



POLYTECHNIQUE
MONTRÉAL

PHS 4700
Physique pour les applications multimédia

**Chapitre 3 — Résolution numérique des
équations de la cinématique**

Djamel Seddaoui

Table des matières

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

- **Introduction**
- **Résolution des équations de la cinématique des particules**
- **Résolution des équations de la cinématique des solides**
- **Résolution numérique d'EDO**
- **Solutions numériques et rotations**
- **Solutions numériques et collisions**
- **Conclusion**

Introduction

Introduction

Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

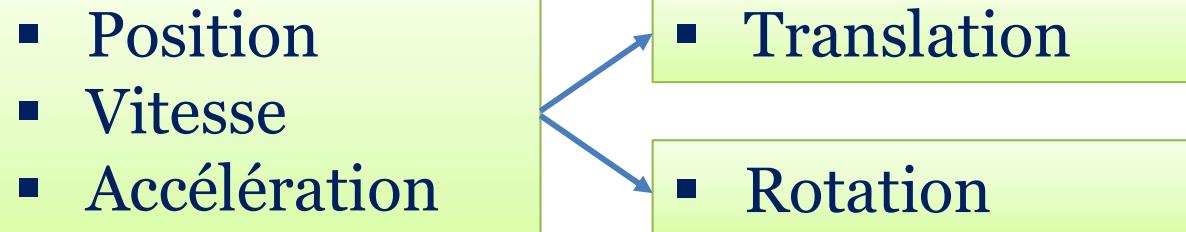
Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

La cinématique des solides.

La cinématique: étude des mouvements des solides sans la prise en compte des causes de ces mouvements.



Introduction

Introduction

Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Mouvement d'un solide \equiv Mouvement de translation du CM + Mouvement de rotation /CM

Mouvement de translation du CM

$$\frac{d\vec{v}_c(t)}{dt} = \vec{a}_c(t, \vec{v}_c(t), \vec{r}_c(t), \vec{\omega}(t), \vec{\Omega}(t))$$

$$\frac{d\vec{r}_c(t)}{dt} = \vec{v}_c(t, \vec{v}_c(t), \vec{r}_c(t), \vec{\omega}(t), \vec{\Omega}(t))$$

Mouvement de rotation /CM

$$\frac{d\vec{\omega}(t)}{dt} = \vec{\alpha}(t, \vec{v}_c(t), \vec{r}_c(t), \vec{\omega}(t), \vec{\Omega}(t))$$

$$\frac{d\vec{\Omega}(t)}{dt} = \vec{\omega}(t, \vec{v}_c(t), \vec{r}_c(t), \vec{\omega}(t), \vec{\Omega}(t))$$

4 équations différentielles vectorielles couplées
(12 équations au total).

Introduction

Introduction

Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

$$\frac{d\mathbf{R}(t)}{dt} = \tilde{\omega}(t)\mathbf{R}(t)$$

avec

Mouvement de rotation /CM $\tilde{\omega}(t) \begin{cases} \frac{d\omega(t)}{dt} = \vec{\alpha}(t, \vec{v}_c(t), \vec{r}_c(t), \vec{\omega}(t), \vec{\Omega}(t)) \\ 0 \\ -\omega_z(t) \\ \omega_y(t) \\ \frac{d\vec{\Omega}(t)}{dt} = \vec{\omega}(t, \vec{v}_c(t), \vec{r}_c(t), \vec{\omega}(t), \vec{\Omega}(t)) \\ \omega_z(t) \\ 0 \\ -\omega_x(t) \\ \omega_y(t) \\ 0 \end{cases}$

4 équations différentielles vectorielles couplées.
 On doit résoudre alors 18 équations différentielles couplées.
 (12 équations au total).

Introduction

Introduction

Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

On peut aussi utiliser les quaternions de rotation $\vec{\vec{R}}$ pour obtenir

$$\frac{d\vec{\vec{R}}(t)}{dt} = \frac{1}{2} \vec{\vec{R}}(t) \vec{\omega}(t)$$

Notre système implique maintenant 13 équations différentielles couplées (une de plus que pour la position angulaire).

Introduction

Introduction

Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

- Ces équations sont incomplètes et pour obtenir une solution finale, il faut aussi fournir des conditions initiales correspondant à $\vec{r}_c(t_0)$, $\vec{v}_c(t_0)$, $\vec{\omega}(t_0)$ et $\mathbf{R}(t_0)$.
- On doit aussi fournir $\vec{r}_{i,c}(t_0)$, la position de chaque point du solide par rapport à $\vec{r}_c(t_0)$.
- Si le solide possède un mouvement de translation et de rotation, il faut combiner les solutions linéaires et angulaires pour obtenir

$$\vec{r}_i(t) = \vec{r}_c(t) + \mathbf{R}(t)\vec{r}_{i,c}(t_0)$$

$$\vec{v}_i(t) = \vec{v}_c(t) + \vec{\omega}(t) \times (\mathbf{R}(t)\vec{r}_{i,c}(t_0))$$

Résolution des équations de la cinématique des particules

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Pour les particules et le centre de masse des solides, les équations à résoudre sont donc (pour simplifier, on néglige l'indice c) :

$$\vec{a}(t) = d\vec{v}(t)/dt$$

$$\vec{v}(t) = d\vec{r}(t)/dt$$

Cas particulier: Accélération ne dépend que du temps

la solution au temps t peut être obtenue par intégration directe (les six équations différentielles sont découplées)

$$\vec{v}(t) = \vec{v}(t_0) + \int_{t_0}^t \vec{a}(t') dt'$$

$$\vec{r}(t) = \vec{r}(t_0) + \int_{t_0}^t \vec{v}(t') dt'$$

avec $\vec{v}(t_0)$ et $\vec{r}(t_0)$ la vitesse et position initiale du point.

Résolution des équations de la cinématique des particules

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Deux cas peuvent alors se présenter :

- les fonctions correspondant à $a_x(t)$, $a_y(t)$ et $a_z(t)$ sont **faciles à intégrer analytiquement:**

On commence alors par obtenir $\vec{v}(t)$ que l'on intègre par la suite pour obtenir $\vec{r}(t)$;

- les fonctions correspondant à $a_x(t)$, $a_y(t)$ et $a_z(t)$ sont **difficiles ou impossibles à intégrer analytiquement:**

On utilise des méthodes numériques qui permettront d'approximer la solution.

Résolution des équations de la cinématique des particules

Introduction
 Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Expansion en série limitée de la solution s'il est possible

d'évaluer analytiquement $\frac{d^m \vec{a}(t)}{dt^m}$

On combine l'expansion en série pour $\vec{a}(t)$ et une intégration par intervalle de temps.

