

SPIRAL OPTIMIZATION ALGORITHM

Tugas Kuliah
SK5001 Analisis Numerik Lanjut

Mohammad Rizka Fadhli
NIM: 20921004

17 October 2021

PENDAHULUAN

Bahasa yang Digunakan

Untuk membuat program *spiral optimization algorithm*, saya 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      # konstraksi
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 konstraksi 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.

Membuat Program

Untuk menyelesaikan tugas soal yang diberikan, pertama-tama saya harus membuat program *spiral optimization algorithm*. Untuk membuatnya, saya akan melakukannya perlahan-lahan dengan bantuan ilustrasi geometri. Berikut adalah langkah-langkah yang ditempuh:

1. **Pertama** saya akan membuat program yang bisa merotasi suatu titik berdasarkan suatu θ tertentu.
2. **Kedua** saya akan memodifikasi program tersebut untuk melakukan rotasi sekaligus konstraksi dengan rasio r tertentu.
3. **Ketiga** saya akan memodifikasi program tersebut untuk melakukan rotasi sekaligus konstraksi dengan **titik pusat rotasi tertentu**.

Dari program yang terakhir, akan saya pakai untuk **membangun program spiral optimization algorithm** yang sebenarnya.

Langkah dan Ilustrasi Geometri

Operasi Matriks Rotasi

Misalkan saya memiliki titik $x \in \mathbb{R}^2$. Untuk melakukan rotasi sebesar θ , 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)
}

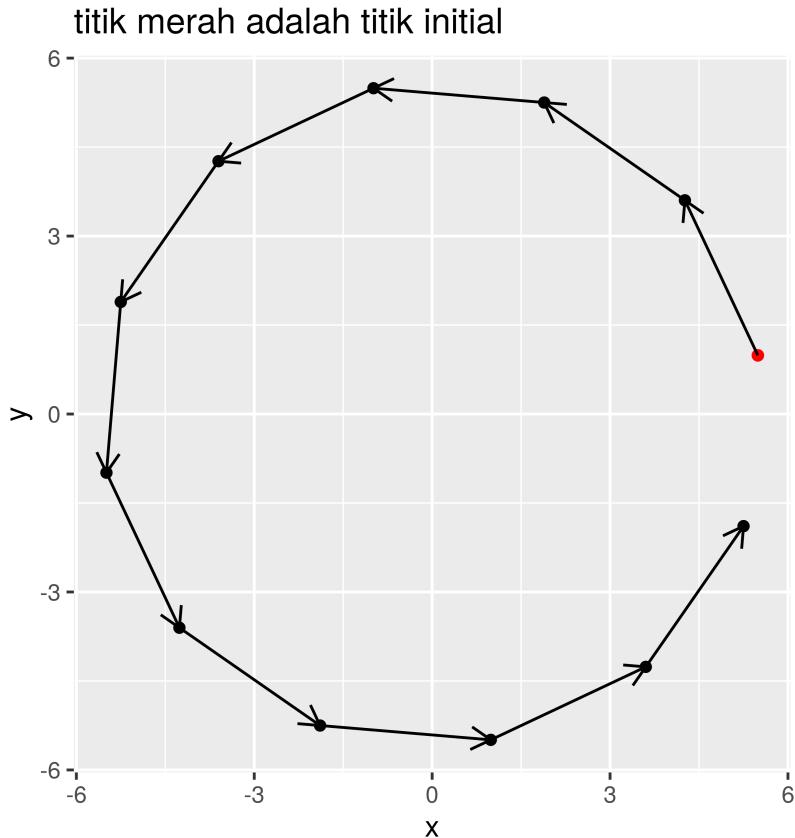
```

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  5.492510  0.988952  

## 2  4.262177  3.602713  

## 3  1.889797  5.251130  

## 4 -0.988952  5.492510  

## 5 -3.602713  4.262177  

## 6 -5.251130  1.889797  

## 7 -5.492510 -0.988952  

## 8 -4.262177 -3.602713  

## 9 -1.889797 -5.251130  

## 10 0.988952 -5.492510  

## 11 3.602713 -4.262177  

## 12 5.251130 -1.889797
```

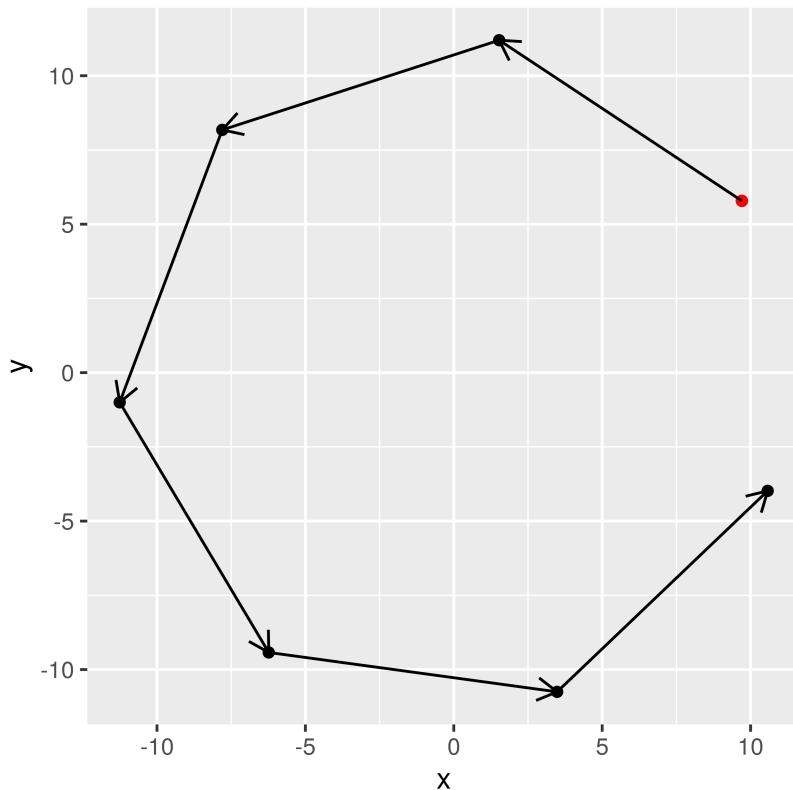
Uji coba kembali dengan titik sembarang lainnya berikut ini:

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

rotasi_kan(x0,rot)
```

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

titik merah adalah titik initial



```
##
## $`Titik-titik rotasi`  
##          x      y  
## 1   9.70694  5.78567  
## 2   1.52876 11.19650  
## 3  -7.80061  8.17613  
## 4 -11.25596 -1.00102  
## 5  -6.23534 -9.42439  
## 6   3.48061 -10.75100  
## 7  10.57560 -3.98189
```

Operasi Matriks Rotasi dan Kontraksi

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)
}

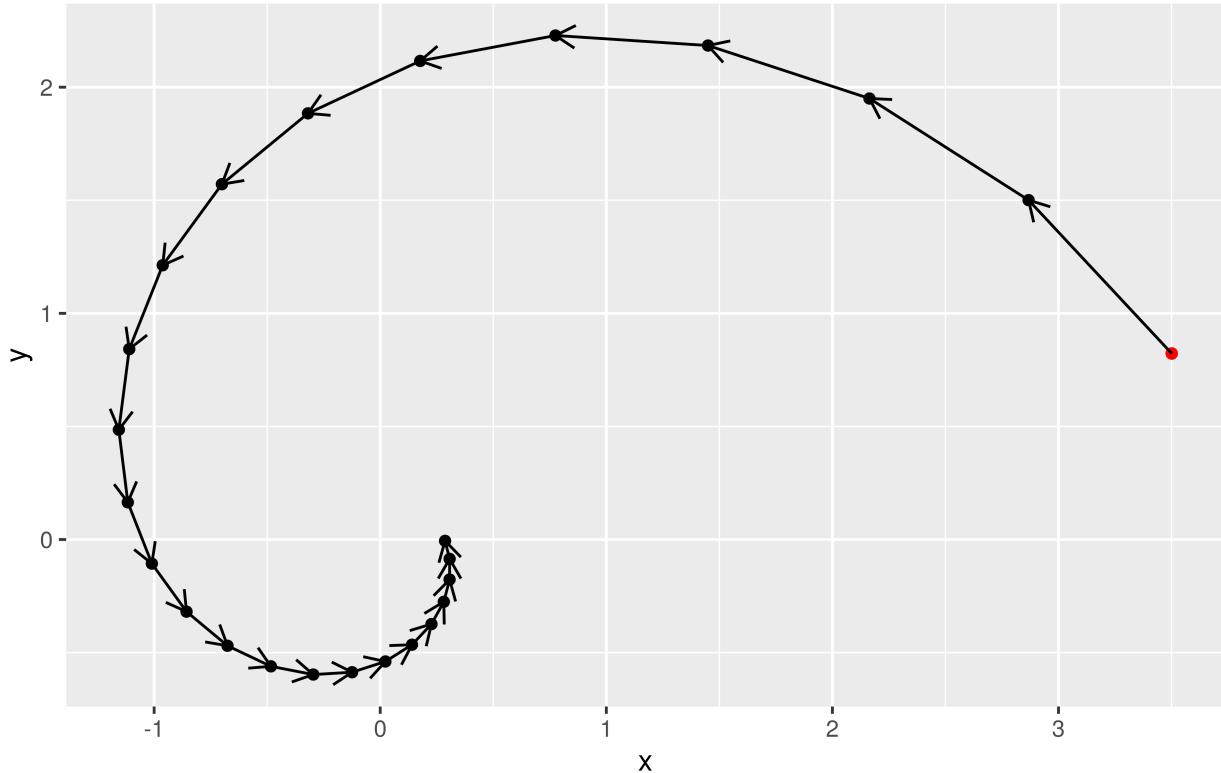
```

