L'objectif de ce TD est d'implémenter l'algorithme du pivot de Gauss afin de solutionner un problème de Statique en SII.

# I. Mise en situation

Cet exercice, qui sera traité cette année en TD de SII, a pour objectif de connaître toutes les inconnues d'actions mécaniques de liaison dans les liaisons entre les constituants d'une passerelle télescopique d'aéroport. La connaissance de ces actions mécaniques entre composants permettrait par la suite de dimensionner les composants pour éviter toute casse ou usure prématurée face aux efforts qu'ils subissent lors de l'utilisation.

Le schéma cinématique de la passerelle est donné, ainsi que les masses et longueurs des différents composants.

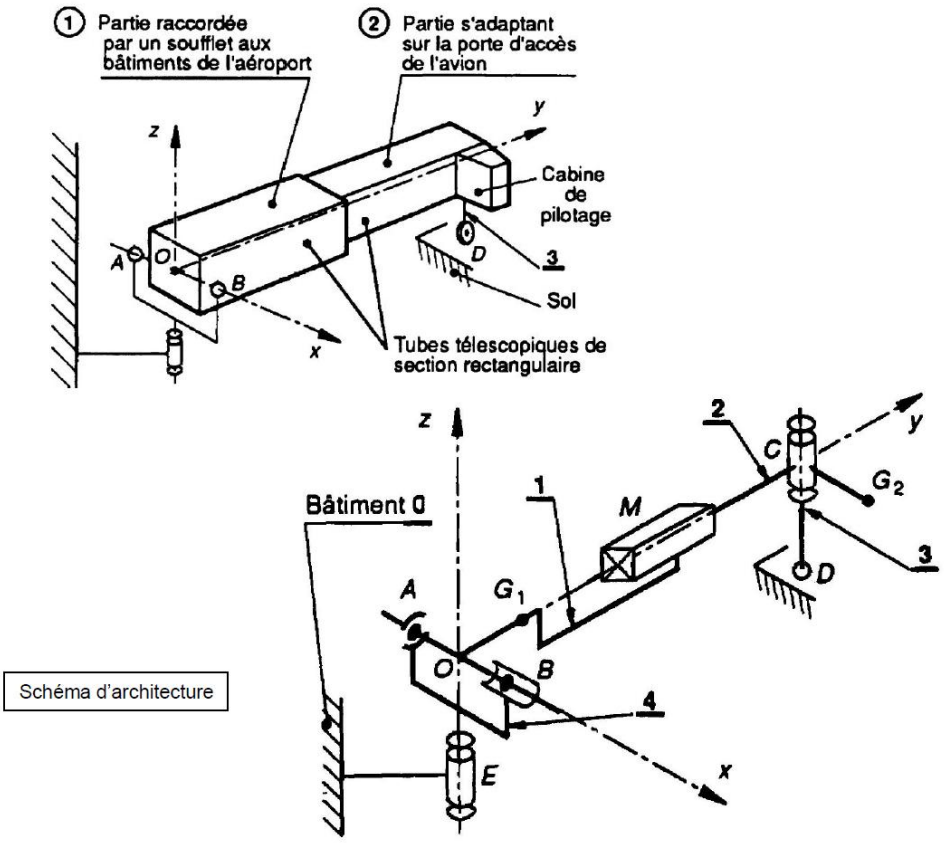


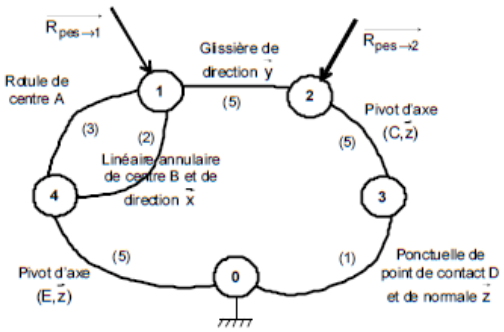

Schéma d'architecture

Il y a en tout 21 inconnues d'actions mécaniques de liaison recherchées dans la passerelle télescopique d'aéroport :

