

# SPIRAL OPTIMIZATION ALGORITHM

Tugas Kuliah  
SK5001 Analisis Numerik Lanjut

Mohammad Rizka Fadhli  
NIM: 20921004

16 October 2021

## PENDAHULUAN

### Bahasa yang Digunakan

Saya membuat program *spiral optimization algorithm* menggunakan bahasa **R** yang bisa dieksekusi pada versi minimal 3.5.3.

### Spiral Optimization Algorithm

*Spiral Optimization Algorithm* adalah salah satu metode *meta heuristic* yang digunakan untuk mencari minimum global dari suatu sistem persamaan.

Algoritmanya mudah dipahami dan intuitif tanpa harus memiliki latar keilmuan tertentu. Proses kerjanya adalah dengan melakukan *random number generating* pada suatu selang dan melakukan rotasi sekaligus kontraksi dengan titik paling minimum pada setiap iterasi sebagai pusatnya.

Berikut adalah algoritmanya:

#### INPUT

```
m >= 2 # jumlah titik
theta # sudut rotasi (0 <= theta <= 2pi)
r      # kontraksi
k_max  # iterasi maksimum
```

#### PROCESS

```
1 generate m buah titik secara acak
  x_i

2 initial condition
  k = 0 # untuk keperluan iterasi

3 cari x_* yang memenuhi
  min(f(x_*))

4 lakukan rotasi dan kontraksi semua x_i
  x_* sebagai pusat rotasi
  k = k + 1

5 ulangi proses 3 dan 4
```

```
6 hentikan proses saat k = k_max
  output x_*
```

Berdasarkan algoritma di atas, salah satu proses yang penting adalah melakukan **rotasi** dan **konstraksi** terhadap semua titik yang telah di-*generate*.

Agar memudahkan, saya akan memberikan ilustrasi geometri beserta operasi matriks aljabar terkait kedua hal tersebut.

## Penjelasan Geometri

### Operasi Matriks Rotasi

Misalkan saya memiliki titik  $x \in \mathbb{R}^2$ . Untuk melakukan rotasi sebesar  $\theta$ , saya bisa menggunakan suatu matriks  $A_{2 \times 2}$  berisi fungsi-fungsi trigonometri sebagai berikut:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}$$