Saya akan uji coba untuk sembarang titik berikut ini:

```
# uji coba
rot = 25 # 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  3.5008444  0.82253871  
## 2  2.8676717  1.50058958  
## 3  2.1639574  1.94994608  
## 4  1.4499367  2.18415533  
## 5  0.7750863  2.22850860  
## 6  0.1768752  2.11612682  
## 7  -0.3194469 1.88426868  
## 8  -0.7002085 1.57106492  
## 9  -0.9620263 1.21281502  
## 10 -1.1100756 0.84191938  
## 11 -1.1561196 0.48546392  
## 12 -1.1164752 0.16442722  
## 13 -1.0100614 -0.10655518  
## 14 -0.8566464 -0.31895965
```

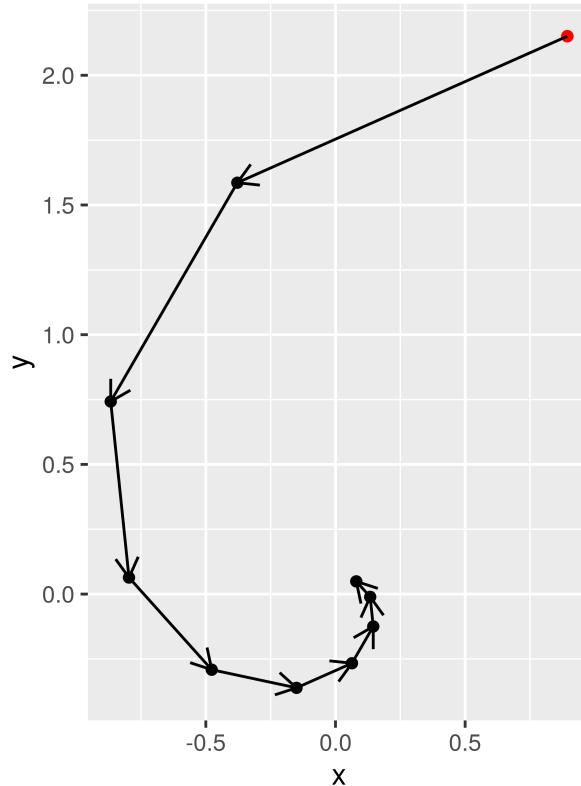
```
## 15 -0.6753701 -0.46978041
## 16 -0.4835902 -0.56068119
## 17 -0.2960654 -0.59699732
## 18 -0.1244671 -0.58668302
## 19  0.0228109 -0.53928451
## 20  0.1405879 -0.46500216
## 21  0.2266312 -0.37388742
## 22  0.2812439 -0.27520215
## 23  0.3067633 -0.17695230
## 24  0.3070188 -0.08559369
## 25  0.2867935 -0.00589693
```

