Національний технічний університет України «КПІ ім. Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

Лабораторна робота № 4

з дисципліни «Спеціальні розділи математики-2.  
Чисельні методи»

Виконав:

студент гр. ІС-34

Колосов Ігор

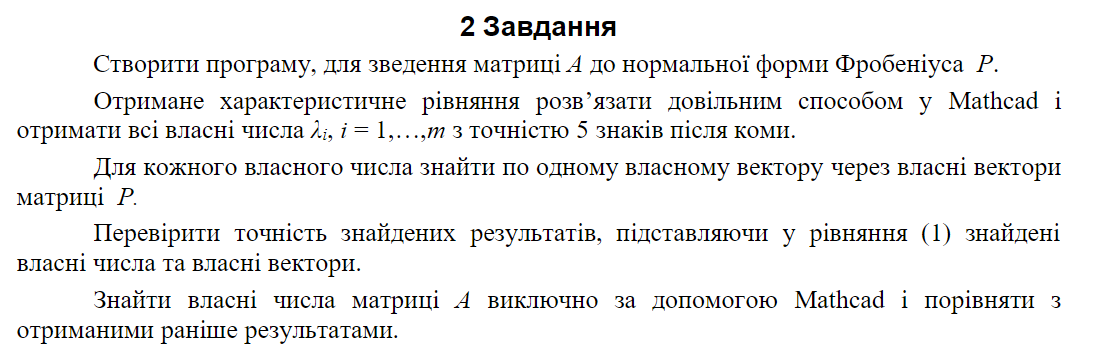
Викладач:

доц. Рибачук Л.В.

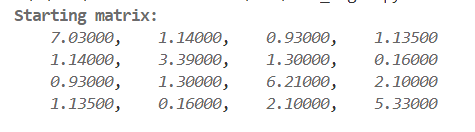
Київ – 2024

**Зміст**

1. **Постановка задачі**

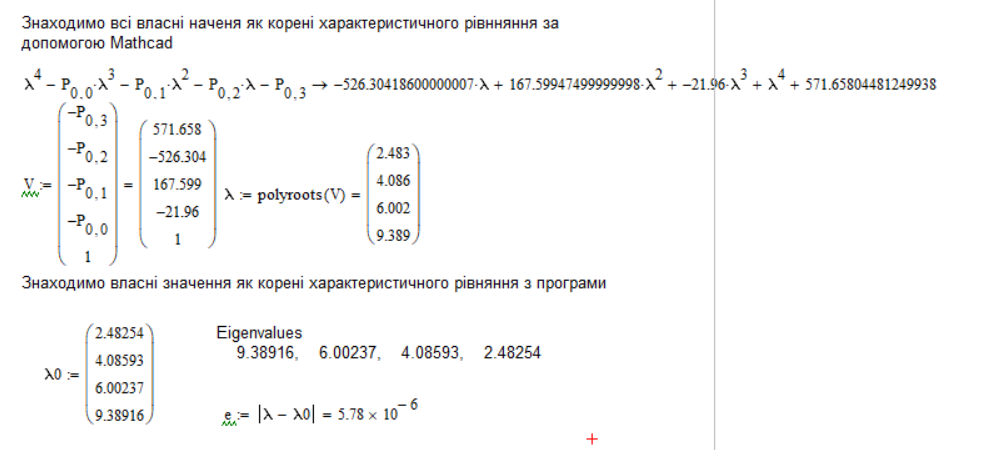
****

1. **Проміжні матриці M^-1 та M\_i, результуюча матриця P у нормальній формі фронебіуса**

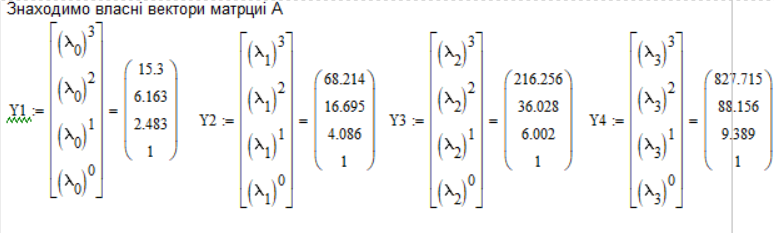
****

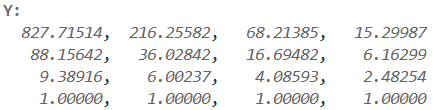
|  |  |
| --- | --- |

1. **Отримане характеристичне рівняння та власні числа**

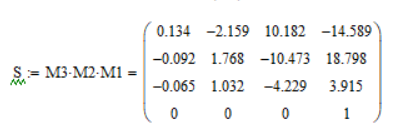
****

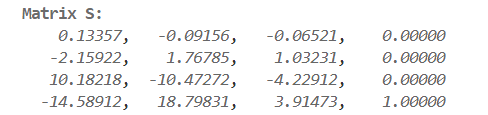
1. **Власні числа - корені характеристичного рівняння**

