

```

In [1]: import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
import pandas as pd
STARTING_X=123
STARTING_Y=123

def rosenbrock(x, y):#zwraca wartość f. Rosenbrocka
    return (1 - x)**2 + 100 * (y - x**2)**2

    #gradient funkcji Rosembrocka
def gradient_rosenbrock(x, y):
    return np.array([-2 * (1 - x) - 400 * (y - x**2) * x, 200 * (y - x**2)])

# hesjan funkcji Rosenbrocka
def hessian_rosenbrock(x, y):
    return np.array([[2 - 400 * y + 1200 * x**2, -400 * x], [-400 * x, 200]])

    # alpha - współczynnik regulujący wpływ hessianu na aktualizację pozycji
    # max_iter - maksymalna liczba iteracji
    # eps - tolerancja błędu, po osiągnięciu której algorytm zatrzymuje iterację

def levenberg_marquardt(x0, y0, alpha=0.015, max_iter=1000, eps=1e-10):
    x, y = x0, y0
    path = []
    for i in range(max_iter):

        grad = gradient_rosenbrock(x, y)
        hessian = hessian_rosenbrock(x, y)
        update = np.linalg.inv(hessian + alpha * np.eye(2)) @ grad
        x -= update[0]
        y -= update[1]
        path.append((x, y))
        if np.linalg.norm(update) < eps:
            print(f"Znaleziono minimum po {i+1} iteracjach w punkcie ({x}, {y}) ")
            break
        else:
            print(f"Nie można osiągnąć wymaganej dokładności w {i+1} iteracjach ")
    return x, y, path

x_min, y_min, path = levenberg_marquardt(STARTING_X, STARTING_Y)

#print(f"Minimum znajduje się w punkcie ({x_min:.6f}, {y_min:.6f})")

# od tego punktu w kodzie już tylko rysuję wykres.

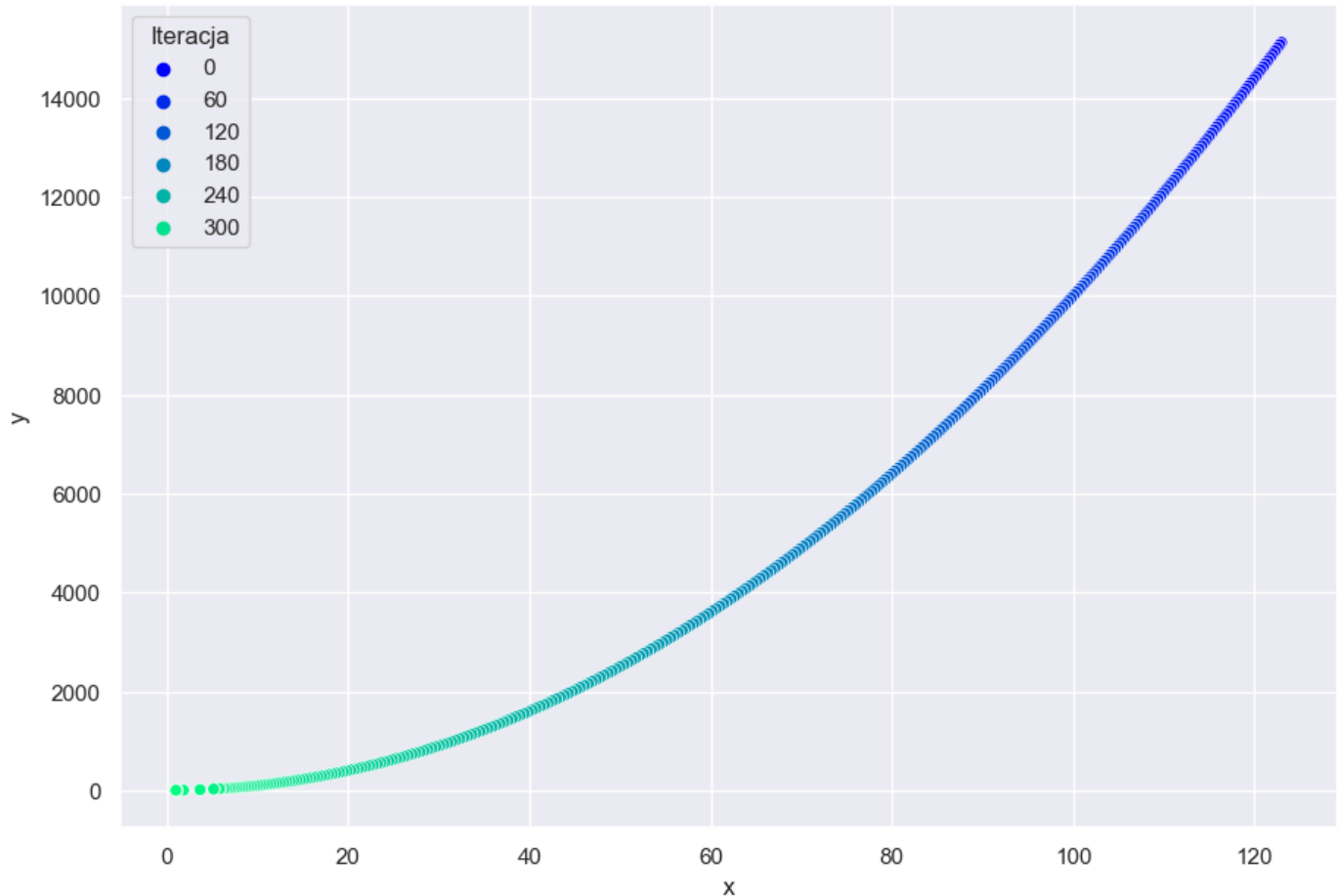
#print(path)
sns.set(rc={'figure.figsize':(10.7,7.27)})
sns.set_theme()
path = np.array(path)
#plt.plot(path[:, 0], path[:, 1], "bo-")
plt.show()

path_df = pd.DataFrame(path, columns=['x', 'y'])
path_df['iter'] = path_df.index
sns.scatterplot(data=path_df, x='x', y='y', hue='iter', palette="winter")

```

```
plt.legend(title='Iteracja')
plt.show()
```

Znaleziono minimum po 343 iteracjach w punkcie (1.00000000000003324, 1.0000000000000666)



Algorytm Levenberga-Marquardta (LMA) możemy stosować do szukania minimum funkcji dwóch zmiennych.

- LMA znajduje jedynie minimum lokalne, niekoniecznie jest ono minimum globalnym; jest tak w przypadku f. Rosenbrocka.
- LMA używa aproksymacji Hesjanu do określenia kierunku, w którym mają się zmieniać x i y, aby zbliżyć się do minimum.
- Algorytm ten łączy w sobie cechy metody największego spadku i metody Gaussa-Newtona.
- W porównaniu z metodą gradientu prostego, metoda Levenberga-Marquardta zwykle daje lepsze rezultaty przy mniejszej liczbie iteracji.
- Na wykresie znajduje się ścieżka algorytmu Levenberga-Marquardta po każdej iteracji.

In []: