Maple code for automatic generation of the model of a planar CDPR model with non-straight cables.

Jorge Ivan AYALA CUEVAS, Edouard LAROCHE, Olivier PICCIN

Published in the Third International Conference on Cable-Driven Parallel Robots, Québec city, Canada, 2017

Remark: It is recommended to check first the file containing the model for one single cable in order to better understand the computation procedure.

Initialisation

```
> restart :
> with(VectorCalculus) :
> with(LinearAlgebra) :
> with(CodeGeneration) :
```

Model parameters selection

```
> nc := 3 :        Choose the number of cables
> N := 1 :         Choose the number of displacement modes
```

Creation of empty matrices used for the model

```
> M := Matrix((N + 1)· nc + 2) :          Kinetic Energy Matrix
Error, missing operator or `;`
> Mp := Matrix((N + 1)· nc + 2) :          Derivate wrt time of Kinetic Energy
Matrix
> qp := Matrix((N + 1)·nc + 2, 1) :         Generalized velocities vector
> qpp := Matrix((N + 1)·nc + 2, 1) :        Accelerations vector
> CC := Matrix((N + 1)·nc + 2, 1) :         Centrifugal and Coriollis forces matrix
> U := Matrix((N + 1)·nc + 2, 1) :          Potential energy matrix
> gamma1 := Matrix((N + 1)·nc + 2, nc) :    Applied forces jacobien
> A := Matrix(2·nc, ((N + 1)·nc + 2)) :     Geometrical consttraints Matrix
> Ap := Matrix(2·nc, ((N + 1)·nc + 2)) :    Derivate wrt time of Constraints Matrix
```

Loop code to full the model matrices - based on computation presented in one single cable code

> **for** i **from** 1 **to** nc **do**

$$\delta y \| i := 0 :$$

$$vxVN \| i := 0 :$$

$$vyVN \| i := 0 :$$

$$EcaN \| i := 0 :$$

for j **from** 2 **to** N **do**

$$\delta y \| i := \delta y \| i + \left(V \| j \| i \right) \cdot x^{j} :$$

end do:

$$d\delta ydx \| i := diff\left(\delta y \| i, x \right) :$$

$$\delta x \| i := -\frac{1}{2} \cdot int\left(\left(d\delta ydx \| i \right)^{2}, x = 0 .. x \right) :$$

$$X1 \| i := simplify\left(x + \delta x \| i \right) :$$

$$Y1 \| i := \delta y \| i :$$

$$X2 \| i := \left(\left(X1 \| i \right) \cdot \cos\left(\phi \| i \right) - \left(Y1 \| i \right) \cdot \sin\left(\phi \| i \right) \right) + XA \| i :$$

$$Y2 \| i := \left(\left(X1 \| i \right) \cdot \sin\left(\phi \| i \right) + \left(Y1 \| i \right) \cdot \cos\left(\phi \| i \right) \right) + YA \| i :$$

$$Ep \| i := \rho \cdot g \cdot int\left(Y2 \| i, x = 0 .. l \| i \right) :$$

for j **from** 2 **to** N **do**

$$dX2dV \| j \| i := diff\left(X2 \| i, V \| j \| i \right) :$$

$$dX2dx \| i := diff\left(X2 \| i, x \right) :$$

$$dX2d\phi \| i := diff\left(X2 \| i, \phi \| i \right) :$$

$$dY2dV \| j \| i := diff\left(Y2 \| i, V \| j \| i \right) :$$

$$dY2dx \| i := diff\left(Y2 \| i, x \right) :$$

```
dY2dϕ||i := diff(Y2||i, ϕ||i) :
vxVN||i := vxVN||i + dX2dV||j||i·V||j||p||i:
vyVN||i := vyVN||i + dY2dV||j||i·V||j||p||i:
dEpdl||i := diff(Ep||i, l||i);
dEpdV||j||i := diff(Ep||i, V||j||i):
dEpdϕ||i := diff(Ep||i, ϕ||i):
end do:
vx||i := vxVN||i + dX2dx||i·lp||i + dX2dϕ||i·ϕp||i:
vy||i := vyVN||i + dY2dx||i·lp||i + dY2dϕ||i·ϕp||i:
v2||i := simplify((vx||i)^2 + (vy||i)^2) :
Ec1||i := 1/2·ρ·int(v2||i, x = 0 .. (l||i)) :
Ec1||i := simplify(Ec1||i) :
J||i := J0 + ρ·(Lt - l||i)/2·R^2 :
Ec2||i := simplify(1/2·J||i·(lp||i/R)^2) :
Ecc||i := simplify(Ec1||i + Ec2||i) :
for j from 2 to N do
M[((N + 1)·i - N), ((N + 1)·i - N)] := 2·coeff(Ecc||i, (lp||i), 2);
M[((N + 1)·i - N + j - 1), ((N + 1)·i - N + j - 1)] := 2·coeff(Ecc||i, (V||j||p||i),
    2);
M[((N + 1)·i), ((N + 1)·i)] := 2·coeff(Ecc||i, (ϕp||i), 2);
EcaN||i := simplify(EcaN||i + 1/2·M[((N + 1)·i - N + j - 1), ((N + 1)·i - N + j
    - 1)]·(V||j||p||i)^2);
end do:
```

```
Eca||i := simplify(Ecc||i − 1/2 · M[((N + 1)·i − N), ((N + 1)·i − N)]·(lp||i)^2
    − EcaN||i − 1/2 ·M[((N + 1)·i), ((N + 1)·i)]·(φp||i)^2);
for j from 2 to N do
EcV||j||i := coeff(Eca||i, (V||j||p||i), 1);
M[((N + 1)·i − N),((N + 1)· i − N + j − 1)] := coeff(EcV||j||i, (lp||i), 1);
M[((N + 1)· i − N + j − 1), ((N + 1)·i − N)] := M[((N + 1)·i − N),((N + 1)· i
    − N + j − 1)];
M[((N + 1)·i),((N + 1)· i − N + j − 1)] := coeff(EcV||j||i, (φp||i), 1);
M[((N + 1)· i − N + j − 1), ((N + 1)·i)] := M[((N + 1)·i),((N + 1)· i − N + j
    − 1)];
for k from (j + 1) to N do
M[((N + 1)· i − N + j − 1),((N + 1)· i − N + k − 1)] := coeff(EcV||j||i, (V||k
    ||p||i), 1);
M[((N + 1)· i − N + k − 1),((N + 1)· i − N + j − 1)] := M[((N + 1)· i − N
    + j − 1),((N + 1)· i − N + k − 1)];