****

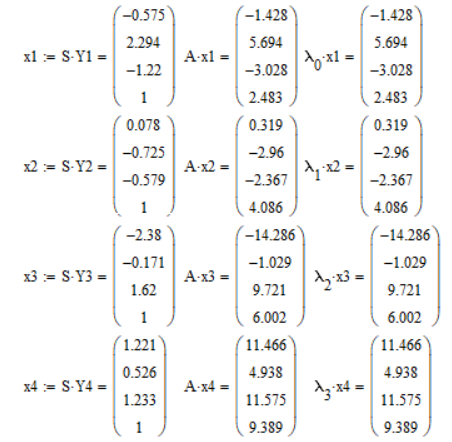
****

1. **Матриця подібності S**

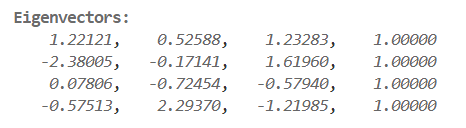
****

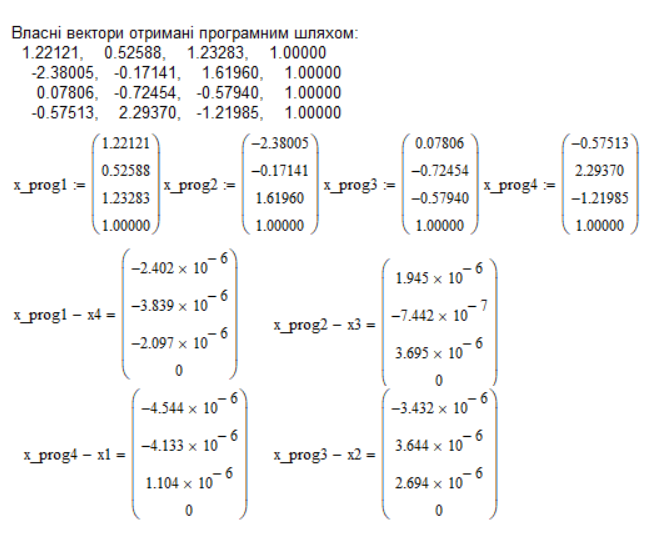
****

1. **Оцінка точності обчислень**

****

1. **Порівняння власного рішення з рішенням Mathcad**

****

****

1. **Лістинг програми**

| import numpy as np  from lib\_print import \*  np.set\_printoptions(suppress=True)  \_\_default\_print\_format = ANSI.FG.BRIGHT\_BLACK + ANSI.Styles.ITALIC  \_\_default\_numbers\_format = "10.5f"  \_\_default\_text\_format = ANSI.Styles.BOLD  def default\_matrix():  t = 7.0  k = 3 \* (3 - 4) + 1  a = 0.11 \* t  b = 0.02 \* k  g = 0.02 \* k  d = 0.015 \* t  return np.array([  [6.26 + a, 1.10 - b, 0.97 + g, 1.24 - d],  [1.10 - b, 4.16 - a, 1.30, 0.16],  [0.97 + g, 1.30, 5.44 + a, 2.1],  [1.24 - d, 0.16, 2.10, 6.10 - a]  ]), 4    a, m = default\_matrix()  true\_eigenvalues, true\_eigenvectors = np.linalg.eig(a)  # підставити значення з маткаду  print(printer(  a,  f"{\_\_default\_text\_format} Starting matrix: ",default\_style= \_\_default\_print\_format, formatting=\_\_default\_numbers\_format  ) + "\n")  """  1. Порахувати нормальну форму фронебіуса (скрипт)  \* Показати M\_i^-1, M\_i, результуючу матрицю P.  Порахувати характеристичне рівняння методом данилевського (скрипт)  3. Знайти власні числа характеристичного рівняння (маткад) -> (numpy)  4. Знайти власнтй вектор для кожного власного числа (скрипт)  5. Перевірити точність підстановкою у рівняння знайдені числа та вектори.  A \* x = lambda \* x  """  # Нормальна форма фронебіуса  mat\_a\_i = a.copy()  s = None  for i in range(1, m):  idx = m - i  mat\_m = np.eye(m)    row = [  -mat\_a\_i[idx,j]/mat\_a\_i[idx,idx-1] # що тут вбіса відбувається  if j != m - i - 1 else 1 / mat\_a\_i[idx, idx-1]  for j in range(m)]  mat\_m[idx-1] = row  mat\_a\_i = np.linalg.inv(mat\_m) @ mat\_a\_i @ mat\_m    if s is None:  s = mat\_m.copy()  else: s = s @ mat\_m  print(printer(  mat\_m,  f"{\_\_default\_text\_format} Matrix M{idx}", default\_style= \_\_default\_print\_format, formatting=\_\_default\_numbers\_format  ) + "\n")  print(printer(  mat\_a\_i,  f"{\_\_default\_text\_format} Matrix A{i}",default\_style= \_\_default\_print\_format, formatting=\_\_default\_numbers\_format  ) + "\n")  p = mat\_a\_i.copy()  print(printer(  p, f"{\_\_default\_text\_format} Matrix P, phronebius form: ", default\_style= \_\_default\_print\_format, formatting=\_\_default\_numbers\_format  ) + "\n")  print(printer(  s, f"{\_\_default\_text\_format} Matrix S: ", default\_style= \_\_default\_print\_format, formatting=\_\_default\_numbers\_format  ) + "\n")  # що таке форма фронебіуса  # що таке eigenvalues (власні значення)  v = np.array([-p[0,3], -p[0,2], -p[0,1], -p[0,0], 1])  lambda\_values = np.roots(v[::-1]) # eigenvals  # що робить метод roots,  # як знаходяться власні значення з нормальної форми фронебіуса  print(printer(  lambda\_values.reshape(-1, 1)[:4], f"{\_\_default\_text\_format} Eigenvalues ", default\_style= \_\_default\_print\_format, formatting=\_\_default\_numbers\_format,  ) + "\n")  # y  mat\_y = np.array([  [l\*\*3, l\*\*2, l, 1] for l in lambda\_values  ])  # чому значення для y рахуються як eigenvalues у якійсь степені?  print(printer(  mat\_y, f"{\_\_default\_text\_format} Y: ", default\_style= \_\_default\_print\_format, formatting=\_\_default\_numbers\_format,  ) + "\n")  # що таке eigenvectors (власні вектори)  # eigenvec  eigenvec = s @ mat\_y.T  print(printer(  eigenvec, f"{\_\_default\_text\_format} Eigenvectors: ", default\_style= \_\_default\_print\_format, formatting=\_\_default\_numbers\_format,  ) + "\n")  np\_eigenvalues, np\_eigenvectors = np.linalg.eig(a)  print(printer(  np\_eigenvalues.reshape(-1, 1)[:4], f"{\_\_default\_text\_format} (numpy) Eigenvalues ", default\_style= \_\_default\_print\_format, formatting=\_\_default\_numbers\_format,  ) + "\n")  # чому eigenvalues через numpy у іншому порядку?  errors = []  for i, (lambda\_i, v) in enumerate(zip(lambda\_values, eigenvec.T), 1):  errors.append(np.linalg.norm(a @ v - lambda\_i \* v))  print(printer(  np.array(errors).reshape(-1, 1)[:4], f"{\_\_default\_text\_format} (numpy) eigenvectorc - (own) eigenvectors ", default\_style= \_\_default\_print\_format, formatting=\_\_default\_numbers\_format,  ) + "\n") |
| --- |