- On sépare l'intervalle de temps $\Delta t = t_i - t_f$ en N sous intervalles de temps de largeur $\Delta t_n = t_n - t_{n-1}$ avec $n = 1, \dots, N$
 $\Rightarrow t_i = t_0, t = t_n$ et $t_f = t_N$
- On utilise une expansion en série pour $\vec{a}(t')$ sur chaque intervalle de temps $t_{n-1} < t' < t_n$

$$\vec{a}(t' - t_{n-1}) \approx \vec{a}(t_{n-1}) + \left. \frac{d\vec{a}(t'')}{dt''} \right|_{t''=t_{n-1}} (t' - t_{n-1}) + \left. \frac{d^2\vec{a}(t'')}{dt''^2} \right|_{t''=t_{n-1}} \frac{(t' - t_{n-1})^2}{2} + \dots$$

Résolution des équations de la cinématique des particules

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

On obtient alors facilement la vitesse à $t \leq t_n$, si on connaît $\vec{a}(t')$ et ses dérivées au temps t_{n-1}

$$\vec{v}(t) = \vec{v}(t_{n-1}) + \int_{t_{n-1}}^t [\vec{a}(t' - t_{n-1})] dt'$$

$$\approx \vec{v}(t_{n-1}) + \vec{a}(t_{n-1})(t - t_{n-1}) + \left. \frac{d\vec{a}(t'')}{dt''} \right|_{t''=t_{n-1}} \frac{(t - t_{n-1})^2}{2}$$

$$+ \left. \frac{d^2 \vec{a}(t'')}{dt''^2} \right|_{t''=t_{n-1}} \frac{(t - t_{n-1})^3}{6} + \dots$$

On peut ensuite obtenir $\vec{r}(t)$ en intégrant $\vec{v}(t)$.

$$\vec{a}(t' - t_{n-1})$$

$$\approx \vec{a}(t_{n-1}) + \left. \frac{d\vec{a}(t'')}{dt''} \right|_{t''=t_{n-1}} (t' - t_{n-1}) + \left. \frac{d^2 \vec{a}(t'')}{dt''^2} \right|_{t''=t_{n-1}} \frac{(t' - t_{n-1})^2}{2} + \dots$$

Résolution des équations de la cinématique des particules

Introduction
 Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Expansion en série limitée de la solution où $\frac{d^m \vec{a}(t)}{dt^m}$ est difficile ou impossible à évaluer analytiquement

On limitera l'expansion en série au terme de première dérivée.

On évalue ensuite numériquement $d\vec{a}(t'')/dt''$ en utilisant l'approximation

$$\left. \frac{d\vec{a}(t'')}{dt''} \right|_{t''=t_{n-1}} \approx \frac{\vec{a}(t_n) - \vec{a}(t_{n-1})}{t_n - t_{n-1}} = \vec{k}_n$$

On obtient alors

$$\vec{v}(t_n) \approx \vec{v}(t_{n-1}) + \vec{a}(t_{n-1})\Delta t_n + \frac{\vec{k}_n(\Delta t_n)^2}{2}$$

Une fois $\vec{v}(t_n)$ connue pour tous les instants t_n ($n = 1, \dots, N$), on répète le processus pour l'équation différentielle de la position et on détermine $\vec{r}(t_n)$.

Résolution des équations de la cinématique des particules

Introduction
 Résolution des
 équations de la
 cinématique des
 particules

Résolution des
 équations de la
 cinématique des
 solides

Résolution
 numérique d'EDO

Solutions numériques
 et rotations

Solutions numériques
 et collisions

Conclusion

Si on choisit des intervalles égaux de largeur Δt on aura alors

$$\vec{v}(t_N) \approx \vec{v}(t_0) + \sum_{n=1}^N \left[\vec{a}(t_{n-1}) + \frac{\vec{k}_n}{2} \Delta t \right] \Delta t$$

On peut aussi utiliser une relation semblable pour la position

$$\vec{r}(t_N) \approx \vec{r}(t_0) + \sum_{n=1}^N \left[\vec{v}(t_{n-1}) + \frac{\vec{l}_n}{2} \Delta t \right] \Delta t$$

avec

$$\vec{l}_n = \frac{\vec{v}(t_n) - \vec{v}(t_{n-1})}{\Delta t} = \vec{a}(t_{n-1}) + \frac{\vec{k}_n}{2} \Delta t$$

Résolution des équations de la cinématique des particules

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO
Solutions numériques et rotations
Solutions numériques et collisions

Conclusion

Pour la position, on peut aussi utiliser le fait que dans l'intervalle n , $\vec{v}(t')$ est donné par

$$\vec{v}(t') = \vec{v}(t_{n-1}) + \vec{a}(t_{n-1})(t' - t_{n-1}) + \vec{k}_n \frac{(t' - t_{n-1})^2}{2}$$

pour obtenir une solution plus précise (on a supposé que $\vec{v}(t')$ avait une dépendance quadratique sur t' au lieu de linéaire):

$$\vec{r}(t_n) = \vec{r}(t_{n-1}) + \vec{v}(t_{n-1})\Delta t + \vec{a}(t_{n-1})\frac{(\Delta t)^2}{2} + \vec{k}_n \frac{(\Delta t)^3}{6}$$

Donc

$$\vec{r}(t_N) = \vec{r}(t_0) + \sum_{n=1}^N \Delta t \left[\vec{v}(t_{n-1}) + \Delta t \left(\frac{\vec{a}(t_{n-1})}{2} + \Delta t \frac{\vec{k}_n}{6} \right) \right]$$

Introduction
 Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Particule avec accélération dépendante du temps.

Une particule de vitesse initiale $\vec{v} = (2, 1, 0)^T$ m/s localisée initialement au point $\vec{r} = (0, 0, 1)^T$ m subit une accélération $\vec{a} = (0, \cos(t), \sin(t))^T$ m/s². Déterminer la vitesse et la position de cette particule en fonction du temps t .

