

▼ NAMA LENGKAP:

NIM:

Suatu persamaan fungsi rotasi kritis bintang memenuhi bentuk persamaan fungsi

$$f(x) = x - \frac{\omega^2 r_p^3}{2GM} x^3 - 1$$

dengan $x = r/r_p$, dan r adalah jejari bintang arah ekuator, sedangkan r_p adalah jejari bintang arah kutub (indeks p untuk *pole*). Konstanta G sebagaimana diketahui bersama adalah sebesar $6,673 \times 10^{-11} \text{ m}^3 \text{ kg}^{-1} \text{ s}^{-2}$ (dalam *coding* dapat ditulis: 6.673E-11). Untuk modul ini, massa bintang adalah serupa Matahari, $M = M_\odot = 1,989 \times 10^{30} \text{ kg}$ (dalam *coding* dapat ditulis: 1.989E30), jejari arah kutub $r_p = 6,95 \times 10^8 \text{ m} \neq R_\odot$ (dalam *coding* dapat ditulis: 6.95E8), dan kecepatan sudut $\omega = 2,424068 \times 10^{-4} \text{ rad s}^{-1}$ (dalam *coding* dapat ditulis: 2.424068E-4) yang setara dengan periode rotasi 0,3 hari.

Akar persamaan fungsi rotasi kritis bintang tersebut akan dipecahkan, yaitu untuk:

$$f(x) = x - \frac{\omega^2 r_p^3}{2GM} x^3 - 1 = 0$$

Jika dimisalkan solusinya adalah $x = x_c = r_c/r_p$, hal ini menyatakan situasi perbandingan jejari arah ekuator sebesar r_c terhadap jejari arah kutub, r_p , di saat kritis menjelang pecahnya kesetimbangan gaya-gaya yang bekerja.

▼ Definisi fungsi

```
1 def f(x):  
2     G, M, rp, w = ....., ....., ....., .....  
3     return x - .....
```

▼ Memuat modul-modul python yang diperlukan

```
1 import scipy.optimize as sco  
2 import numpy as np  
3 import matplotlib.pyplot as plt
```

▼ Menyiapkan domain x dan y untuk fungsi menghitung dan plot $f(x)$

Sebagai contoh:

```
1 xleft2plot, xright2plot, pointsNumber = .., .., ...
```

```
2 x = np.linspace(xleft2plot, xright2plot, pointsNumber)
3 y = f(x)
```

▼ Plot esensial untuk $f(x)$

```
1 plt.xlabel('x')
2 plt.ylabel('y', rotation='horizontal')
3 plt.grid()
4
5 plt.plot(x,y)
```

▼ Mari kita baca sebentar dokumentasi `bisect` (untuk metode *bisection*) di `sco` (`scipy.optimize`)

Perhatikan pada penggunaan `bisect` (dan fungsi yang biasa dijelaskan pada dokumentasi modul python), sebagian argumen fungsi sudah diberi nilai baku (*default*; bertanda `=`) walaupun dalam bentuk *list* / *tuples* kosong.

Karena itu, misalnya `maxiter=100` sebagai nilai baku, kita dapat mengubah, misalnya, `maxiter=70` bila diperlukan. Demikian pula argumen bernilai baku yang lain.

Sebagai metode pengurung, perhatikan dua titik pengurung. Tipe data keduanya dipastikan masing-masing adalah **skalar**.

```
1 help(sco.bisect)
```

▼ TUGAS 1

Lakukan dan tunjukkan contoh eksperimen numerik yang Anda lakukan di bawah ini dengan menggunakan metode *bisection*:

```
1 r1 = sco.bisect(..., ..., ..., ..., full_output=False) # or full_output=True
2 r1
```

▼ Mari kita baca sebentar dokumentasi `newton` (untuk metode *Newton-Raphson*) di `sco` (`scipy.optimize`)

Perhatikan pada penggunaan `newton`, sebagian argumen fungsi juga sudah diberi nilai baku (*default*; bertanda `=`) walaupun dalam bentuk *list* / *tuples* kosong.

Karena itu, misalnya `tol=1.48e-08` sebagai nilai baku untuk toleransi galat, kita dapat mengubah, misalnya, `tol=1e-07` bila diperlukan. Demikian pula argumen bernilai baku yang lain.

Mengingat metode ini bukan metode pengurung, titik(-titik) dugaan awal lokasi akar mungkin dapat diberikan sebagai **skalar** atau sebagai **list**. Silakan cek di dokumentasi untuk tipe data `x0` yang diizinkan

```
1 help(sco.newton)
```

▼ TUGAS 2

Lakukan dan tunjukkan contoh eksperimen numerik yang Anda lakukan di bawah ini dengan menggunakan metode Newton-Raphson:

```
1 r = sco.newton(..., ..., ..., full_output=True) # or full_output=False
2 r
```

▼ Dokumentasi lain yang dapat dibaca di lain waktu misalnya `sco.fsolve` atau dokumentasi umum untuk `sco`

```
1 help(sco.fsolve)
```

▼ PERTANYAAN

Setelah solusi untuk x ditemukan,

1. solusi manakah yang tidak mungkin terjadi untuk bintang? Apa alasannya?
2. solusi manakah yang lebih mirip dengan keadaan Matahari saat ini?
3. solusi manakah yang paling ekstrim? Jejari arah ekuator $r = \dots$?

JAWAB (DI SINI):

1. Solusi yang tidak mungkin:
2. Solusi yang mirip dengan keadaan Matahari saat ini:
3. Solusi yang paling ekstrim:

© 2022, Muhamad Irfan Hakim

Hanya untuk kepentingan di dalam Prodi Astronomi FMIPA - ITB

--- END OF FILE ---