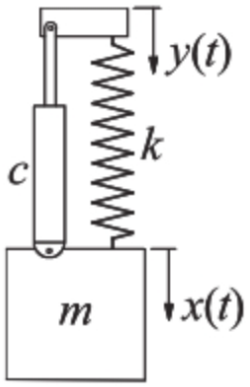
Agnieszka Cych, 252906, Fizyka II rok

SPRAWOZDANIE Z WYKONANEGO ZADANIA

Spis treści:

1. TREŚĆ ZADANIA (Zad. 21):



Rozważ wymuszone drgania układu sprężyna-masa-tłumik pokazanego na rysunku. Gdy górna część układu podlega harmonicznemu przemieszczeniu danemu wzorem $y(t)=Y\sin\omega t$, odpowiedź w postaci przemieszczenia masy opisuje równanie $x(t)=X\sin(\omega t-\beta)$, gdzie

$$\frac{X}{Y}=\sqrt{(1+Z\cos\phi)^2+(Z\sin\phi)^2} \qquad \tan\beta=\frac{Z\sin\phi}{1+Z\cos\phi} \quad .$$

Notacja użyta w powyższych wyrażeniach:

$$Z=\frac{\left(\frac{\omega}{p}\right)^2}{\sqrt{\left[1-\left(\frac{\omega}{p}\right)^2\right]^2+\left(2\zeta\frac{\omega}{p}\right)^2}} \qquad \tan\phi=\frac{2\zeta\frac{\omega}{p}}{1+\left(\frac{\omega}{p}\right)^2}$$

$p=\sqrt{\frac{k}{m}}$ - częstość drgań własnych nietłumionych

$\zeta=\frac{c}{2mp}$ - współczynnik tłumienia

Wiedząc, że m = 0.2 kg, k = 2 880 N/m, a ω = 96 rad/s, wyznacz najmniejsze c (rzeczywisty współczynnik tłumienia [kg/s]), dla którego stosunek X/Y nie przekracza wartości 1.5.

2. METODA:

Do wykonania obliczeń stworzono program w języku Python wraz z zaimportowaną biblioteką math.

W pierwszym kroku zdefiniowano funkcję ‘calculate_x_over_y’ liczącą stosunek X do Y w zależności od zadanego c, ω, m i p. Do zbudowania tej funckji wykorzystano wzory na X/Y, tan β, Z i tan Φ podane w zadaniu. Funkcja ta zwaraca obliczoną wartość X/Y, czyli zmienną x_over_y.

Mając już zdefiniowaną funkcję ‘calulate_x_over_y’ przystąpiono do definicji funkcji ‘main’. Wprowadzono dane podane w zadaniu: masę, współczynnik k i częstość kołową ω. Na tej podstawie obliczono częstość drgań własnych nietłumionych p. Wiedząc, że ζ przyjmuje wartości od 0 do 1, obliczono maksymalny rzeczywisty współczynnik tłumienia c, przyjmując ζ=1 i wyświetlono tę wartość. Jednocześnie za c podstawiono właśnie tę maksymalną wartość. Założono także, że X/Y wynosi 2, aby pętla w kolejnym kroku rozpoczęła swoje działanie.

Po zdefiniowaniu danych i dokonaniu koniecznych założeń ($c=c_{max}$, X/Y=2) rozpoczęto pętlę. Jeśli wartość X/Y była różna od 1.5 podstawiano nowe c i ponownie liczono X/Y. Wartości c zmieniały się w zależności od otrzymanego w poprzednim kroku wyniku – jeśli X/Y było większe od 1.5, to do wartości c dodawano $\frac{c_{max}}{2^{i+1}}$, gdzie i+1 – numer kroku iteracyjnego. W innym przypadku odejmowano od c tę wartość. Jest to tzw. metoda bisekcji. Granice badanego obszaru wyznaczyła od dołu wartość 0, a od góry wartość c_{max}. Ponadto wykorzystano zależność, że wraz ze wzorstem c maleje stosunek X/Y.