La solution exacte est

$$\vec{v}(t) = \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ \sin(t) - \sin(t_0) \\ -\cos(t) + \cos(t_0) \end{pmatrix}$$

$$\vec{r}(t) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix}(t - t_0) + \begin{pmatrix} 0 \\ -\cos(t) + \cos(t_0) - \sin(t_0)(t - t_0) \\ -\sin(t) + \sin(t_0) + \cos(t_0)(t - t_0) \end{pmatrix}$$

Résolution des équations de la cinématique des particules

Introduction

Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

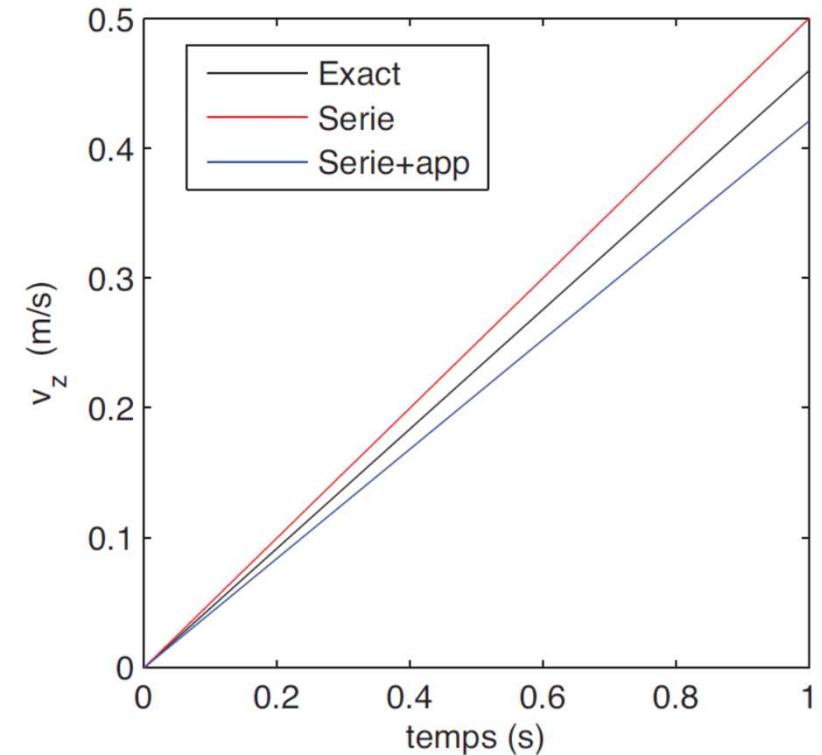
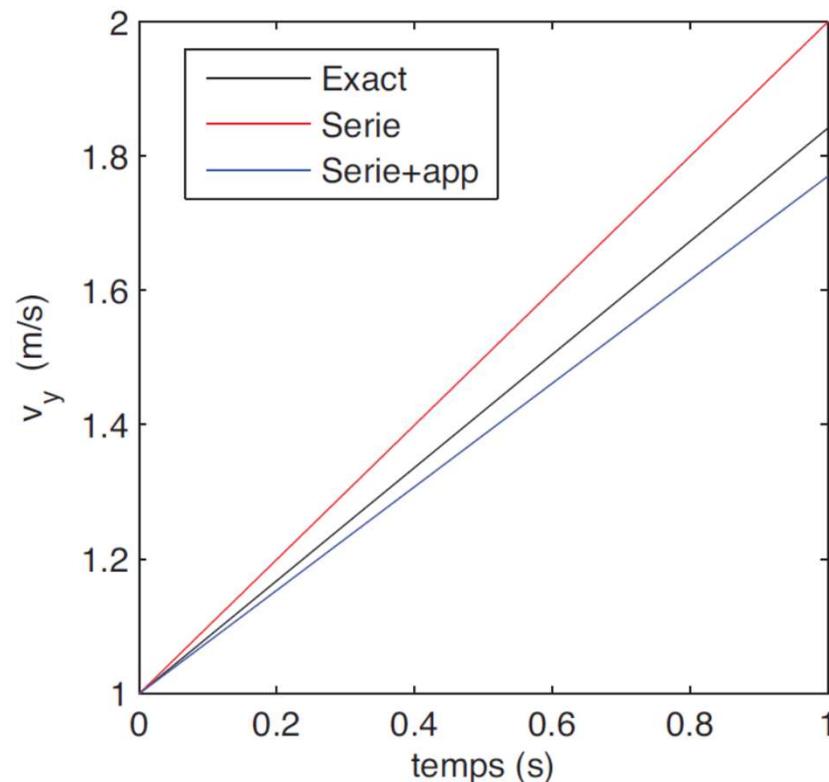
Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

$$\Delta t = 1s$$



Résolution des équations de la cinématique des particules

Introduction

Résolution des équations de la cinématique des particules

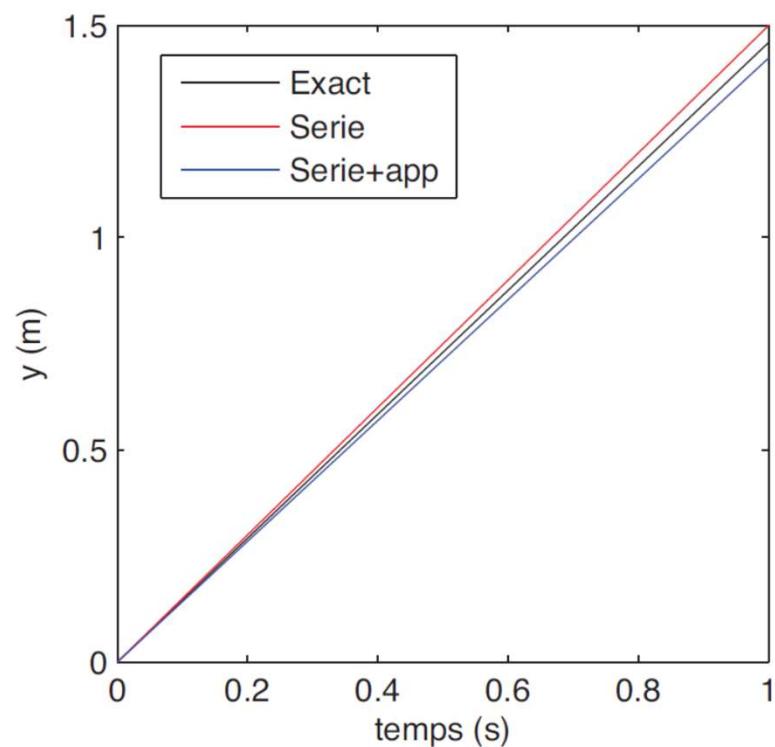
Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