$$\{T_{0\to4}\}=\begin{Bmatrix}X_{0\to4} & L_{E,0\to4}\\ Y_{0\to4} & M_{E,0\to4}\\ Z_{0\to4} & 0\end{Bmatrix}_{E,(\vec{x},\vec{y},\vec{z})} \qquad \{T_{0\to3}\}=\begin{Bmatrix}0 & 0\\ 0 & 0\\ Z_{0\to3} & 0\end{Bmatrix}_{\forall P\in(D,\vec{z}),(\vec{x},\vec{y},\vec{z})}$$

$$\{T_{1\to4}{}^{LA}\}=\begin{Bmatrix}X^{LA}_{1\to4} & 0\\ Y^{LA}_{1\to4} & 0\\ Z^{LA}_{1\to4} & 0\end{Bmatrix}_{A,(\vec{x},\vec{y},\vec{z})} \qquad \{T_{1\to4}{}^{LB}\}=\begin{Bmatrix}0 & 0\\ Y^{LB}_{1\to4} & 0\\ Z^{LB}_{1\to4} & 0\end{Bmatrix}_{B,(\vec{x},\vec{y},\vec{z})}$$

$$\{T_{2\to1}\}=\begin{Bmatrix}X_{2\to1} & L_{M,2\to1}\\ 0 & M_{M,2\to1}\\ Z_{2\to1} & N_{M,2\to1}\end{Bmatrix}_{M,(\vec{x},\vec{y},\vec{z})} \qquad \{T_{2\to3}\}=\begin{Bmatrix}X_{2\to3} & L_{C,2\to3}\\ Y_{2\to3} & M_{C,2\to3}\\ Z_{2\to3} & 0\end{Bmatrix}_{C,(\vec{x},\vec{y},\vec{z})}$$




L'application du Principe Fondamental de la Statique aux ensembles de solides {1,2,3,4}, {4}, {1,4} et {3} permet d'obtenir 21 équations reliant les inconnues (et 3 équations inutiles du type 0=0) :

$$\begin{cases}X_{0\to4}=0\\ Y_{0\to4}=0\\ Z_{0\to4}+Z_{0\to3}-m_1.g-m_2.g=0\\ L_{E,0\to4}+y_0.Z_{0\to3}-d.m_1.g-y_0.m_2.g=0\\ M_{E,0\to4}+e.m_2.g=0\\ 0=0\end{cases} \qquad \begin{cases}X^{LA}_{1\to4}=0\\ Y^{LA}_{1\to4}+Y^{LB}_{1\to4}=0\\ Z^{LA}_{1\to4}+Z_{0\to4}+Z^{LB}_{1\to4}=0\\ L_{E,0\to4}=0\\ -e.m_2.g-2.a.Z^{LB}_{1\to4}-a.Z_{0\to4}=0\\ 2.a.Y^{LB}_{1\to4}=0\end{cases}$$

$$\begin{cases}X_{2\to1}=0\\ 0=0\\ Z_{2\to1}+\dfrac{(y_0-d).m_1.g}{y_0}-m_1.g=0\\ L_{M,2\to1}-l.\dfrac{(y_0-d).m_1.g}{y_0}+(l-d).m_1.g=0\\ M_{M,2\to1}-e.m_2.g=0\\ N_{M,2\to1}=0\end{cases} \qquad \begin{cases}X_{2\to3}=0\\ Y_{2\to3}=0\\ Z_{2\to3}+\dfrac{d.m_1.g+y_0.m_2.g}{y_0}=0\\ L_{C,2\to3}=0\\ M_{C,2\to3}=0\\ 0=0\end{cases}$$

*Remarque :* On prendra g=9.81m/s2. Les masses et les longueurs caractéristiques du système sont connues :

$y_0$ = 16 m
d = 6 m
a = 1,5 m
e = 1 m
h = 3 m
$\ell$ = 7 m

$m_1$ = $10^4$ kg
$m_2$ = $15.10^3$ kg

*Objectif du TD d'informatique :* On souhaite résoudre numériquement le système linéaire obtenu de 21 équations pour déterminer les 21 inconnues d'actions mécaniques du système.