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  0.8941988  2.1506390  
## 2  -0.3784843  1.5858503  
## 3  -0.8668378  0.7423586  
## 4  -0.7963437  0.0637464  
## 5  -0.4772074 -0.2915550  
## 6  -0.1502880 -0.3614579  
## 7   0.0636119 -0.2665339  
## 8   0.1456894 -0.1247682  
## 9   0.1338415 -0.0107139  
## 10  0.0802042  0.0490017
```

Catatan penting:

Terlihat bahwa semakin banyak rotasi dan konstraksi yang dilakukan akan membuat titik *initial* menuju **pusat** (0, 0).

Operasi Matriks Rotasi dan Kontraksi dengan Titik x^* Sebagai Pusatnya

Salah satu prinsip utama dari *spiral optimization algorithm* adalah menjadikan titik x^* sebagai pusat rotasi di setiap iterasinya. Operasi matriksnya adalah sebagai berikut:

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

Oleh karena itu kita akan modifikasi program bagian sebelumnya menjadi seperti ini:

```
# mendefinisikan program
rotasi_konstraksi_pusat_kan = function(x0,rot,r,x_bin){
  # pusat rotasi
  pusat = 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 kontraksi
  for(i in 2:rot){
    xk = A %*% (x0-pusat) # diputar dengan x_bin sebagai pusat
    xk = pusat + (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") +
    geom_point(aes(x = pusat[1],
                  y = pusat[2]),
```

```

        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)
}

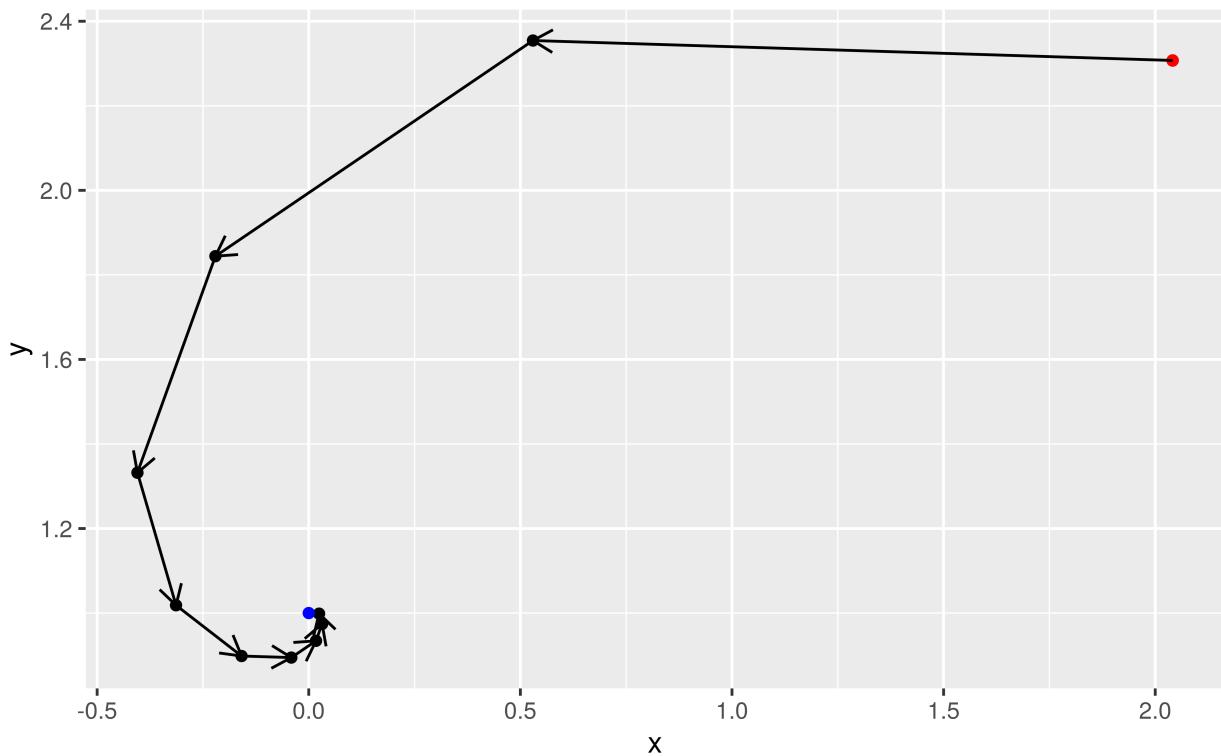
```

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(0,1) # contoh pusat rotasi
r = .6
rotasi_konstraksi_pusat_kan(x0,rot,r,x_bintang)
```