Z założenia pętla była iterowana 100 razy, tak by osiągnąć dokładność $\frac{c_{max}}{2^{100}}$, co w przypadku $c_{max}=48$, dawało wartość: $3{,}8\cdot10^{-29}$. Uznano to za wystarczające przybliżenie. Pętlę wyposażono w przerywnik w postaci osiągniętego wyniku X/Y=1.5. Jeśli bowiem została odnaleziona wartość c, dla której X/Y wynosi dokładnie wartość graniczną, to jako najmniejsze c spełniajace równość X/Y<1.5, wystarczy przyjąć $c_{gr}+\varepsilon$, gdzie ε – najmniejsza wartość większa od 0. W przypadku przerwania pętli z wyżej opisanego powodu, program podaje wartość graniczną c_{gr}, krok iteracyjny, w którym tę wartość otrzymano i dokładność wyznaczonej wielkości. (Dodanie funkcji wyświetlenia kroku iteracyjnego nie jest konieczne dla osiągnięcia wyniku. Jest to jedynie podyktowane ciekawością, jak szybko program poprawny wynik odnajduje, by ocenić przydatność metody bisekcji.) Jeśli pętla kończy się dopiero wraz z wyczerpaniem zadanej jej ilości iteracji, użytkownikowi wyświetlany jest komunikat z obliczoną wartością c wraz z osiągniętą dokładnością (wspomniane $\frac{c_{max}}{2^{100}}$).

Na samym końcu programu umieszczono konstrukcję if __name__ == "__main__". Jest to w Pythonie powszechną praktyką, która ma na celu przede wszystkim zapewnić, że pewne fragmenty kodu zostaną wykonane tylko wtedy, gdy plik zostanie uruchomiony jako program, a nie zaimportowany jako moduł w innym skrypcie. Jest to też wskazówka dla osoby korzystającej z kodu, że to jest moment, od którego program się wykonuje.

Ostatecznie, wywołano funkcję main().

3. PROGRAM:

```python
import math

def calculate_x_over_y(c, omega, m, p):
    zeta = c / 2 * m * p
    Z = (omega/p) ** 2 / math.sqrt((1 - (omega/p) ** 2) ** 2 + (2* zeta * omega /p)  ** 2)

    theta = math.atan((2 * zeta * omega / p) / (1 - (omega / p) ** 2))

    x_over_y = math.sqrt((1 + Z * math.cos(theta)) ** 2 + (Z * math.sin(theta)) ** 2)
    return x_over_y

def main():
    # Dane
    m = 0.2
    k = 2880
    omega = 96
    p = math.sqrt(k/m)
    c = 2*m*p #zaczynamy od c max i będziemy "szli w dół"
    print("Maksymalna wartosc, jaka moze przyjac c to", c)
    x_over_y = 2 # Bo się wykrzaczy, jak nie pójdzie pętla



    for i in range(100):
        if x_over_y!=1.5: #zeby przerwało liczenie, gdy osiągnie wynik X/Y=1.5
            x_over_y = calculate_x_over_y(c, omega, m, p)
            if x_over_y < 1.5:
                c -= (2*m*p/(2**i)) #metoda bisekcji
            else: c += (2*m*p/(2**i))
        else:
            print("Graniczna wartosc c, dla ktorego X/Y wynosi 1.5 to c_gr =", c)
            print("Najmniejsze c spelniajace wymogi zadania jest o 'epsilon' wieksze od c granicznego.")
            print("Krok iteracyjny, w ktorym osiagnieto ten wynik:", i+1)
            print("Dokladnosc wyniku wynosi:", (2*m*p/(2**(i+1))))
            break
    if x_over_y!= 1.5:
        print("c =", c)
        print("dokladnosc wyznaczonego c wynosi: ", (2*m*p/(2**100)))
        print("X/Y = ", x_over_y)

#Dobra praktyka, niekonieczne
if __name__ == "__main__":
    main()
```

4. WYNIKI:

Dla podanych w zadaniu danych: m = 0.2 kg, k = 2 880 N/m i ω = 96 rad/s, otrzymano wynik postaci:

```
Maksymalna wartosc, jaka moze przyjac c to 48.0
Graniczna wartosc c, dla ktorego X/Y wynosi 1.5 to c_gr = 0.03920875398286575
Najmniejsze c spelniajace wymogi zadania jest o 'epsilon' wieksze od c granicznego.
Krok iteracyjny, w ktorym osiagnieto ten wynik: 63
Dokladnosc wyniku wynosi: 1.0408340855860843e-17
```

Z powyższego wynika, że program dla c = 0.03920875398286575 osiągnął wartość X/Y = 1.5 i zakończył pętlę po 63 iteracji. Poszukiwana wartość c wynosii zatem:

$$[0{,}039208753982865750 + \varepsilon \mp 0{,}000000000000000011]\ \text{kg/s}\ .$$

Metoda bisekcji sprawdziła się w tym zadaniu, gdyż relatywnie szybko odnaleziono poszukiwany wynik.