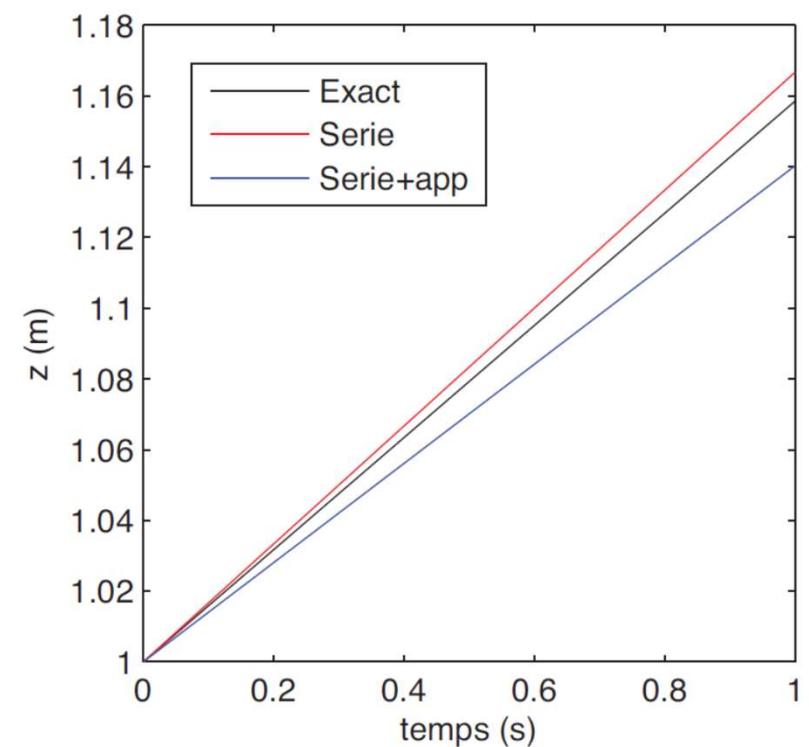
Solutions numériques et rotations

Solutions numériques et collisions

Conclusion



$\Delta t = 1s$



Résolution des équations de la cinématique des particules

Introduction
 Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

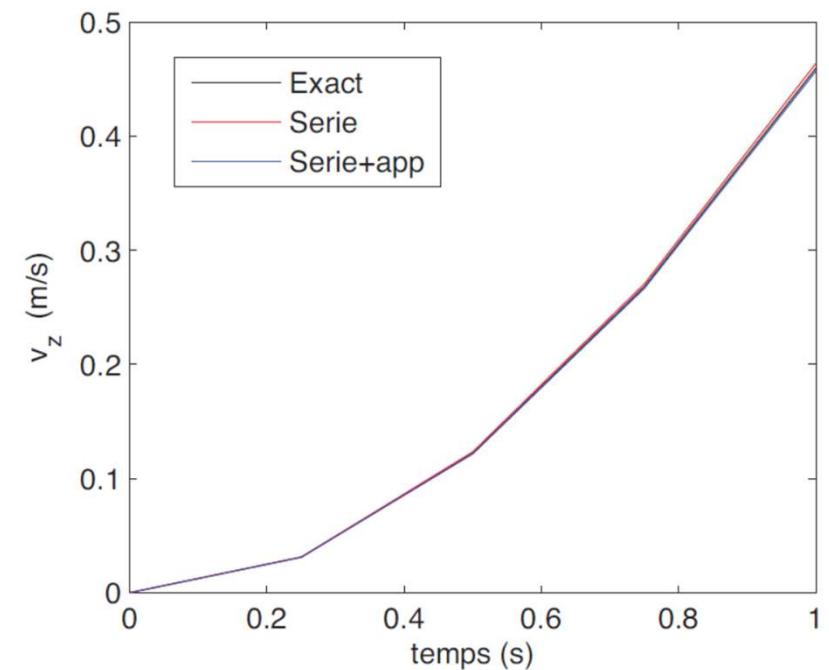
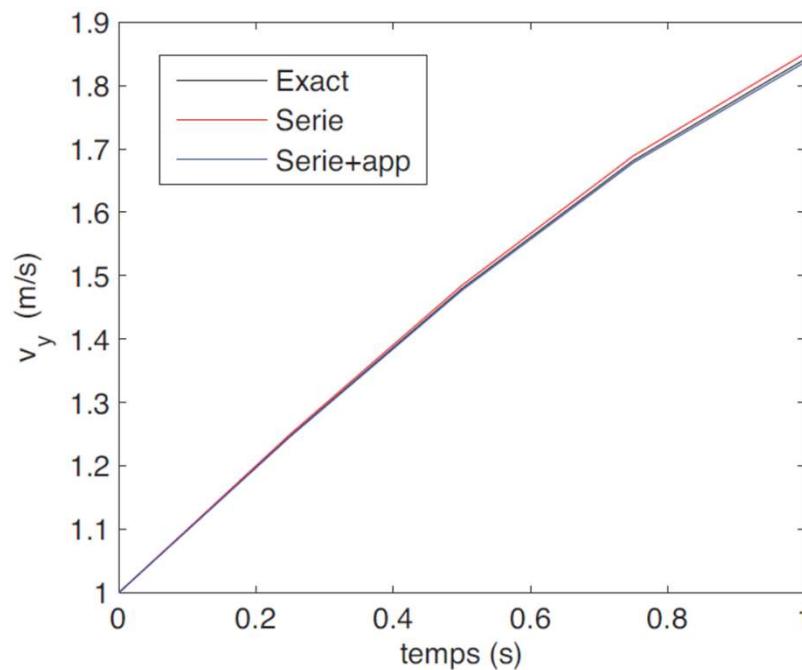
Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

$$\Delta t = 0,25s$$



Résolution des équations de la cinématique des particules

Un exemple sera présenté en classe.