end do:
Ecl||i := coeff(Eca||i, (lp||i), 1);
M[((N + 1)·i − N), ((N + 1)·i)] := coeff(Ecl||i, (φp||i), 1);
M[((N + 1)·i), ((N + 1)·i − N)] := M[((N + 1)·i − N), ((N + 1)·i)];
dMdl||i := (map(diff, M, l||i));
dMdV||j||i := (map(diff, M, V||j||i) );
Mp := Mp + (dMdV||j||i·V||j||p||i );
end do:
```

```
Mp := Mp + (dMdl||i·lp||i);
qp[((N+1)·i−N), 1] := lp||i;
qp[((N+1)·i), 1] := ϕp||i;
qpp[((N+1)·i−N), 1] := lpp||i;
qpp[((N+1)·i), 1] := ϕpp||i;
for j from 2 to N do
 qp[((N+1)·i−N+j−1), 1] := V||j||p||i;
 qpp[((N+1)·i−N+j−1), 1] := V||j||pp||i;
 CC[((N+1)·i−N+j−1), 1] := simplify(1/2 · qp^%T.dMdV||j||i.qp);
CC||((N+1)·i−N+j−1) := CC[((N+1)·i−N+j−1), 1];
CC||((N+1)·i−N+j−1)||((N+1)·i−N+j−1) := CC||((N+1)·i−N+j
    −1)[1, 1];
 CC[((N+1)·i−N+j−1), 1] := CC||((N+1)·i−N+j−1)||((N+1)·i−N
    +j−1);
U[((N+1)·i−N+j−1), 1] := dEpdV||j||i;
end do:
CC[((N+1)·i−N), 1] := simplify(1/2 · qp^%T.dMdl||i.qp);
CC||((N+1)·i−N) := CC[((N+1)·i−N), 1];
CC||((N+1)·i−N)||((N+1)·i−N) := CC||((N+1)·i−N)[1, 1];
CC[((N+1)·i−N), 1] := CC||((N+1)·i−N)||((N+1)·i−N);
U[((N+1)·i−N), 1] := dEpdl||i;
U[(N+1)·i, 1] := dEpdϕ||i;
gamma1[((N+1)·i−N), i] := -1/R;
vxl||i := subs(x = l||i, vx||i);
```

```
vyl||i := subs(x = l||i, vy||i);
A1||i := simplify((xnp − vxl||i));
A2||i := simplify((ynp − vyl||i));
A[(2·i − 1), ((N + 1)·i − N)] := diff(A1||i, lp||i);
A[(2·i − 1), ((N + 1)·i)] := diff(A1||i, ϕp||i);
A[(2·i − 1), ((N + 1)·nc + 1)] := diff(A1||i, xnp);
A[(2·i − 1), ((N + 1)·nc + 2)] := diff(A1||i, ynp);
A[(2·i), ((N + 1)·i − N)] := diff(A2||i, lp||i);
A[(2·i), ((N + 1)·i)] := diff(A2||i, ϕp||i);
A[(2·i), ((N + 1)·nc + 1)] := diff(A2||i, xnp);
A[(2·i), ((N + 1)·nc + 2)] := diff(A2||i, ynp);
for j from 2 to N do
A[(2·i − 1), ((N + 1)· i − N + j − 1)] := diff(A1||i, V||j||p||i);
A[(2·i), ((N + 1)· i − N + j − 1)] := diff(A2||i, V||j||p||i);
dAdV||j||i := (map(diff, A, V||j||i));
Ap := simplify(Ap + (dAdV||j||i·V||j||p||i));
end do:
dAdl||i := (map(diff, A, l||i));
dAdϕ||i := (map(diff, A, ϕ||i));
Ap := simplify(Ap + (dAdl||i·lp||i + dAdϕ||i·ϕp||i));
end do:
> vn2 := xnp^2 + ynp^2 :
> Ecn := 1/2·Mn·vn2 :
```

