

NB. 戸田、小野「入門数値計算」 p.196 =====

```
DT3 =: 3 3$3 1 5 4 5 10 0 0 30
```

```
DT31 =: 3 3$3 1 1 4 5 2 0 0 6
```

NB. QR calc on 戸田、小野「入門数値計算」 p.196

NB. programed by T. Nishikawa 2016/2/2

NB. Usage: gramsm DT3 => Q-matrix and R-matrix

```
gramsm =: 3 : 0
```

```
A =. y.
```

```
N =. {. $A
```

NB. Q Matrix Calc. =====

```
i =. 0
```

```
while. i < N
```

```
do.
```

```
  if. i = 0 do.
```

```
    U =. (N, 1)$ i {"(1) A
```

```
    Uab =. (%:@(+/@: *:)), U
```

```
    Uabs =. Uab
```

```
    Q =. U % Uab
```

```
      goto_next.
```

```
  end.
```

```
  if. i > 0
```

```
    do.
```

```
      U =. (N, 1)$ i {"(1) A
```

```
      j =. 0
```

```
      while. j < i
```

```
        do.
```

```
          Aj =. (N, 1)$ i {"(1) A
```

```
          Qj =. (j) {"(1) Q
```

```
          COEF =. ({.@,)(, Aj) ipn Qj
```

```
          U =. U - COEF * Qj
```

```
          j =. j + 1
```

```
        end.
```

```
      Uab =. (%:@(+/@: *:)), U
```

```
      Uabs =. Uabs, Uab
```

```
      Qn =. U % Uab
```

```
      Qn =. cleanz Qn
```

```
      Q =. Q , "(1) Qn
```

```
    end.
```

```

label_next.
  i =. i + 1
end.
NB. R Matrix Calc. =====
R =. (N, N)$0
i =. 0
while. i < N
do.
  j =. 0
  while. j < N
  do.
    if. j = i do.
      R =. (j{Uabs) (<i, j) } R
    end.
    if. i < j do.
      AR =. j {"(1) A
      QR =. i {"(1) Q
      R =. (AR ipn QR) (<i, j) } R
    end.
    j =. j + 1
  end.
  i =. i + 1
end.
NB. Q-matrix and R-matrix Calculated =====
Q;R
)

diag =: i.@# {"_1 ]

NB. QR method =====
qr =: 3 : 0
NB. A1 =. (128!:0) y.      using J primitive
A1 =. gramsm y.          NB. using J program by T. Nishikawa
((>@{:) (+/ . *) (>@{.) A1
)

eigen_value =: 3 : 0
40 eigen_value y.
:
diag qr^:x. y.

```

```

)
NB. 3D graphics ^e2^80^93 Gram-Schmidt Process =====
load 'trig numeric'
require 'gl3'

NB. import from jzopengl
load 'jzopengl'

NB. e.g. coinsert 'jzopengl' =====
coinsert=: 3 : 0
n=. ;: :: ] y.
p=. ; (, 18!:2) @ < each n
p=. ~. (18!:2 coname''), p
(p /: p = <,'z') 18!:2 coname''
)

coinsert 'jzopengl'

A=: noun define
pc a closeok;
xywh 0 0 340 300;cc g isigraph ws_clipchildren ws_clipsiblings rightmove bottommove;
pas 0 0;
rem form end;
)

run=: a_run
a_run=: verb define
JOB =: y.
wd A
glaRC''
NB. R =: 120 0 0
R =: 330 0 188
NB. R =: 90 0 0
NB. T =: 0 0 0
T =: _1.5 _0.5 _6 NB. rl/ud/nf
NB. SC =: 0.375
SC =: 1
glaFont 'arial 30'
glaUseFontBitmaps 0 32 26 32
setfocus g

```

```
wd 'pshow;ptop'
```

```
)
```

```
a_g_size=:verb define
```

```
wh=.glqwh''
```

```
glViewport 0 0,wh
```

```
glMatrixMode GL_PROJECTION
```

```
glLoadIdentity''
```

```
glOrtho _6, 6, _6, 6, _2, 10
```

```
NB. gluPerspective 45, (%/wh),1 8
```

```
)
```

```
NB. Key-In Command =====
```

```
a_g_char =: verb define
```

```
R =: 360 | R + 2 * 'xyz' = 0 { sysdata
```

```
R =: 360 | R - 2 * 'XYZ' = 0 { sysdata
```

```
T =: T + 0.5 * 'run' = 0 { sysdata NB. r(right) = +x, u(up) = +y, n(near) = +z
```

```
T =: T - 0.5 * 'ldf' = 0 { sysdata NB. l(left) = -x, d(down) = -y, f(far) = -z
```

```
SC =: SC * 1 + 0.25 * 'b' = 0 { sysdata NB. bigger
```

```
SC =: SC * 1 - 0.25 * 's' = 0 { sysdata NB. smaller
```

```
glpaintx''
```

```
)
```

```
NB. Paint on the Display =====
```

```
a_g_paint =: verb define
```

```
glClearColor 1 1 1 1
```

```
glClear GL_COLOR_BUFFER_BIT + GL_DEPTH_BUFFER_BIT
```

```
glEnable GL_DEPTH_TEST
```

```
glMatrixMode GL_MODELVIEW
```

```
glLoadIdentity''
```

```
glTranslate T
```

```
glRotate R ,. 3 3 $ 1 0 0 0
```

```
glScale 3#SC
```

```
gramsm1 '' NB. Gram Schmidt Process Display
```

```
drawtext ''
```

```
glaSwapBuffers ''
```

```
)
```

```
NB. Write Data on the display =====
```

```
NB. indicate rotated angle x, y, z in degree
```

```

drawtext =: verb define
glMatrixMode GL_MODELVIEW
glLoadIdentity ''
glColor 0 0 0 1 NB. color is determined by glMaterial Command
glRasterPos _3 _4 _2 NB. be carefull, not change position
glCallLists 10 ": R
glpaintx''
)

```

```

gramsm1 =: 3 : 0
NB. X, Y, Z axes =====
glLineWidth 1
glBegin GL_LINES
  glColor 0 0 0 1 NB. X-axis
  glVertex 0 0 0
  glVertex 6 0 0
  glColor 0 0 0 1 NB. Y-axis
  glVertex 0 0 0
  glVertex 0 6 0
  glColor 0 0 0 1 NB. Z-axis
  glVertex 0 0 0
  glVertex 0 0 7
glEnd ''

```

```

NB. Data Display =====
NB. Using DT31 Data
NB. k=1 step
glColor 1 0 0 1
glPointSize 6
glBegin GL_POINTS
  glVertex 3 4 0
glEnd ''
glLineWidth 1
glBegin GL_LINES
  glVertex (L:0) 3 4 0;0 0 0
glEnd ''
glLineWidth 6
glBegin GL_LINES
  glVertex (L:0) 0.6 0.8 0;0 0 0 NB. Q1

```

```
glEnd ''
```

```
NB. k=2 step
```

```
glColor 0.8 0 1 1 NB. A2
```

```
glPointSize 6
```

```
glBegin GL_POINTS
```

```
glVertex 1 5 0
```

```
glEnd ''
```

```
glLineWidth 1
```

```
glBegin GL_LINES
```

```
glVertex (L:0) 1 5 0;0 0 0
```

```
glColor 1 0 1 1
```

```
glVertex (L:0) 1 5 0;2.76 3.68 0
```

```
glVertex (L:0) 2.76 3.68 0;-1.76 1.32 0
```

```
glVertex (L:0)-1.76 1.32 0;0 0 0
```

```
glEnd ''
```

```
glLineWidth 6
```

```
glBegin GL_LINES
```

```
glVertex (L:0) -0.8 0.6 0;0 0 0 NB. Q2
```

```
glEnd ''
```

```
NB. k=3 step
```

```
glColor 0 1 0 1 NB. A3
```

```
glPointSize 6
```

```
glBegin GL_POINTS
```

```
glVertex 1 2 6
```

```
glEnd ''
```

```
glLineWidth 1
```

```
glBegin GL_LINES
```

```
glVertex (L:0) 1 2 6;0 0 0
```

```
glColor 0 1 0.8 1
```

```
glVertex (L:0) 1 2 6;0 0 0
```

```
glVertex (L:0) 1 2 6;1.32 1.76 0 NB. <a3, q1>*q1
```

```
glVertex (L:0) 1.32 1.76 0; -0.32 0.24 0 NB. <a3, q2>*q2
```

```
glVertex (L:0) -0.32 0.24 0;0 0 6
```

```
glEnd ''
```

```
glLineWidth 6
```

```
glBegin GL_LINES
```

```
glVertex (L:0) 0 0 1;0 0 0 NB. Q3
```

```
glEnd ''
```

)