Berdasarkan operasi matriks di atas, saya membuat **program** di **R** dengan beberapa modifikasi. Sebagai contoh, saya akan membuat program yang bertujuan untuk melakukan rotasi suatu titik  $x \in \mathbb{R}$  sebanyak  $n$  kali:

```
# mendefinisikan program
rotasi_kan = function(x0,rot){
  # menghitung theta
  theta = 2*pi/rot

  # definisi matriks rotasi
  A = matrix(c(cos(theta),-sin(theta),
               sin(theta),cos(theta)),
             ncol = 2,byrow = T)

  # membuat template
  temp = vector("list")
  temp[[1]] = x0

  # proses rotasi
  for(i in 2:rot){
    xk = A %*% x0
    temp[[i]] = xk
    x0 = xk
  }

  # membuat template data frame
  final = data.frame(x = rep(NA,rot),
                    y = rep(NA,rot))

  # gabung data dari list
  for(i in 1:rot){
    tempura = temp[[i]]
    final$x[i] = tempura[1]
    final$y[i] = tempura[2]
  }

  # membuat plot
```

```

plot =
  ggplot() +
    geom_point(aes(x,y),data = final) +
    geom_point(aes(x[1],y[1]),
               data = final,
               color = "red") +
    coord_equal() +
    labs(title = "titik merah adalah titik initial")

