

# 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)
}

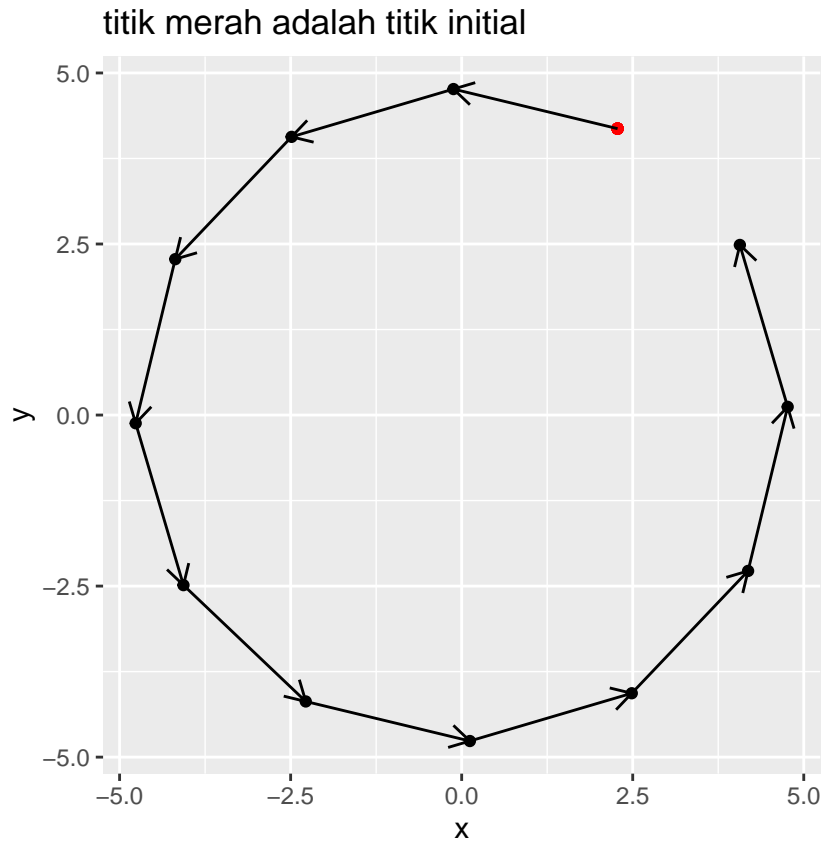
```

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`
```



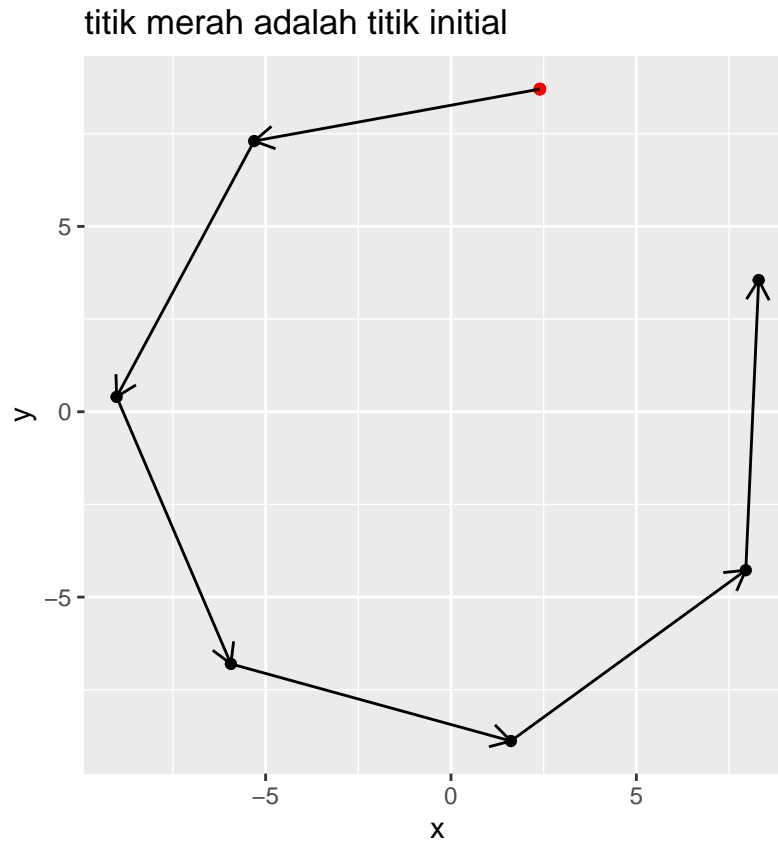
```
##
## $`Titik-titik rotasi`
##      x      y
## 1  2.2786 4.1864
## 2 -0.1199 4.7648
## 3 -2.4862 4.0665
## 4 -4.1864 2.2786
## 5 -4.7648 -0.1199
## 6 -4.0665 -2.4862
## 7 -2.2786 -4.1864
## 8  0.1199 -4.7648
## 9  2.4862 -4.0665
## 10 4.1864 -2.2786
## 11 4.7648 0.1199
## 12 4.0665 2.4862
```

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  2.398  8.7039
## 2 -5.310  7.3016
## 3 -9.019  0.4011
## 4 -5.937 -6.8015
## 5  1.616 -8.8824
## 6  7.952 -4.2747
## 7  8.300  3.5519
```

## Operasi Matriks Rotasi dan Kontraksi

Jika pada sebelumnya saya **hanya melakukan rotasi**, kali ini saya akan memodifikasi operasi matriks agar melakukan rotasi dan kontraksi 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 kontraksi
  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)
}

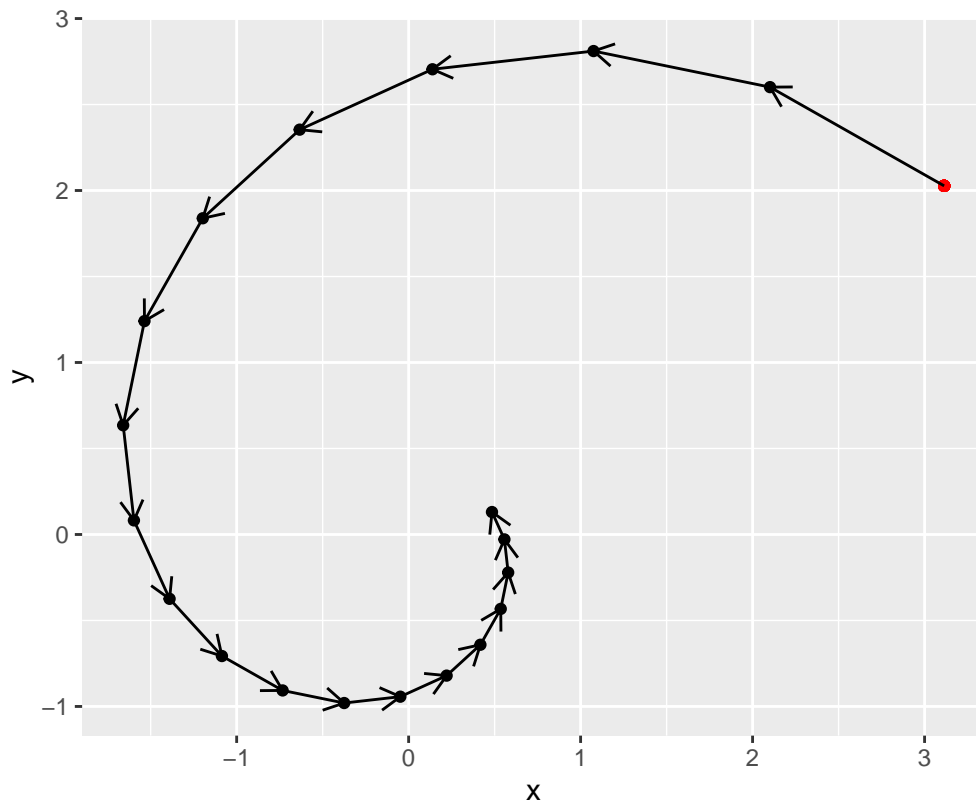
```

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 = .9
rotasi_konstraksi_kan(x0,rot,r)
```

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

titik merah adalah titik initial



```
##
## $`Titik-titik rotasi`
##      x      y
## 1  3.11374 2.02718
## 2  2.10142 2.60114
## 3  1.07529 2.81089
## 4  0.13865 2.70503
## 5 -0.63364 2.35394
## 6 -1.19703 1.83863
## 7 -1.53595 1.24087
## 8 -1.65980 0.63495
## 9 -1.59730 0.08187
## 10 -1.38998 -0.37416
## 11 -1.08569 -0.70683
## 12 -0.73272 -0.90696
## 13 -0.37493 -0.98010
## 14 -0.04834 -0.94319
```

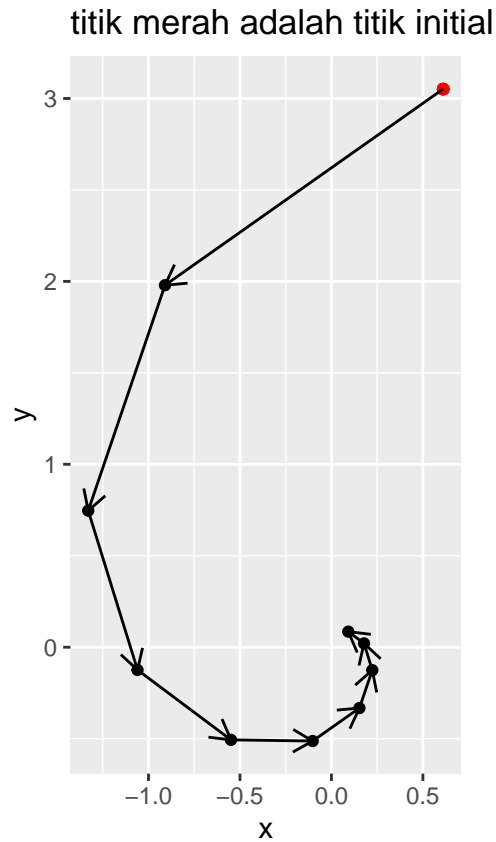