Introduction

Résolution des
équations de la
cinématique des
particules

Résolution des
équations de la
cinématique des
solides

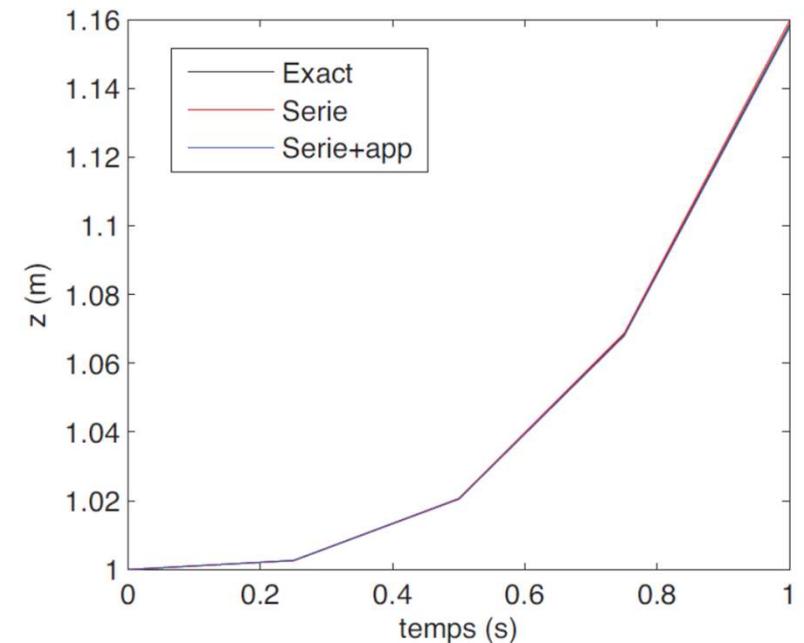
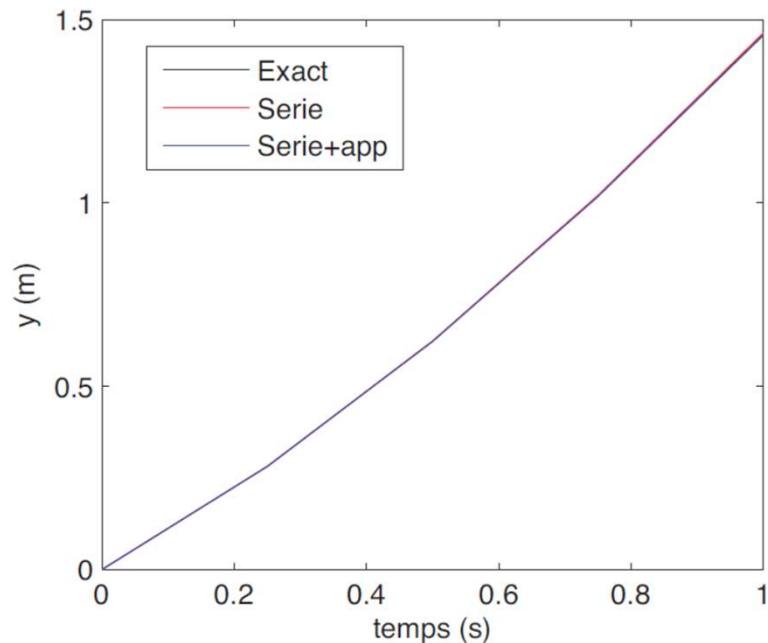
Résolution
numérique d'EDO

Solutions numériques
et rotations

Solutions numériques
et collisions

Conclusion

$$\Delta t = 0,25s$$



Résolution des équations de la cinématique des particules

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Si l'accélération dépend de la vitesse et de la position:

- Le problème devient beaucoup plus difficile à résoudre.
- Dans la majorité des cas, il faut résoudre 6 équations différentielles couplées et des solutions analytiques ne peuvent être générées que pour un nombre de cas extrêmement limité.
- Pour toutes les autres situations, seules des solutions numériques sont possibles.

Résolution des équations de la cinématique des particules

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Exemple d'une particule avec accélération dépendante de la vitesse.

$$a(t) = -\frac{v^2(t)}{k}$$

avec $k = 2 \text{ m}$. La vitesse initiale de la particule est $v(t_0 = 0) = v_{x,0} = 1 \text{ m/s}$ et sa position initiale est nulle ($x(t_0 = 0) = x_0 = 0 \text{ m}$). Il s'agit ici de déterminer la trajectoire de cette particule.

La solution pour la vitesse est

$$v(t) = v_0 \left[\frac{k}{k + v_0 t} \right]$$

Résolution des équations de la cinématique des particules

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

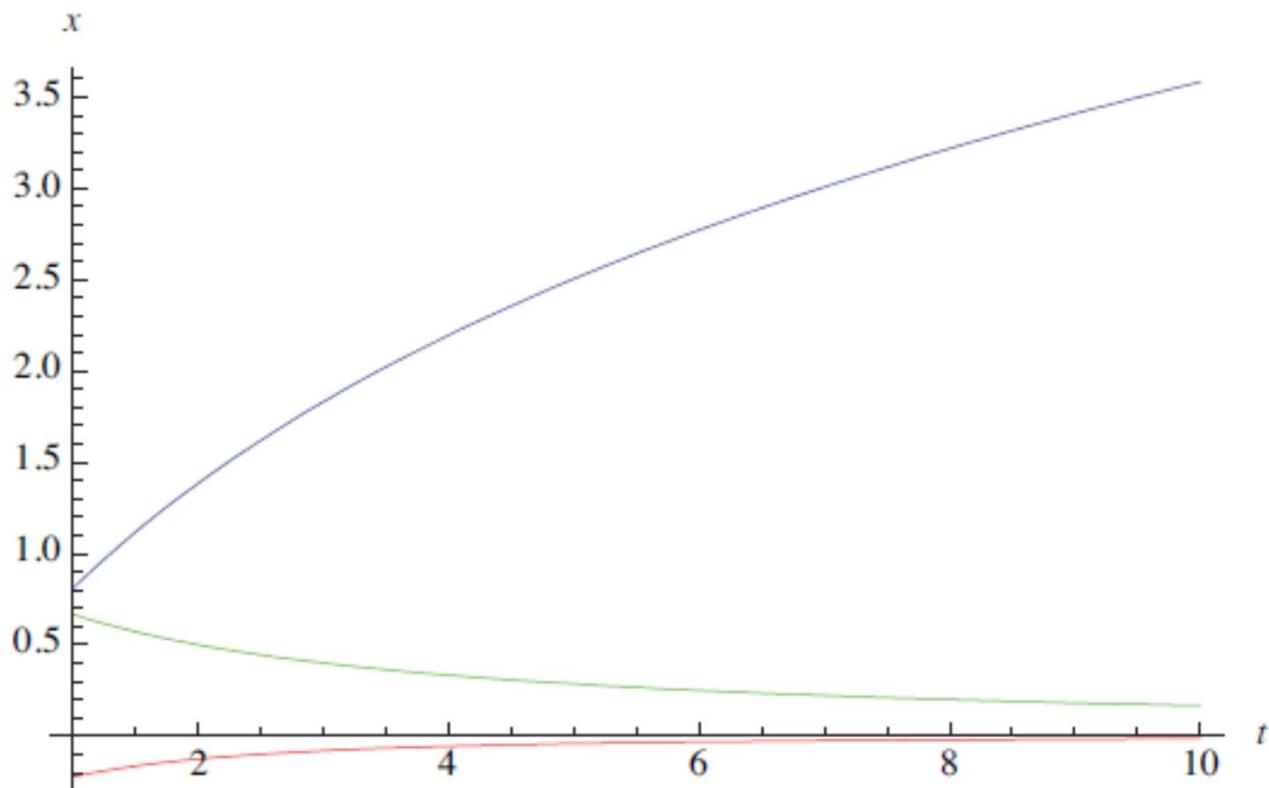
Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Comportement de la particule pour les 10 premières secondes de simulation.



