

Praktikum - Kestimbangan

Kode Python ([kesetimbangan.py](#)) yang pertama dibuat oleh Sandy digunakan untuk membuat stream plot dengan persamaan diferensial:

$$\frac{dx}{dt} = -x^2 + 3x - 2$$

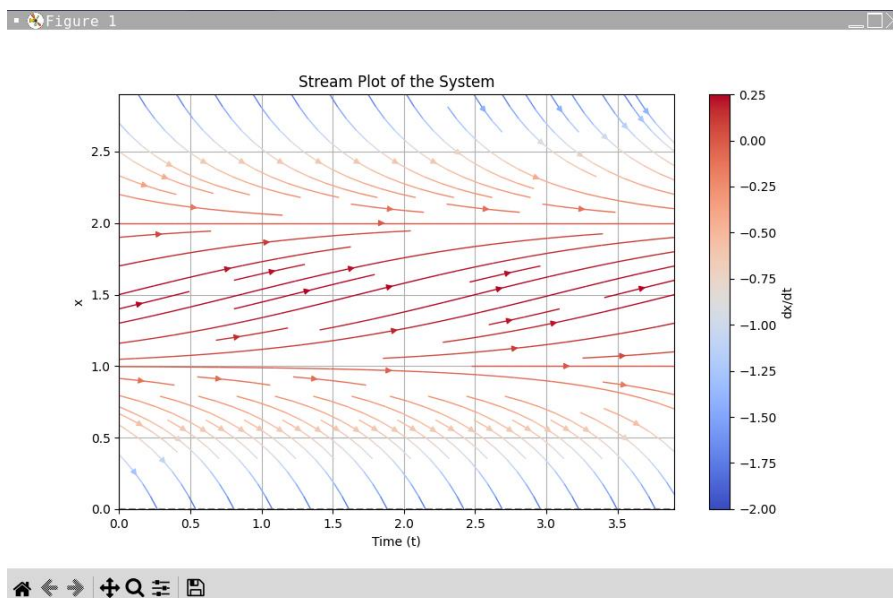
Tujuan dari stream plot yang dihasilkan untuk mengetahui kestimbangan persamaan tersebut (dimana $\frac{dx}{dt} = 0$) dan juga arah dari aliran (stream/flow).

Tipe/sifat solusi setimbang Persamaan Diferensial Tak Linier ditentukan sebagai berikut:

1. Jika $f'(x^*) \neq 0$ maka x^* disebut solusi setimbang hiperbolik.
Jika $f'(x^*) \neq 0$ maka dapat terjadi bertanda positif atau bertanda negatif:
 - a. Jika $f'(x^*) < 0$ maka x^* merupakan solusi setimbang stabil (asimtotik)
 - b. Jika $f'(x^*) > 0$ maka x^* merupakan solusi setimbang tidak stabil.
2. Jika $f'(x^*) = 0$ maka x^* merupakan titik saddle.

Catatan Presentasi “Kestimbangan” FLEARN

Output yang dihasilkan dalam menjalankan file kesetimbangan.py milik Sandy:



Data tersebut memvisualisasikan karakteristik pada persamaan diferensial tersebut. Setiap arah panah menunjukkan arah dari x setiap t berjalan. Warna yang semakin merah menunjukkan perubahan $\frac{dx}{dt}$ semakin tinggi dan sebaliknya jika warna semakin biru.

Penjelasan kode pertama kesetimbangan.py Arsandy:

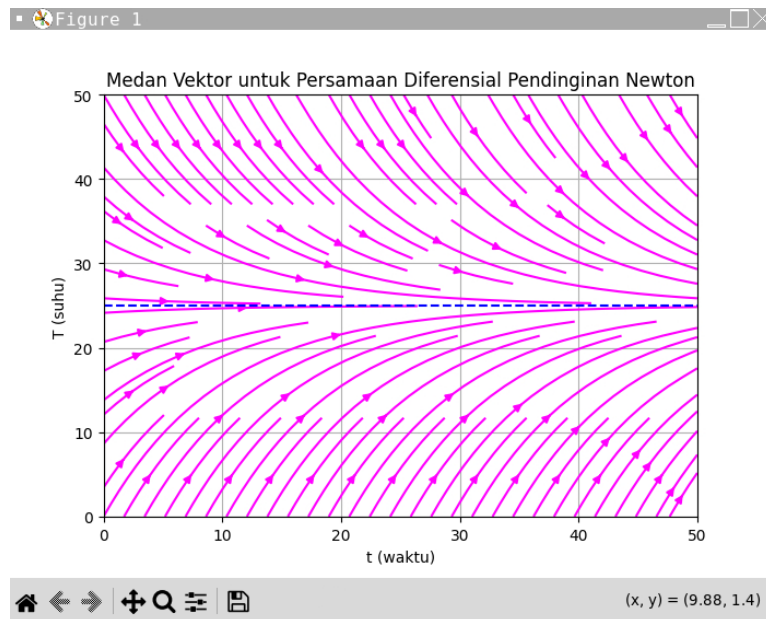
1. Mendefinisikan range untuk x dan t:
 - `x = np.arange(0, 3, 0.1)` membuat array/larik dengan jangka 0 hingga 3 dengan perbedaan 0,1.
 - `t = np.arange(0, 4, 0.1)` membuat array/larik dengan jangka 0 hingga 4 dengan perbedaan 0,1.
2. Membuat meshgrid:
 - `T, X = np.meshgrid(t,x)` membuat meshgrid 2D dari nilai x dan t.
3. Mendefinisikan arah persamaan diferensial
 - `dX` terdefinisi jelas sesuai yang dicari
 - `dT = np.ones(dX.shape)` membuat array/larik berisi 1 dengan bentuk "matriks" yang sama dengan `dX`.
4. Membuat streamplot
 - `plt.streamplot(T,X,dT,dX, color=dX, cmap='coolwarm', linewidth=1)` adalah setting untuk membuat grid (T,X) dengan `dT` dan `dX` sebagai komponen vektor untuk menunjukkan kesetimbangan.

Kode Python yang kedua digunakan untuk membuat stream plot dengan persamaan diferensial Hukum Pendinginan Newton yaitu:

$$\frac{dT}{dt} = -k(T - T_a)$$

Tujuan dari stream plot yang dihasilkan adalah untuk mengetahui akan terjadi kesetimbangan dimana $T = T_a$ atau suhu benda akan menjadi sama dengan suhu ruangan/sekitarnya.

Output yang dihasilkan dalam menjalankan kode bagian kedua dari file kesetimbangan.py milik Sandy:



Sama dengan kode sebelumnya yaitu untuk memvisualisaikan karakteristik persamaan diferensial hukum pendinginan Newton. Terdapat garis biru yang menunjukkan ekuilibrium atau setimbang pada saat suhu sama dengan T_a yang ditentukan.

Penjelasan kode kedua kesetimbangan.py Arsandy:

1. Mendefinisikan Konstan dan Sistem yang memuat persamaan:
 - $k = 0.1$ adalah konstan pendinginan yang ditentukan
 - $T_a = 25$ adalah suhu ruangan/sekitar
 - Mendefinisikan fungsi `system(T)` yang memuat persamaan.
2. Mendefinisikan range untuk t (waktu) dan T (suhu):
 - `t_vals = np.linspace(0, 50, 20)` dan `T_vals = np.linspace(0, 50, 20)` membuat array/larik dengan jarak yang sama sejumlah 20 mulai dari 0 hingga 50.
3. Membuat meshgrid:
 - `t, T = np.meshgrid(t_vals, T_vals)` membuat meshgrid 2D dari nilai t dan T .
4. Mendefinisikan arah persamaan diferensial
 - `U = np.ones_like(T)` akan digunakan untuk menunjukkan arah horizontal (waktu) dengan bentuk larik berisi 1 yang seperti T . Sama dengan sebelumnya berisi 1 karena waktu sifatnya seragam.
 - `V = system(T)` akan digunakan untuk menunjukkan arah vertikal sesuai dengan persamaan diferensial yang ditentukan
5. Membuat streamplot

- `plt.streamplot(t,T,U,V, color=magenta)` adalah setting untuk membuat grid (t,T) dengan U dan V sebagai komponen vektor untuk menunjukkan arah kesetimbangan.