# enrich dengan garis panah
panah = data.frame(
  x_start = final$x[1:(rot-1)],
  x_end = final$x[2:rot],
  y_start = final$y[1:(rot-1)],
  y_end = final$y[2:rot]
)

# menambahkan garis panah ke plot
plot =
  plot +
    geom_segment(aes(x = x_start,
                     xend = x_end,
                     y = y_start,
                     yend = y_end),
                 data = panah,
                 arrow = arrow(length = unit(.3,"cm")))
)

# menyiapkan output
list("Grafik rotasi" = plot,
     "Titik-titik rotasi" = final)
}

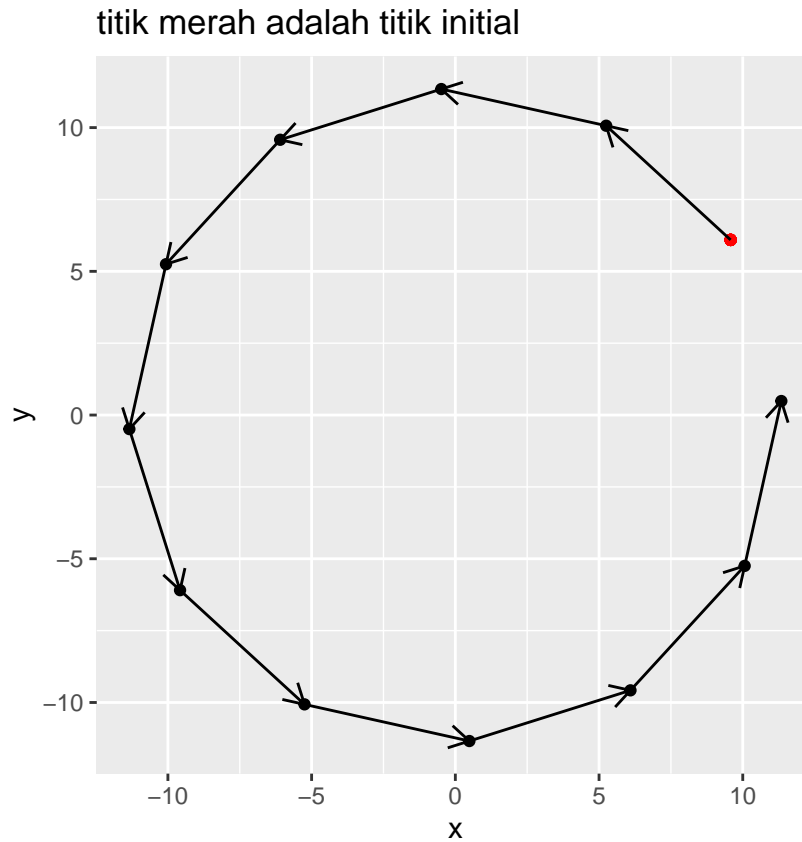
```

Berikut adalah uji coba dengan titik sembarang berikut ini:

```
# uji coba
rot = 12 # berapa banyak rotasi
x0 = rand_titik(0,10) # generate random titik

rotasi_kan(x0,rot)
```

```
## $`Grafik rotasi`
```



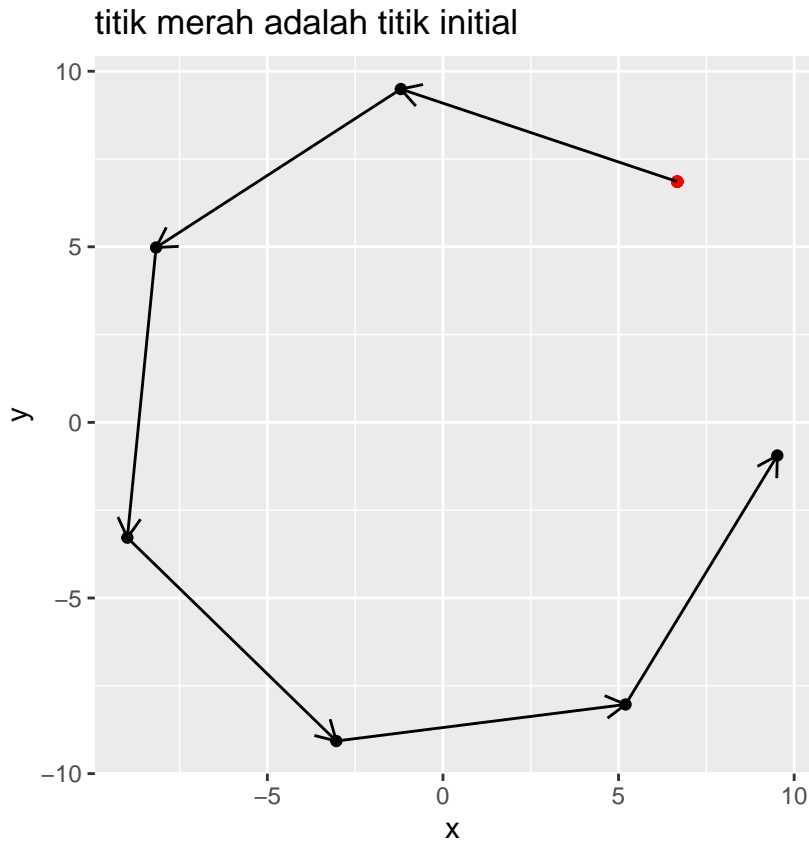
```
##
## $`Titik-titik rotasi`
##      x      y
## 1    9.578  6.093
## 2    5.248 10.066
## 3   -0.488 11.341
## 4   -6.093  9.578
## 5  -10.066  5.248
## 6  -11.341 -0.488
## 7   -9.578 -6.093
## 8   -5.248 -10.066
## 9    0.488 -11.341
## 10   6.093 -9.578
## 11  10.066 -5.248
## 12  11.341  0.488
```

