnees])
    return dates, t, x, y
```

In [37]:
```
# pour tracer des polygones (pour nous des triangles)
from matplotlib.patches import Polygon
```

In [38]:
```
%matplotlib inline

donnees = "ephemerides/halimede-neptune.csv"

dates, t, x, y = jpl_horizons(donnees)

fig, ax = plt.subplots()

plt.scatter(0, 0)
plt.plot(x, y, 'r.')
plt.axis("equal")
plt.title(f"Position de Halimede (en km) dans le référentiel de N
eptune\n")

pas = 4 # 'pas' entre les zones où on trace un triangle et calcul
e son aire
for i in range(0, len(x), pas):
    xA, yA, xB, yB = x[i], y[i], x[i+1], y[i+1]
    triangle = Polygon([(0,0),(xA, yA),(xB, yB)], facecolor='0.8'
, edgecolor='0.5')
    ax.add_patch(triangle)

plt.show()

for i in range(0, len(x), pas):
    xA, yA, xB, yB = x[i], y[i], x[i+1], y[i+1]
    dateA, dateB = dates[i], dates[i+1]
    a, b = t[i], t[i+1]
    s = surface(xA, yA, xB, yB)
    print(f"Entre {dateA} et {dateA} (durée {b-a} j): aire balayé
e = {s:#.2g} km^2")
```

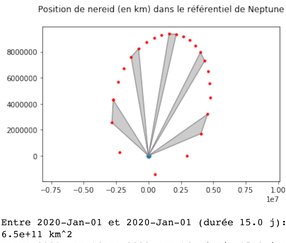



```
Entre 2020-Jan-01 et 2020-Jan-01 (durée 100.0 j): aire balayée =
4.2e+12 km^2
Entre 2021-Feb-04 et 2021-Feb-04 (durée 100.0 j): aire balayée =
4.2e+12 km^2
Entre 2022-Mar-11 et 2022-Mar-11 (durée 100.0 j): aire balayée =
4.1e+12 km^2
Entre 2023-Apr-15 et 2023-Apr-15 (durée 100.0 j): aire balayée =
4.2e+12 km^2
Entre 2024-May-19 et 2024-May-19 (durée 100.0 j): aire balayée =
4.2e+12 km^2
Entre 2025-Jun-23 et 2025-Jun-23 (durée 100.0 j): aire balayée =
4.3e+12 km^2
```

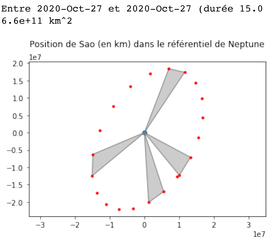
Script auto

In [39]:
```
def kepler_2(donnees, nom, pas):
    """ Reprend la démarche vue avec Halimède pour un satellites
dont l'éphéméride est dans le fichier donnees """

    dates, t, x, y = jpl_horizons(donnees)

    fig, ax = plt.subplots()

    plt.scatter(0, 0)
    plt.plot(x, y, 'r.')
    plt.axis("equal")
    plt.title(f"Position de {nom} (en km) dans le référentiel de
Neptune\n")

    for i in range(0, len(x), pas):
        xA, yA, xB, yB = x[i], y[i], x[i+1], y[i+1]
        triangle = Polygon([(0,0),(xA, yA),(xB, yB)], facecolor='
0.8', edgecolor='0.5')
        ax.add_patch(triangle)

    plt.show()

    for i in range(0, len(x), pas):
        xA, yA, xB, yB = x[i], y[i], x[i+1], y[i+1]
        dateA, dateB = dates[i], dates[i+1]
        a, b = t[i], t[i+1]
        s = surface(xA, yA, xB, yB)
        print(f"Entre {dateA} et {dateA} (durée {b-a} j): aire ba
layée = {s:#.2g} km^2")
```

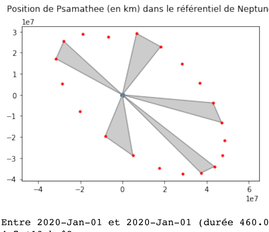
In [41]:
```
donnees = "ephemerides/nereid-neptune.csv"
nom = "nereid"
pas = 5
kepler_2(donnees, nom, pas)

donnees = "ephemerides/sao-neptune.csv"
nom = "Sao"
pas = 6
kepler_2(donnees, nom, pas)

donnees = "ephemerides/psamathee-neptune.csv"
nom = "Psamathee"
pas = 4
kepler_2(donnees, nom, pas)
```




```
Entre 2020-Jan-01 et 2020-Jan-01 (durée 15.0 j): aire balayée =
6.5e+11 km^2
Entre 2020-Mar-16 et 2020-Mar-16 (durée 15.0 j): aire balayée =
6.5e+11 km^2
Entre 2020-May-30 et 2020-May-30 (durée 15.0 j): aire balayée =
6.6e+11 km^2
Entre 2020-Aug-13 et 2020-Aug-13 (durée 15.0 j): aire balayée =
6.6e+11 km^2
Entre 2020-Oct-27 et 2020-Oct-27 (durée 15.0 j): aire balayée =
6.6e+11 km^2
```




```
Entre 2020-Jan-01 et 2020-Jan-01 (durée 145.0 j): aire balayée =
1.1e+13 km^2
Entre 2022-May-20 et 2022-May-20 (durée 145.0 j): aire balayée =
1.1e+13 km^2
Entre 2024-Oct-06 et 2024-Oct-06 (durée 145.0 j): aire balayée =
1.1e+13 km^2
Entre 2027-Feb-23 et 2027-Feb-23 (durée 145.0 j): aire balayée =
1.1e+13 km^2
```




```
Entre 2020-Jan-01 et 2020-Jan-01 (durée 460.0 j): aire balayée =
4.7e+13 km^2
Entre 2025-Jan-14 et 2025-Jan-14 (durée 460.0 j): aire balayée =
4.8e+13 km^2
Entre 2030-Jan-28 et 2030-Jan-28 (durée 460.0 j): aire balayée =
4.4e+13 km^2
Entre 2035-Feb-11 et 2035-Feb-11 (durée 460.0 j): aire balayée =
4.2e+13 km^2
Entre 2040-Feb-25 et 2040-Feb-25 (durée 460.0 j): aire balayée =
4.0e+13 km^2
```