# II. Méthode du Pivot de Gauss de résolution d'un système linéaire

Afin de résoudre le système linéaire de 21 équations à 21 inconnues, on se propose ici de programmer la méthode dite du Pivot de Gauss de résolution d'un système linéaire.

## a. Définition des matrices de départ

On considère au départ un système linéaire de n équations à n inconnues (S) suivant :

$$(S):\begin{cases} a_{1,1}x_1 + \ldots + a_{1,n}x_n = b_1 \\ \ldots \\ a_{n,1}x_1 + \ldots + a_{n,n}x_n = b_n \end{cases}$$

La matrice A du système, le vecteur inconnu et le vecteur B second membre du système (S) sont définis par :

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \ldots & a_{1,n} \\ a_{2,1} & a_{2,2} & \ldots & a_{2,n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{n,1} & a_{n,2} & \ldots & a_{n,n} \end{pmatrix} \qquad X = \begin{pmatrix} x_1 \\ x_2 \\ \ldots \\ x_n \end{pmatrix} \qquad B = \begin{pmatrix} b_1 \\ b_2 \\ \ldots \\ b_n \end{pmatrix}$$

L'écriture matricielle du système est : $AX = B$

Pour la passerelle télescopique d'aéroport, les matrices A et B de départ sont données dans le fichier python « pivot_gauss.py » présent sur le réseau et ont été construites grâce au système de 21 équations à 21 inconnnues, compilé dans le tableau suivant :

| X04 | Y04 | Z04 | L04 | MO4 | Z03 | XA14 | YA14 | ZA14 | YB14 | ZB14 | X21 | Z21 | L21 | M21 | N21 | X23 | Y23 | Z23 | L23 | M23 | 2nd terme B |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 245250 |
| 0 | 0 | 0 | 1 | 0 | 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2943000 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -147150 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | -2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 147150 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 36788 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 331088 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 147150 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | -183938 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

*Rappel*: on écrit une matrice par une liste de liste : A = [[ , , , ],[ , , , ],[ , , , ]] où chaque sous liste est une ligne de A (elles ont donc toutes la même longueur). Pour accéder à la ligne i on peut donc directement accéder à A[i]. Pour avoir accès au coefficient d'indice (i; j) on peut donc taper A[i][j].

### b. Programmation de la méthode du Pivot de Gauss

Compléter le fichier « pivot_gauss.py », puis appliquer votre programme aux matrices A et B données pour résoudre le système des 21 équations à 21 inconnues de la passerelle télescopique d'aéroport.

*Remarque :* Pour les fonctions « echelonne » et « reduit » on traduit la présence optionnelle d'un 2nd membre B en écrivant une valeur par défaut :

```
def f(A, B=None):
        #si B n'est pas donnée à l'appel de la fonction on a B == None
        pass
```

Pour vérifier si B est donnée ou non, on utilisera :

```
if B is not None:
        #ici B a été spécifiée
else: #généralement on ne fera rien et donc on ne notera pas le else
```

## III. Construction des matrices A et B

On donne le fichier .csv « Construction_matrices_systeme .csv » sur le réseau.

Utiliser les données de ce fichier .csv pour construire dans le fichier python « construction_matrices.py » les matrices A et B qui vous ont été données au début.

## IV. Opération sur les matrices

L'objectif de cette partie est de programmer des opérations élémentaires sur les matrices, pour, à la fin, afficher directement les solutions possibles d'un système AX=B (en différenciant les cas d'un système sans solutions, avec solution unique, ou avec une infinité de solutions).

Compléter le fichier « matrices.py » dans ce but.

*Remarque :* Les opérations précédemment réalisées sur les matrices modifient la matrice donnée de départ. C'est une mauvaise pratique en général de modifier les données fournies. Il est donc préférable de copier les matrices avant d'appliquer nos fonctions, en utilisant par exemple la fonction *deepcopy* à importer du module *copy*:

```
from copy import deepcopy
```