> $Ec := simplify(Ecc + Ecn):$
> $M[((N+1)\cdot nc+1),((N+1)\cdot nc+1)] := 2\cdot coeff(Ec, xnp, 2):$
> $M[((N+1)\cdot nc+2),((N+1)\cdot nc+2)] := 2\cdot coeff(Ec, ynp, 2):$
> $U[((N+1)\cdot nc+2),1] := Mn\cdot g:$
> $qp[((N+1)\cdot nc+1),1] := xp:$
> $qp[((N+1)\cdot nc+2),1] := yp:$
> $qpp[((N+1)\cdot nc+1),1] := xpp:$
> $qpp[((N+1)\cdot nc+2),1] := ypp:$

Obtention of vectors and matrices

Potential Energy Matrix U

> $U;$

$$\begin{bmatrix} dEpdl1 \\ dEpd\phi 1 \\ dEpdl2 \\ dEpd\phi 2 \\ dEpdl3 \\ dEpd\phi 3 \\ 0 \\ Mn\, g \end{bmatrix} \quad \textbf{(5.1.1)}$$

Kinetic Energy Matrix M

> $M;$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & Mn & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & Mn \end{bmatrix} \quad \textbf{(5.2.1)}$$

d/dt (M)

> $Mp;$

$$dMdl1\ lp1 + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + dMdl2\ lp2 + dMdl3\ lp3 \qquad \textbf{(5.3.1)}$$

Generalized velocities vector

> qp;

$$\begin{bmatrix} lp1 \\ \phi p1 \\ lp2 \\ \phi p2 \\ lp3 \\ \phi p3 \\ xp \\ yp \end{bmatrix} \qquad \textbf{(5.4.1)}$$

Accelerations vector

> qpp;

$$\begin{bmatrix} lpp1 \\ \phi pp1 \\ lpp2 \\ \phi pp2 \\ lpp3 \\ \phi pp3 \\ xpp \\ ypp \end{bmatrix} \qquad \textbf{(5.5.1)}$$

Centrifugal and Coriollis forces matrix C

> CC;

$$\begin{bmatrix} \frac{dMdl1\left(\phi p1^2 + lp1^2\right)}{2} \\ 0 \\ \frac{dMdl2\left(\phi p1^2 + \phi p2^2 + lp1^2 + lp2^2\right)}{2} \\ 0 \\ \frac{dMdl3\left(\phi p1^2 + \phi p2^2 + \phi p3^2 + lp1^2 + lp2^2 + lp3^2\right)}{2} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (5.6.1)$$

Applied Efforts Matrix Γ

> *gamma1*;

$$\begin{bmatrix} -\frac{1}{R} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -\frac{1}{R} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{R} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (5.7.1)$$

Constraints Matrix A

> *A*;

$$\begin{bmatrix} -dX2dx1 & -dX2d\phi1 & 0 & 0 & 0 & 0 & 1 & 0 \\ -dY2dx1 & -dY2d\phi1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -dX2dx2 & -dX2d\phi2 & 0 & 0 & 1 & 0 \\ 0 & 0 & -dY2dx2 & -dY2d\phi2 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -dX2dx3 & -dX2d\phi3 & 1 & 0 \\ 0 & 0 & 0 & 0 & -dY2dx3 & -dY2d\phi3 & 0 & 1 \end{bmatrix} \quad (5.8.1)$$

d/dt (A)

```
> Ap := simplify(Ap);
```

$$Ap := \begin{bmatrix} 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0 \end{bmatrix} \qquad \textbf{(5.9.1)}$$

```
>
```

Generation of Matrix [M A^T; A 0] (Matrix MMA for future identification)

This matrix is used to solve the system of linear equations with respect to qpp and λ. It is necessary to invert this matrix in order to solve the equations, in this case, the inversion will be done online during the computation in matlab.

```
> MMA := Matrix(((N+3)·nc+2), ((N+3)·nc+2)) :
> for i from 1 to nc do
   for j from 2 to N do
    MMA[((N+1)·i−N), ((N+1)·i−N)] := M[((N+1)·i−N), ((N+1)·i
      −N)];
   MMA[((N+1)· i−N+j−1), ((N+1)· i−N+j−1)] := M[((N+1)· i−N+j
      −1), ((N+1)· i−N+j−1)];
   MMA[((N+1)·i), ((N+1)·i)] := M[((N+1)·i), ((N+1)·i)];
    MMA[((N+1)·i−N), ((N+1)· i−N+j−1)] := M[((N+1)·i−N), ((N+1)
      · i−N+j−1)];
   MMA[((N+1)· i−N+j−1), ((N+1)·i−N)] := M[((N+1)· i−N+j−1),
      ((N+1)·i−N)];
   MMA[((N+1)·i−N), ((N+1)·i)] := M[((N+1)·i−N), ((N+1)·i)];
   MMA[((N+1)·i), ((N+1)·i−N)] := M[((N+1)·i), ((N+1)·i−N)];
   MMA[((N+1)· i−N+j−1), ((N+1)·i)] := M[((N+1)· i−N+j−1), ((N
      +1)·i)];
   MMA[((N+1)·i), ((N+1)· i−N+j−1)] := M[((N+1)·i), ((N+1)· i−N+j
      −1)];
    MMA[((N+1)·nc+1), ((N+1)·nc+1)] := M[((N+1)·nc+1), ((N+1)·nc
      +1)];
    MMA[((N+1)·nc+2), ((N+1)·nc+2)] := M[((N+1)·nc+2), ((N+1)·nc
      +2)];
    MMA[(2·i+1 + (N+1)·nc), ((N+1)·i−N)] := A[(2·i−1), ((N+1)·i−N)];
    MMA[(2·i+1 + (N+1)·nc), ((N+1)· i−N+j−1)] := A[(2·i−1), ((N+1)· i
      −N+j−1)];
     MMA[(2·i+1 + (N+1)·nc), ((N+1)·i)] := A[(2·i−1), ((N+1)·i)];
    MMA[(2·i+1 + (N+1)·nc), ((N+1)·nc+1)] := A[(2·i−1), ((N+1)·nc+1)];
```