Uji coba kembali dengan titik sembarang lainnya berikut ini:

```
# uji coba
rot = 7 # berapa banyak rotasi
x0 = rand_titik(0,10) # generate random titik

rotasi_kan(x0,rot)
```

```
## $`Grafik rotasi`
```



```
##
## $`Titik-titik rotasi`
##      x      y
## 1  6.673  6.8564
## 2 -1.200  9.4920
## 3 -8.169  4.9800
## 4 -8.987 -3.2821
## 5 -3.037 -9.0727
## 6  5.200 -8.0313
## 7  9.521 -0.9422
```

## Operasi Matriks Rotasi dan Konstraksi

Jika pada sebelumnya saya **hanya melakukan rotasi**, kali ini saya akan memodifikasi operasi matriks agar melakukan rotasi dan konstraksi secara bersamaan. Untuk melakukan hal tersebut, saya akan definisikan  $r, 0 < r < 1$  dan melakukan operasi matriks sebagai berikut:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} r \\ r \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}$$

Oleh karena itu saya akan modifikasi program **R** sebelumnya menjadi sebagai berikut:

```
# mendefinisikan program
rotasi_konstraksi_kan = function(x0,rot,r){
  # menghitung theta
  theta = 2*pi/rot

  # definisi matriks rotasi
  A = matrix(c(cos(theta),-sin(theta),
               sin(theta),cos(theta)),
             ncol = 2,byrow = T)

  # membuat template
  temp = vector("list")
  temp[[1]] = x0

  # proses rotasi dan konstraksi
  for(i in 2:rot){
    xk = A %*% x0
    xk = r * xk
    temp[[i]] = xk
    x0 = xk
  }

  # membuat template data frame
  final = data.frame(x = rep(NA,rot),
                    y = rep(NA,rot))

  # gabung data dari list
  for(i in 1:rot){
    tempura = temp[[i]]
    final$x[i] = tempura[1]
    final$y[i] = tempura[2]
  }

  # membuat plot
  plot =
    ggplot() +
    geom_point(aes(x,y),data = final) +
    geom_point(aes(x[1],y[1]),
              data = final,
              color = "red") +
    coord_equal() +
    labs(title = "titik merah adalah titik initial")

  # enrich dengan garis panah
```

```

panah = data.frame(
  x_start = final$x[1:(rot-1)],
  x_end = final$x[2:rot],
  y_start = final$y[1:(rot-1)],
  y_end = final$y[2:rot]
)
# menambahkan garis panah ke plot
plot =
  plot +
  geom_segment(aes(x = x_start,
                   xend = x_end,
                   y = y_start,
                   yend = y_end),
               data = panah,
               arrow = arrow(length = unit(.3,"cm")))
)

# menyiapkan output
list("Grafik rotasi" = plot,
     "Titik-titik rotasi" = final)
}

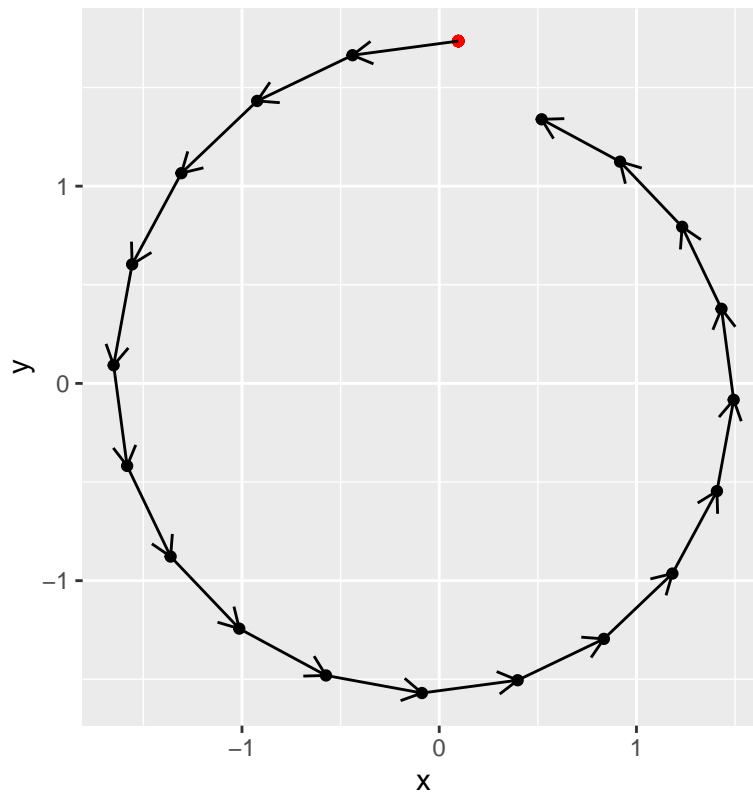
```

Saya akan uji coba untuk sembarang titik berikut ini:

```
# uji coba
rot = 20 # berapa banyak rotasi
x0 = rand_titik(0,4) # generate random titik
r = .99
rotasi_konstraksi_kan(x0,rot,r)
```

```
## $`Grafik rotasi`
```

titik merah adalah titik initial



```
##
## $`Titik-titik rotasi`
##      x      y
## 1  0.09691  1.73560
## 2 -0.43972  1.66379
## 3 -0.92302  1.43201
## 4 -1.30715  1.06593
## 5 -1.55684  0.60373
## 6 -1.65053  0.09216
## 7 -1.58225 -0.41817
## 8 -1.36183 -0.87778
## 9 -1.01369 -1.24309
## 10 -0.57414 -1.48054
## 11 -0.08764 -1.56964
## 12  0.39768 -1.50470
## 13  0.83476 -1.29509
## 14  1.18217 -0.96401
```