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

titik merah adalah titik initial
titik biru adalah pusat rotasi



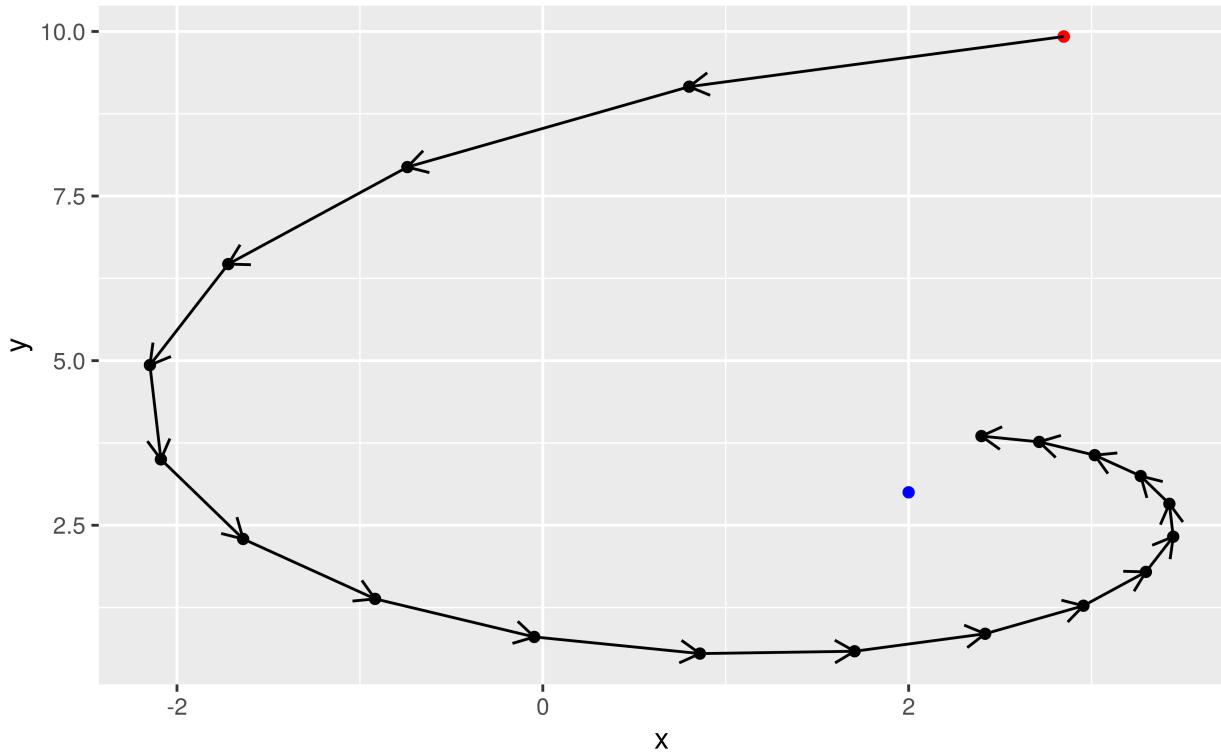
```
##
## $`Titik-titik rotasi`#
##      x      y
## 1  2.0410855 2.307241
## 2  0.5297374 2.354380
## 3 -0.2205109 1.844253
## 4 -0.4047819 1.332041
## 5 -0.3135866 1.018421
## 6 -0.1587148 0.898349
## 7 -0.0411924 0.894683
## 8  0.0171469 0.934351
## 9  0.0314758 0.974180
## 10 0.0243845 0.998568
```