```
## 15  0.22094 -0.82077
## 16  0.41738 -0.64109
## 17  0.53555 -0.43266
## 18  0.57874 -0.22139
## 19  0.55694 -0.02855
## 20  0.48465  0.13046
```

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-titik rotasi`
##      x      y
## 1  0.61039  3.05144
## 2 -0.90985  1.97921
## 3 -1.32960  0.74650
## 4 -1.06012 -0.12432
## 5 -0.54921 -0.50659
## 6 -0.10259 -0.51286
## 7  0.15292 -0.33265
## 8  0.22347 -0.12546
## 9  0.17817  0.02089
## 10 0.09231  0.08514
```

## Operasi Matriks Rotasi dan Konstraksi dengan Titik $y$ Sebagai Pusatnya

Salah satu prinsip utama dari *spiral optimization algorithm* adalah menjadikan titik  $x^*$  sebagai pusat rotasi di setiap iterasinya. Oleh karena itu kita akan modifikasi program bagian sebelumnya menjadi sebagai berikut:

```
# mendefinisikan program
rotasi_konstraksi_pusat_kan = function(x0,rot,r,x_bin){
  # 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-x_bin) # diputar dengan x_bin sebagai pusat
    xk = x_bin + (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() +
    geom_point(aes(x = x_bin[1],
                  y = x_bin[1]),
              color = "blue") +
    labs(title = "titik merah adalah titik initial\ntitik biru adalah pusat rotasi")

  # 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")))
  )

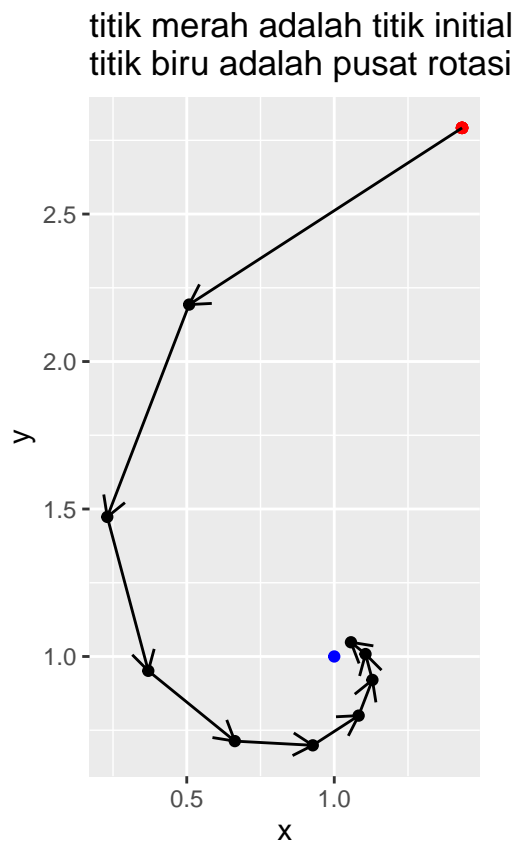
  # menyiapkan output
  list("Grafik rotasi" = plot,
       "Titik-titik rotasi" = final)
}

```

Saya akan coba dengan sembarang titik berikut:

```
# uji coba
rot = 10 # berapa banyak rotasi
x0 = rand_titik(0,4) # generate random titik
x_bintang = c(1,1) # contoh pusat rotasi
r = .7
rotasi_konstraksi_pusat_kan(x0,rot,r,x_bintang)
```

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



```
##
## $`Titik-titik rotasi`
##      x      y
## 1  1.4333 2.7921
## 2  0.5080 2.1931
## 3  0.2305 1.4733
## 4  0.3695 0.9514
## 5  0.6629 0.7130
## 6  0.9272 0.6988
## 7  1.0827 0.7995
## 8  1.1293 0.9205
## 9  1.1060 1.0082
## 10 1.0567 1.0482
```

## Program *Spiral Optimization Algorithm*

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

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