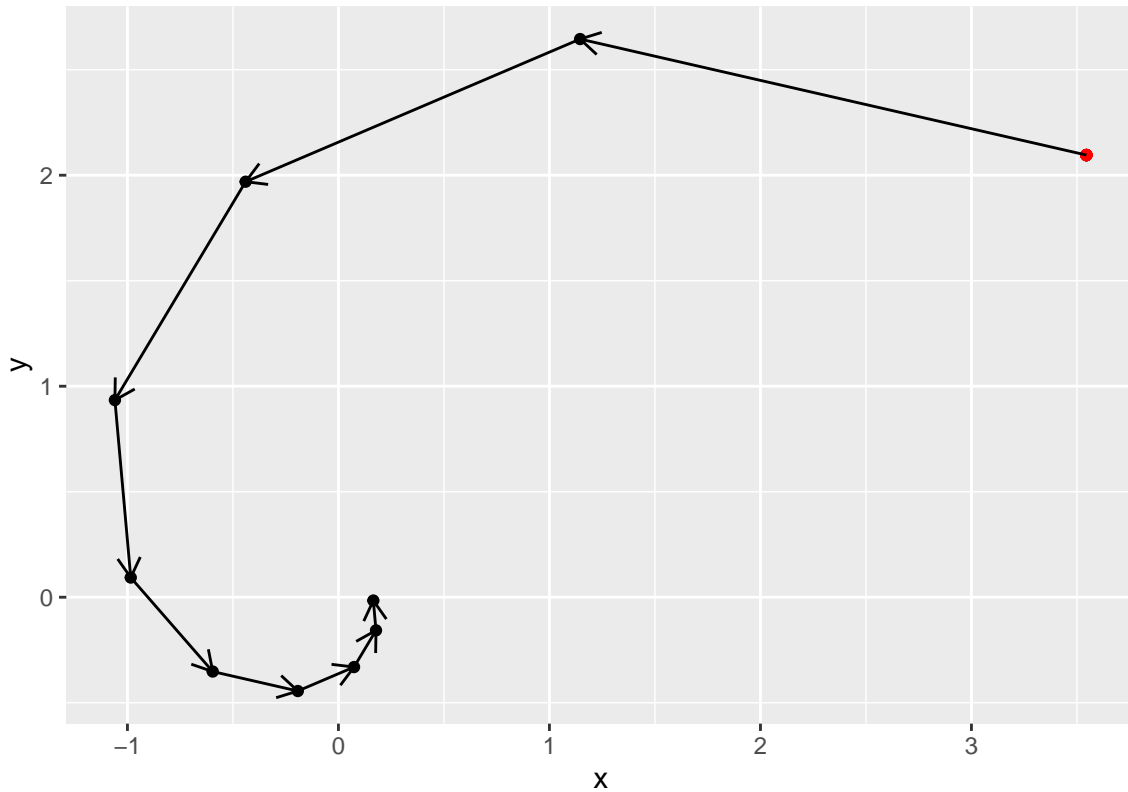
##	15	1.40798	-0.54600
##	16	1.49271	-0.08335
##	17	1.43096	0.37819
##	18	1.23161	0.79385
##	19	0.91676	1.12423
##	20	0.51924	1.33897

Saya akan uji coba kembali untuk sembarang titik lainnya berikut ini:

```
# uji coba
rot = 10 # berapa banyak rotasi
x0 = rand_titik(0,4) # generate random titik
r = .7
rotasi_konstraksi_kan(x0,rot,r)
```

```
## $`Grafik rotasi`
```

titik merah adalah titik initial



```
##
## $`Titik-titik rotasi`
##      x      y
## 1  3.54479  2.09560
## 2  1.14522  2.64526
## 3 -0.43984  1.96925
## 4 -1.05933  0.93424
## 5 -0.98430  0.09321
## 6 -0.59577 -0.35221
## 7 -0.19248 -0.44459
## 8  0.07392 -0.33097
## 9  0.17804 -0.15702
## 10 0.16543 -0.01567
```

## Program *Spiral Optimization Algorithm*

Berbekal program yang telah dituliskan di bagian sebelumnya, kita akan sempurnakan program untuk melakukan *spiral optimization* sebagai berikut:

baru sampai sini....

## SOAL 1

Tentukanlah akar-akar sistem persamaan berikut dengan **SOA**. Buatlah terlebih dahulu *contour plot*-nya:

$$f_1(x_1, x_2) = \cos(2x_1) - \cos(2x_2) - 0.4 = 0$$

$$f_2(x_1, x_2) = 2(x_2 - x_1) + \sin(x_2) - \sin(x_1) - 1.2 = 0$$

dengan  $-10 \leq x_1, x_2 \leq 10$

## JAWAB

*Contour Plot*

## SOAL 2

Tentukanlah akar-akar sistem persamaan berikut dengan **SOA**. Buatlah terlebih dahulu *contour plot*-nya:

$$f_1(x_1, x_2) = \sin(x_1) \cos(x_2) + 2 \cos(x_1) \sin(x_2) = 0$$

$$f_2(x_1, x_2) = \cos(x_1) \sin(x_2) + 2 \sin(x_1) \cos(x_2) = 0$$

dengan  $0 \leq x_1, x_2 \leq 2\pi$