Saya akan coba kembali dengan sembarang titik lainnya:

```
# uji coba
rot = 20 # berapa banyak rotasi
x0 = rand_titik(0,10) # generate random titik
x_bintang = c(2,3) # contoh pusat rotasi
r = .9
rotasi_konstraksi_pusat_kan(x0,rot,r,x_bintang)
```

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

titik merah adalah titik initial
titik biru adalah pusat rotasi



```
##
## $`Titik-titik rotasi`#
##      x      y
## 1  2.8478352 9.922904
## 2  0.8003396 9.161462
## 3 -0.7404471 7.940265
## 4 -1.7196513 6.466464
## 5 -2.1479153 4.932631
## 6 -2.0879059 3.500638
## 7 -1.6382817 2.291613
## 8 -0.9171770 1.381793
## 9 -0.0469122 0.803583
## 10 0.8588009 0.550697
## 11 1.7043781 0.586132
## 12 2.4182957 0.851631
## 13 2.9555348 1.277436
```

```
## 14 3.2969622 1.791319
## 15 3.4462886 2.326133
## 16 3.4253646 2.825438
## 17 3.2685904 3.246999
## 18 3.0171567 3.564234
## 19 2.7137141 3.765843
## 20 2.3979115 3.854019
```

Program *Spiral Optimization Algorithm*

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

```
soa_mrf = function(N,      # banyak titik
                    a,      # batas bawah
                    b,      # batas atas
                    rot,    # berapa banyak rotasi
                    k_max, # iterasi maks
                    r){    # berapa rate konstraksi

# N pasang titik random di selang [a,b] di R2
x1 = runif(N,a,b)
x2 = runif(N,a,b)

# hitung theta
theta = 2*pi / rot

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

# bikin data frame
temp = data.frame(x1,x2) %>% mutate(f = f(x1,x2))

# proses iterasi
for(i in 1:k_max){
  # mencari titik x* dengan min(f)
  f_min = temp %>% filter(f == min(f))
  pusat = c(f_min$x1,f_min$x2)

  for(j in 1:N){
    # kita akan ambil titiknya satu persatu
    x0 = c(temp$x1[j],temp$x2[j])

    # proses rotasi dan konstraksi terhadap pusat x*
    xk = A %*% (x0-pusat) # diputar dengan x_bin sebagai pusat
    xk = pusat + (r * xk)

    # proses mengembalikan nilai ke temp
    temp$x1[j] = xk[1]
    temp$x2[j] = xk[2]
  }

  # hitung kembali nilai f(x1,x2)
  temp = temp %>% mutate(f = f(x1,x2))
}

# proses output hasil
output = temp %>% filter(f == min(f))
return(output)
}
```

Contoh Penggunaan Program

Kita akan coba performa program tersebut untuk menyelesaikan fungsi berikut:

$$f(x_1, x_2) = \frac{x_1^4 - 16x_1^2 + 5x_1}{2} + \frac{x_2^4 - 16x_2^2 + 5x_2}{2}$$

$$-4 \leq x_1, x_2 \leq 4$$

Dengan $r = 0.8, N = 50, rot = 20, k_{max} = 60$.

```
# definisi
N = 50
a = -4
b = 4
k_max = 60
r = .8
rot = 20
f = function(x1,x2){
  ((x1^4 - 16 * x1^2 + 5 * x1)/2) + ((x2^4 - 16 * x2^2 + 5* x2)/2)
}

# solving
soa_mrf(N,a,b,rot,k_max,r)

##           x1           x2           f
## 1 -2.90353 -2.90353 -78.3323
```

Catatan

Pada algoritma ini, penentuan θ, r, x menjadi penentu hasil perhitungan.