$MMA[(2 \cdot i+1+(N+1) \cdot nc), ((N+1) \cdot nc+2)] := A[(2 \cdot i-1), ((N+1) \cdot nc+2)];$
$MMA[(2 \cdot i+2+(N+1) \cdot nc), ((N+1) \cdot i-N)] := A[(2 \cdot i), ((N+1) \cdot i-N)];$
$MMA[(2 \cdot i+2+(N+1) \cdot nc), ((N+1) \cdot i-N+j-1)] := A[(2 \cdot i), ((N+1) \cdot i-N +j-1)];$
$MMA[(2 \cdot i+2+(N+1) \cdot nc), ((N+1) \cdot i)] := A[(2 \cdot i), ((N+1) \cdot i)];$
$MMA[(2 \cdot i+2+(N+1) \cdot nc), ((N+1) \cdot nc+1)] := A[(2 \cdot i), ((N+1) \cdot nc+1)];$
$MMA[(2 \cdot i+2+(N+1) \cdot nc), ((N+1) \cdot nc+2)] := A[(2 \cdot i), ((N+1) \cdot nc+2)];$
$MMA[((N+1) \cdot i-N), (2 \cdot i+1+(N+1) \cdot nc)] := A[(2 \cdot i-1), ((N+1) \cdot i-N)];$
$MMA[((N+1) \cdot i-N+j-1), (2 \cdot i+1+(N+1) \cdot nc)] := A[(2 \cdot i-1), ((N+1) \cdot i -N+j-1)];$
$MMA[((N+1) \cdot i), (2 \cdot i+1+(N+1) \cdot nc)] := A[(2 \cdot i-1), ((N+1) \cdot i)];$
$MMA[((N+1) \cdot nc+1), (2 \cdot i+1+(N+1) \cdot nc)] := A[(2 \cdot i-1), ((N+1) \cdot nc+1)];$
$MMA[((N+1) \cdot nc+2), (2 \cdot i+1+(N+1) \cdot nc)] := A[(2 \cdot i-1), ((N+1) \cdot nc+2)];$
$MMA[((N+1) \cdot i-N), (2 \cdot i+2+(N+1) \cdot nc)] := A[(2 \cdot i), ((N+1) \cdot i-N)];$
$MMA[((N+1) \cdot i-N+j-1), (2 \cdot i+2+(N+1) \cdot nc)] := A[(2 \cdot i), ((N+1) \cdot i-N +j-1)];$
$MMA[((N+1) \cdot i), (2 \cdot i+2+(N+1) \cdot nc)] := A[(2 \cdot i), ((N+1) \cdot i)];$
$MMA[((N+1) \cdot nc+1), (2 \cdot i+2+(N+1) \cdot nc)] := A[(2 \cdot i), ((N+1) \cdot nc+1)];$
$MMA[((N+1) \cdot nc+2), (2 \cdot i+2+(N+1) \cdot nc)] := A[(2 \cdot i), ((N+1) \cdot nc+2)];$
end do:
end do:

Generation of Matlab code

Potential Energy Matrix U

> $Matlab(U, resultname = "U");$

```
Warning, the following variable name replacements were made: 
dEpd&phi;1 -> cg, dEpd&phi;2 -> cg1, dEpd&phi;3 -> cg3
U = [dEpdl1; cg; dEpdl2; cg1; dEpdl3; cg3; 0; Mn * g;];
```

Applied efforts Matrix Γ

> $Matlab(gamma1, resultname = "gam");$

```
gam = [-0.1e1 / CodeGeneration:-R 0 0; 0 0 0; 0 -0.1e1 /
CodeGeneration:-R 0; 0 0 0; 0 0 -0.1e1 / CodeGeneration:-R; 0
0 0; 0 0 0; 0 0 0;];
```

d/dt (M)

> $Matlab(Mp, resultname = "Mp");$

```
Warning, precedence for Array unspecified
Mp = dMdl1 * lp1 + ([0 0 0 0 0 0 0 0; 0 0 0 0 0 0 0 0; 0 0 0 0
0 0 0 0; 0 0 0 0 0 0 0 0; 0 0 0 0 0 0 0 0; 0 0 0 0 0 0 0 0; 0
0 0 0 0 0 0 0; 0 0 0 0 0 0 0 0;]) + dMdl2 * lp2 + dMdl3 * lp3;
```

d/dt (A)

> **for** i **from** 1 **to** nc **do**

```
Ap := subs(ϕp||i = phip||i, Ap);
end do:
> Matlab(Ap, resultname = "Ap");
Ap = [0 0 0 0 0 0 0 0; 0 0 0 0 0 0 0 0; 0 0 0 0 0 0 0 0; 0 0 0
0 0 0 0 0; 0 0 0 0 0 0 0 0; 0 0 0 0 0 0 0 0;];
>
```

Centrifugal and Corillis Forces Matrix.

```
> for i from 1 to nc do
  CC[((N + 1)·i − N), 1] := subs(ϕp||i = phip||i, CC[((N + 1)·i − N), 1]);
   for j from 2 to N do
    CC[((N + 1)· i − N + j − 1), 1] := subs(ϕp||i = phip||i, CC[((N + 1)· i − N + j
       − 1), 1]);
  end do:
  end do:
> Matlab(CC, resultname = "CC");
Warning, the following variable name replacements were made:
&phi;p1 -> cg, &phi;p2 -> cg1
CC = [dMdl1 * (lp1 ^ 2 + phip1 ^ 2) / 0.2e1; 0; dMdl2 * (cg ^
2 + lp1 ^ 2 + lp2 ^ 2 + phip2 ^ 2) / 0.2e1; 0; dMdl3 * (cg ^ 2
+ cg1 ^ 2 + lp1 ^ 2 + lp2 ^ 2 + lp3 ^ 2 + phip3 ^ 2) / 0.2e1;
0; 0; 0;];
```

Matrix MMA

```
> Matlab(MMA, resultname = "MMA");
MMA = [0 0 0 0 0 0 0 0 0 0 0 0 0 0; 0 0 0 0 0 0 0 0 0 0 0 0 0
0; 0 0 0 0 0 0 0 0 0 0 0 0 0 0; 0 0 0 0 0 0 0 0 0 0 0 0 0 0; 0
0 0 0 0 0 0 0 0 0 0 0 0 0; 0 0 0 0 0 0 0 0 0 0 0 0 0 0; 0 0 0
0 0 0 0 0 0 0 0 0 0 0; 0 0 0 0 0 0 0 0 0 0 0 0 0 0; 0 0 0 0 0
0 0 0 0 0 0 0 0 0; 0 0 0 0 0 0 0 0 0 0 0 0 0 0; 0 0 0 0 0 0 0
0 0 0 0 0 0 0; 0 0 0 0 0 0 0 0 0 0 0 0 0 0; 0 0 0 0 0 0 0 0 0
0 0 0 0 0; 0 0 0 0 0 0 0 0 0 0 0 0 0 0;];
>
```