(position en bleu, vitesse en vert, accélération en rouge)

Résolution des équations de la cinématique des solides

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO
Solutions numériques et rotations
Solutions numériques et collisions
Conclusion

Mouvement de rotation

Solides étendus

Les équations de translation → s'appliquent au centre de masse

Ajouter:

$$\frac{d\vec{\omega}(t)}{dt} = \vec{\alpha}(t, \vec{v}, \vec{r}, \vec{\omega}(t), \mathbf{R}(t))$$

$$\frac{d\mathbf{R}(t)}{dt} = \tilde{\omega}(t)\mathbf{R}(t)$$

Résolution des équations de la cinématique des solides

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO
Solutions numériques et rotations
Solutions numériques et collisions
Conclusion

$$\frac{d\omega_x(t)}{dt} = \alpha_x(\vec{\omega}(t), \mathbf{R}(t), t)$$

$$\frac{d\omega_y(t)}{dt} = \alpha_y(\vec{\omega}(t), \mathbf{R}(t), t)$$

$$\frac{d\omega_z(t)}{dt} = \alpha_z(\vec{\omega}(t), \mathbf{R}(t), t)$$

$$\frac{dR_{xx}(t)}{dt} = \omega_y(t)R_{zx}(t) - \omega_z(t)R_{yx}(t)$$

$$\frac{dR_{xy}(t)}{dt} = \omega_y(t)R_{zy}(t) - \omega_z(t)R_{yy}(t)$$

$$\frac{dR_{xz}(t)}{dt} = \omega_y(t)R_{zz}(t) - \omega_z(t)R_{yz}(t)$$

$$\frac{dR_{yx}(t)}{dt} = \omega_z(t)R_{xx}(t) - \omega_x(t)R_{zx}(t)$$

$$\frac{dR_{yy}(t)}{dt} = \omega_z(t)R_{xy}(t) - \omega_x(t)R_{zy}(t)$$

$$\frac{dR_{yz}(t)}{dt} = \omega_z(t)R_{xz}(t) - \omega_x(t)R_{zz}(t)$$

$$\frac{dR_{zx}(t)}{dt} = \omega_x(t)R_{yx}(t) - \omega_y(t)R_{xx}(t)$$

$$\frac{dR_{zy}(t)}{dt} = \omega_x(t)R_{yy}(t) - \omega_y(t)R_{xy}(t)$$

$$\frac{dR_{zz}(t)}{dt} = \omega_x(t)R_{yz}(t) - \omega_y(t)R_{xz}(t)$$

Résolution des équations de la cinématique des solides

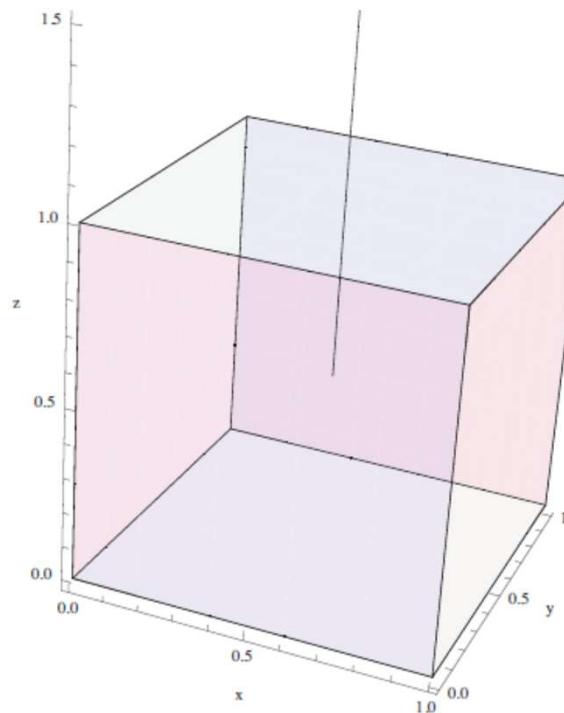
Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO
Solutions numériques et rotations
Solutions numériques et collisions
Conclusion

Exemple: Rotations dans un plan 2-D.

Pour un solide en rotation dans un plan 2-D (plan $x - y$), le mouvement de rotation doit nécessairement se faire autour du troisième axe (z).



Résolution des équations de la cinématique des solides

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO
Solutions numériques et rotations
Solutions numériques et collisions
Conclusion

Ici on supposera : $\vec{\alpha}(t) = \vec{0}$ et $\vec{\omega}(t_0) = \begin{pmatrix} 0 \\ 0 \\ \omega_z \end{pmatrix}$

- Le solide tourne donc initialement autour de l'axe des z avec une vitesse angulaire ω_z rad/s.
- On peut alors découpler l'équation différentielle pour la vitesse angulaire et celle pour la matrice de rotation. On aura donc pour la vitesse angulaire

$$\frac{d\vec{\omega}(t)}{dt} = 0$$

La solution est alors $\vec{\omega}(t) = \begin{pmatrix} 0 \\ 0 \\ \omega_z \end{pmatrix}$

Résolution des équations de la cinématique des solides

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO
Solutions numériques et rotations
Solutions numériques et collisions
Conclusion

Examinons maintenant l'équation différentielle pour la matrice de rotation

$$\frac{d\mathbf{R}(t)}{dt} = \tilde{\boldsymbol{\omega}}(t)\mathbf{R}(t)$$

Le produit $\tilde{\boldsymbol{\omega}}(t)\mathbf{R}(t)$ devient

$$\tilde{\boldsymbol{\omega}}(t)\mathbf{R}(t) = \begin{pmatrix} -\omega_z(t)R_{yx}(t) & -\omega_z(t)R_{yy}(t) & -\omega_z(t)R_{yz}(t) \\ \omega_z(t)R_{xx}(t) & \omega_z(t)R_{xy}(t) & \omega_z(t)R_{xz}(t) \\ 0 & 0 & 0 \end{pmatrix}$$

La condition initiale sera

$$\mathbf{R}(t_0) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Résolution des équations de la cinématique des solides

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO
Solutions numériques et rotations
Solutions numériques et collisions