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

Pertama-tama, saya akan buat *contour plot* dari $f_1(x_1, x_2)$ sebagai berikut:

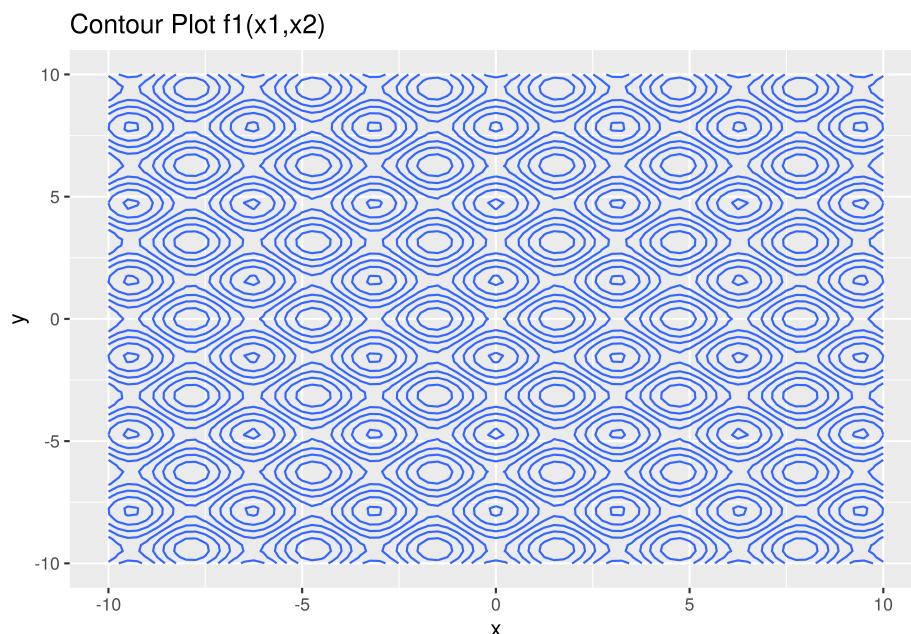


Figure 1: Contour Plot Soal 1: f1

Selanjutnya, saya akan buat *contour plot* dari $f_2(x_1, x_2)$ sebagai berikut:

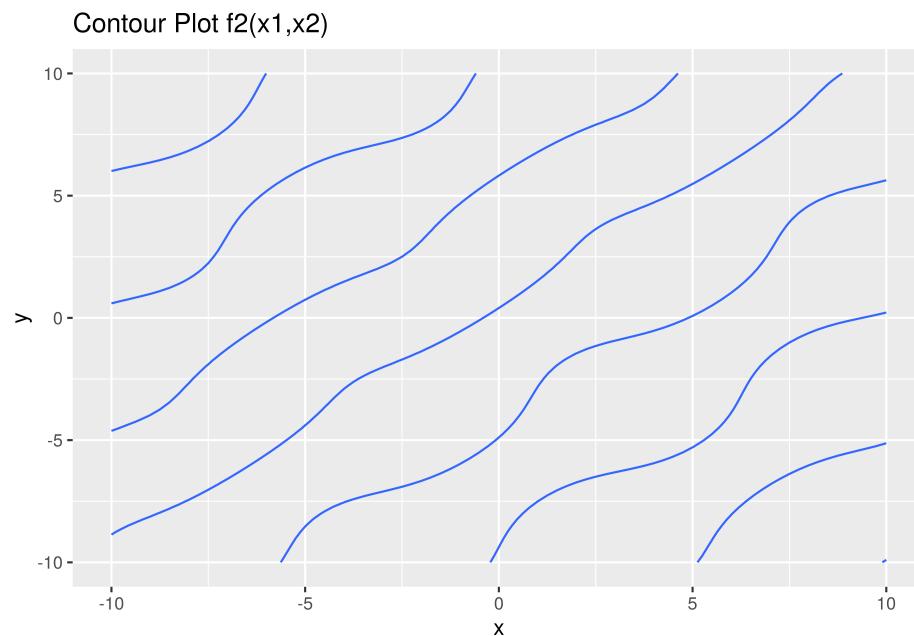


Figure 2: Contour Plot Soal 1: f_2

Grafik Sistem Persamaan

Kita akan mencari akar-akar sistem persamaan saat $f_1 = 0$ dan $f_2 = 0$ dengan bantuan grafik sebagai berikut:

xxxxx

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

Contour Plot

Pertama-tama, saya akan buat *contour plot* dari $f_1(x_1, x_2)$ sebagai berikut:

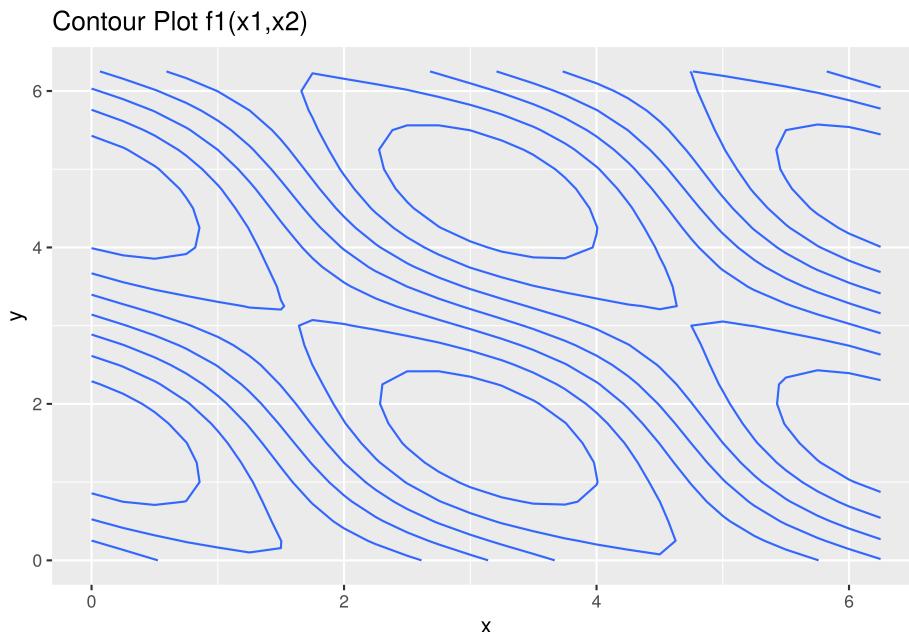


Figure 3: Contour Plot Soal 2: f_1

Berikutnya adalah *contour plot* dari $f_2(x_1, x_2)$ sebagai berikut:

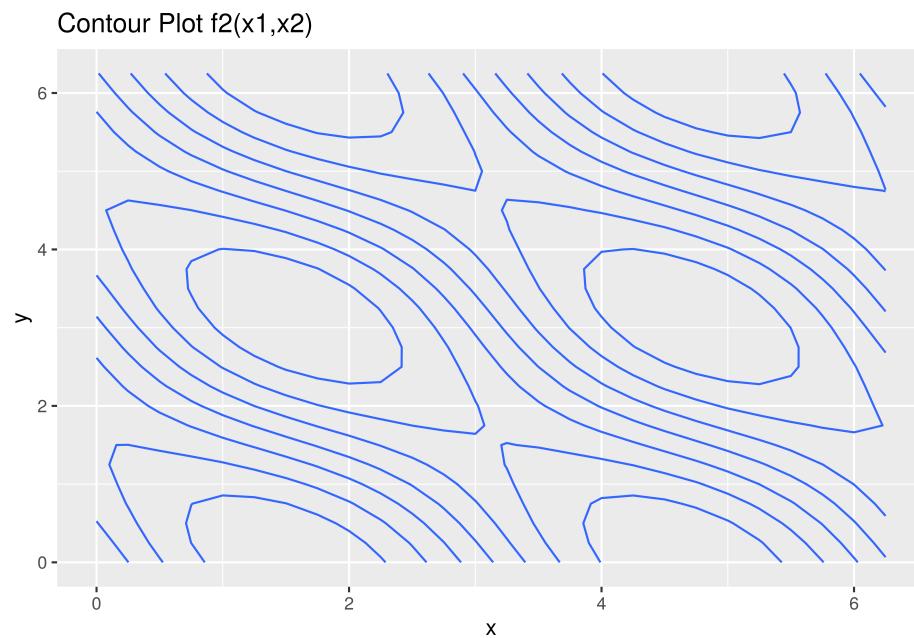


Figure 4: Contour Plot Soal 2: f_2