W celu uzyskania wysokiej precyzji wykonywanych operacji na liczbach zmiennoprzecinkowych, używam biblioteki `decimal`.

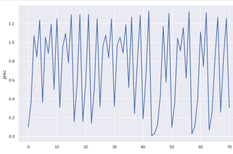
In [1]:
```
import decimal
from decimal import Decimal

import pandas as pd
import seaborn as sns
from matplotlib import pyplot as plt
```

Pierwsze k liczb ciagu

In [2]:
```
PRECISION = 64
k = 70

precise = []

with decimal.localcontext() as ctx:
    x_0 = Decimal('1') / Decimal('10')

    ctx.prec = PRECISION
    ctx.rounding = decimal.ROUND_HALF_EVEN

    x_k = x_0
    precise.append(float(x_0))

    for i in range(1, k+1):
        x_k = 4 * x_k - 1 * (1 - x_k)
        precise.append(float(x_k))
```

In [3]:
```
data = pd.DataFrame(columns=['prec'])
data['prec'] = precise
```

In [4]:
```
sns.set_theme()
plt.figure(figsize=(11,7))
sns.lineplot(data=data['prec'], linewidth=2)
plt.show()
```



Wyniki obliczeń w C++

float (4 bajty)

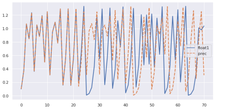
Mantysa: 23 bity

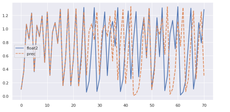
Wykładnik: 8 bitów

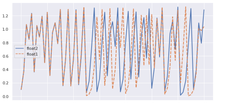
Dokładność: 7-8 cyfr znaczących

In [5]:
```
data['float1'] = [0.1, 0.37000000, 1.06929994, 0.84609287, 1.23578095, 0.36166012, 1.05424643, 0.88267910, 1.16
data['float2'] = [0.1, 0.37000000, 1.06929994, 0.84609273, 1.23578095, 0.36166000, 1.05424619, 0.88267970, 1.16
```

In [6]:
```
plt.figure(figsize=(10,5))
sns.lineplot(data=data[['float1', 'prec']], linewidth=2)
plt.show()
plt.figure(figsize=(10,5))
sns.lineplot(data=data[['float2', 'prec']], linewidth=2)
plt.show()
plt.figure(figsize=(10,5))
sns.lineplot(data=data[['float2', 'float1']], linewidth=2)
plt.show()
```







double (8 bajtów)

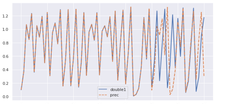
Mantysa: 53 bity

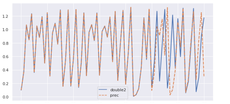
Wykładnik: 11 bitów

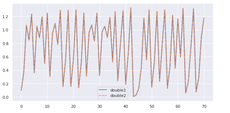
Dokładność: 15-16 cyfr znaczących

In [7]:
```
data['double1'] = [0.1, 0.3700000000000, 1.0693000000000, 0.8460925300000, 1.2357810237598, 0.3616596701
data['double2'] = [0.1, 0.3700000000000, 1.0693000000000, 0.8460925300000, 1.2357810237598, 0.3616596701
```

In [8]:
```
plt.figure(figsize=(10,5))
sns.lineplot(data=data[['double1', 'prec']], linewidth=2)
plt.show()
plt.figure(figsize=(10,5))
sns.lineplot(data=data[['double2', 'prec']], linewidth=2)
plt.show()
plt.figure(figsize=(10,5))
sns.lineplot(data=data[['double1', 'double2']], linewidth=2)
plt.show()
```







long double (16 bajtów)

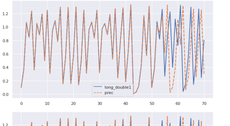
Mantysa: conajmniej 53 bity

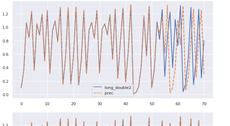
Wykładnik: conajmniej 11 bitów

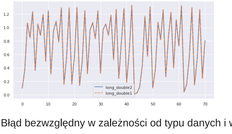
Dokładność: conajmniej 15-16 cyfr znaczących

In [9]:
```
data['long_double1'] = [0.1, 0.3700000000000, 1.0693000000000, 0.8460925300000, 1.2357810237597, 0.3616
data['long_double2'] = [0.1, 0.3700000000000, 1.0693000000000, 0.8460925300000, 1.2357810237597, 0.3616
```

In [10]:
```
plt.figure(figsize=(10,5))
sns.lineplot(data=data[['long_double1', 'prec']], linewidth=2)
plt.show()
plt.figure(figsize=(10,5))
sns.lineplot(data=data[['long_double2', 'prec']], linewidth=2)
plt.show()
plt.figure(figsize=(10,5))
sns.lineplot(data=data[['long_double2', 'long_double1']], linewidth=2)
plt.show()
```







Błąd bezwzględny w zależności od typu danych i wzoru

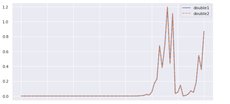
In [11]:
```
abs_err = data.sub(data['prec'], axis='rows').abs()
abs_err.drop(columns='prec', inplace=True)
```

In [12]:
```
plt.figure(figsize=(10,5))
sns.lineplot(data=abs_err[['float1', 'float2']], linewidth=2)
plt.show()
```

float:

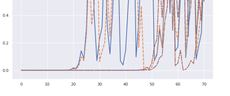
In [13]:
```
plt.figure(figsize=(10,5))
sns.lineplot(data=abs_err[['double1', 'double2']], linewidth=2)
plt.show()
```

double:



In [14]:
```
plt.figure(figsize=(10,5))
sns.lineplot(data=abs_err[['long_double1', 'long_double2']], linewidth=2)
plt.show()
```

long double:

In [15]:
```
plt.figure(figsize=(10,10))
sns.lineplot(data=abs_err, linewidth=2)
plt.show()
```

wszystkie typy