Conclusion

On aura donc les équations suivantes à résoudre

$$dR_{xx}(t)/dt = -\omega_z R_{yx}(t)$$

avec $R_{xx}(t_0) = 1$

$$dR_{xy}(t)/dt = -\omega_z R_{yy}(t)$$

avec $R_{xy}(t_0) = 0$

$$dR_{xz}(t)/dt = -\omega_z R_{yz}(t)$$

avec $R_{xz}(t_0) = 0$

$$dR_{yx}(t)/dt = \omega_z R_{xx}(t)$$

avec $R_{yx}(t_0) = 0$

$$dR_{yy}(t)/dt = \omega_z R_{xy}(t)$$

avec $R_{yy}(t_0) = 1$

$$dR_{yz}(t)/dt = \omega_z R_{xz}(t)$$

avec $R_{yz}(t_0) = 0$

$$dR_{zx}(t)/dt = 0$$

avec $R_{zx}(t_0) = 0$

$$dR_{zy}(t)/dt = 0$$

avec $R_{zy}(t_0) = 0$

$$dR_{zz}(t)/dt = 0$$

avec $R_{zz}(t_0) = 1$

Résolution des équations de la cinématique des solides

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO
Solutions numériques et rotations
Solutions numériques et collisions

Conclusion

On aura donc les équations suivantes à résoudre

$$dR_{xx}(t)/dt = -\omega_z R_{yx}(t)$$

$$R_{xx}(t_0) = 1$$

$$\frac{d^2 R_{xx}(t)}{dt^2} = -\omega_z \frac{dR_{yx}(t)}{dt}$$

$$dR_{yx}(t)/dt = \omega_z R_{xx}(t)$$

$$R_{yx}(t_0) = 0$$

$$\frac{d^2 R_{xx}(t)}{dt^2} = -\omega_z^2 R_{xx}(t)$$

$$R_{xx}(t) = \cos(\omega_z t)$$

Pour $t_0 = 0$

$$R_{yx}(t) = \sin(\omega_z t)$$

De la même façon: $R_{yy}(t) = \cos(\omega_z t)$

$$R_{yx}(t) = -\sin(\omega_z t)$$

Résolution des équations de la cinématique des solides

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO
Solutions numériques et rotations
Solutions numériques et collisions
Conclusion

La solution est alors

$$\boldsymbol{R}(t) = \begin{pmatrix} \cos(\omega_z t) & -\sin(\omega_z t) & 0 \\ \sin(\omega_z t) & \cos(\omega_z t) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

qui est identique à celle qu'on aurait obtenue en utilisant

$$d\vec{\Omega}(t)/dt = \vec{\omega}(t) = (0, 0, \omega_z)^T$$

qui est

$$\vec{\Omega}(t) = \begin{pmatrix} 0 \\ 0 \\ \omega_z t \end{pmatrix}$$

La matrice correspondant à une rotation de $\omega_z t$ autour de l'axe des z étant bien $\boldsymbol{R}(t)$ donnée ci-dessus.

Résolution numérique d'EDO

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Résolution numérique d'équations différentielles ordinaires.

Nous présenterons ici deux techniques communément utilisées pour la résolution numérique des équations différentielles.

1. La méthode d'Euler qui est de loin la plus simple, même si elle est relativement peu utilisée en physique du fait de sa faible précision (erreur d'ordre 1).
2. La méthode de Runge-Kutta d'ordre 4 qui est un plus complexe à programmer, mais qui a une précision de beaucoup supérieure (erreur d'ordre 4). Il existe aussi une version simple à l'ordre 2 qui est un peu moins précise.

Résolution numérique d'EDO

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Les équations différentielles générales que nous devrons traiter ont toutes la forme

$$\frac{d\vec{q}(t)}{dt} = \vec{g}(\vec{q}(t), t)$$

avec des conditions initiales

$$\vec{q}(t_0) = \vec{q}_0$$

Résolution numérique d'EDO

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Dans le cas des équations de la cinématique du point (centre de masse pour les solides):

$$\vec{q}(t) = \begin{pmatrix} q_1(t) \\ q_2(t) \\ q_3(t) \\ q_4(t) \\ q_5(t) \\ q_6(t) \end{pmatrix} = \left\{ \begin{pmatrix} v_{c,x}(t) \\ v_{c,y}(t) \\ v_{c,z}(t) \\ x_c(t) \\ y_c(t) \\ z_c(t) \end{pmatrix} \right\} \vec{v}_c(t) \quad \vec{g}(\vec{q}, t) = \left\{ \begin{pmatrix} a_{c,x}(\vec{q}, t) \\ a_{c,y}(\vec{q}, t) \\ a_{c,z}(\vec{q}, t) \\ q_1(t) \\ q_2(t) \\ q_3(t) \end{pmatrix} \right\} \vec{\ddot{r}}_c(t) \quad \vec{a}_c(t) = \left\{ \begin{pmatrix} a_{c,x}(\vec{q}, t) \\ a_{c,y}(\vec{q}, t) \\ a_{c,z}(\vec{q}, t) \end{pmatrix} \right\} \vec{\ddot{v}}_c(t)$$

$$\frac{d\vec{q}(t)}{dt} = \vec{g}(\vec{q}(t), t)$$

$$\vec{q}(t_0) = \vec{q}_0$$

Résolution des équations de la cinématique des solides

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Pour les solides, la forme générale de $\vec{q}(t)$ et $\vec{g}[\vec{q}(t), t]$ est :

$$\vec{q}(t) = \begin{pmatrix} q_1(t) \\ q_2(t) \\ q_3(t) \\ q_4(t) \\ q_5(t) \\ q_6(t) \\ q_7(t) \\ q_8(t) \\ q_9(t) \\ q_{10}(t) \\ q_{11}(t) \\ q_{12}(t) \\ q_{13}(t) \\ q_{14}(t) \\ q_{15}(t) \\ q_{16}(t) \\ q_{17}(t) \\ q_{18}(t) \end{pmatrix} = \begin{pmatrix} v_{c,x}(t) \\ v_{c,y}(t) \\ v_{c,z}(t) \\ x_c(t) \\ y_c(t) \\ z_c(t) \\ \omega_x(t) \\ \omega_y(t) \\ \omega_z(t) \\ R_{xx}(t) \\ R_{xy}(t) \\ R_{xz}(t) \\ R_{yx}(t) \\ R_{yy}(t) \\ R_{yz}(t) \\ R_{zx}(t) \\ R_{zy}(t) \\ R_{zz}(t) \end{pmatrix}$$

$\vec{v}_c(t)$

$\vec{r}_c(t)$

$\vec{\omega}(t)$

$R(t)$

$\tilde{\omega}R(t)$

$$\vec{g}(\vec{q}, t) = \begin{pmatrix} \vec{a}_c(t) \\ \vec{v}_c(t) \\ \vec{\alpha}(t) \\ \tilde{\omega}R(t) \end{pmatrix} = \begin{pmatrix} a_{c,x}(\vec{q}, t) \\ a_{c,y}(\vec{q}, t) \\ a_{c,z}(\vec{q}, t) \\ q_1(t) \\ q_2(t) \\ q_3(t) \\ \alpha_x(\vec{q}, t) \\ \alpha_y(\vec{q}, t) \\ \alpha_z(\vec{q}, t) \\ q_8(t)q_{16}(t) - q_9(t)q_{13}(t) \\ q_8(t)q_{17}(t) - q_9(t)q_{14}(t) \\ q_8(t)q_{18}(t) - q_9(t)q_{15}(t) \\ q_9(t)q_{10}(t) - q_7(t)q_{16}(t) \\ q_9(t)q_{11}(t) - q_7(t)q_{17}(t) \\ q_9(t)q_{12}(t) - q_7(t)q_{18}(t) \\ q_7(t)q_{13}(t) - q_8(t)q_{10}(t) \\ q_7(t)q_{14}(t) - q_8(t)q_{11}(t) \\ q_7(t)q_{15}(t) - q_8(t)q_{12}(t) \end{pmatrix}$$

Résolution des équations de la cinématique des solides

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Si on décide d'utiliser les quaternions de rotation au lieu des matrices de rotation, on utilisera

$$\vec{q}(t) = \begin{pmatrix} v_{c,x}(t) \\ v_{c,y}(t) \\ v_{c,z}(t) \\ x_c(t) \\ y_c(t) \\ z_c(t) \\ \omega_x(t) \\ \omega_y(t) \\ \omega_z(t) \\ \vec{R}_0(t) \\ \vec{R}_1(t) \\ \vec{R}_2(t) \\ \vec{R}_3(t) \end{pmatrix} \quad \vec{g}(\vec{q}, t) = \begin{pmatrix} a_{c,x}(\vec{q}, t) \\ a_{c,y}(\vec{q}, t) \\ a_{c,z}(\vec{q}, t) \\ q_1(t) \\ q_2(t) \\ q_3(t) \\ \alpha_x(\vec{q}, t) \\ \alpha_y(\vec{q}, t) \\ \alpha_z(\vec{q}, t) \\ (-q_{11}(t)q_7(t) - q_{12}(t)q_8(t) - q_{13}(t)q_9(t))/2 \\ (q_{10}(t)q_7(t) + q_{12}(t)q_9(t) - q_{13}(t)q_8(t))/2 \\ (q_{10}(t)q_8(t) + q_{13}(t)q_7(t) - q_{11}(t)q_9(t))/2 \\ (q_{10}(t)q_9(t) + q_{11}(t)q_8(t) - q_{12}(t)q_7(t))/2 \end{pmatrix}$$

$\frac{1}{2} \vec{R}(t) \vec{\omega}(t)$

Résolution numérique d'EDO

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Méthode d'Euler.

La solution au temps $t > t_0$ est obtenue par étapes en résolvant le problème sur n intervalles de temps consécutifs $t_1 = t_0 + \Delta t$, $t_2 = t_1 + \Delta t$ et ce jusqu'à $t = t_N = t_{N-1} + \Delta t$ avec

$$\Delta t = \frac{(t - t_0)}{N}$$

Résolution numérique d'EDO

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Si l'intervalle de temps Δt est suffisamment petit, on peut supposer que $\vec{g}(\vec{q}(t'), t')$ demeure constant lors de l'évolution de t_{n-1} à t_n (avec $t_n = t_{n-1} + \Delta t$)

$$\vec{g}(\vec{q}(t'), t') = \vec{g}(\vec{q}(t_{n-1}), t_{n-1}) \text{ pour } t_{n-1} \leq t' \leq t_n$$

Comme $\vec{g}(\vec{q}(t'), t')$ est constant sur chaque intervalle, on peut alors facilement intégrer les équations de la cinématique et obtenir

$$\vec{q}(t_n) = \vec{q}(t_{n-1}) + \vec{g}(\vec{q}(t_{n-1}), t_{n-1})\Delta t + O(\Delta t^2)$$

qui est la solution désirée.

Résolution numérique d'EDO

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

$$\vec{q}(t_n) = \vec{q}(t_{n-1}) + \vec{g}(\vec{q}(t_{n-1}), t_{n-1})\Delta t + O(\Delta t^2)$$

$$\vec{q}(t_1) = \vec{q}(t_0) + \vec{g}(\vec{q}(t_0), t_0)\Delta t$$

$$\vec{q}(t_2) = \vec{q}(t_1) + \vec{g}(\vec{q}(t_1), t_1)\Delta t$$

$$\vec{q}(t_3) = \vec{q}(t_2) + \vec{g}(\vec{q}(t_2), t_2)\Delta t$$

$$\begin{array}{ccc} \vdots & \vdots & \vdots \\ \bullet & \bullet & \bullet \end{array}$$

$$\vec{q}(t_n) = \vec{q}(t_{n-1}) + \vec{g}(\vec{q}(t_{n-1}), t_{n-1})\Delta t$$

$$\begin{array}{ccc} \bullet & \bullet & \bullet \\ \vdots & \vdots & \vdots \\ \bullet & \bullet & \bullet \end{array}$$

$$\vec{q}(t_N) = \vec{q}(t_{N-1}) + \vec{g}(\vec{q}(t_{N-1}), t_{N-1})\Delta t$$

Résolution numérique d'EDO

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Remarques:

- La méthode d'Euler consiste à remplacer la dérivée par une différence

$$\frac{d\vec{q}(t')}{dt'} = \frac{\vec{q}(t_n) - \vec{q}(t_{n-1})}{\Delta t}$$

Cette solution correspond au cas approximatif décrit précédemment avec une différence majeure : toutes les équations sont résolues en même temps (au lieu de résoudre pour $\vec{v}(t)$ et ensuite pour $\vec{r}(t)$).

- l'erreur à chaque étape de l'algorithme est proportionnelle à $(\Delta t)^2$;
l'erreur globale après N étapes (égales) est proportionnelle à Δt .

On peut aussi décider de choisir un intervalle de temps Δt_n différent à chaque étape.

Résolution numérique d'EDO

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Algorithme d'Euler en MATLAB.

```
function qs=SEDEulert0(q0,t0,DeltaT,g)
% Solution equations differentielles par
% methode de Euler
% Equation a resoudre : dq/dt=g(q,t)
% avec
%   qs      : solution [q(t0+DeltaT)]
%   q0      : conditions initiales [q(t0)]
%   DeltaT  : intervalle de temps
%   g       : membre de droite de ED.
%           C'est un m-file de matlab
%           qui retourne la valeur de g
%           au temps choisi
qs=q0+feval(g,q0,t0)*DeltaT;
```

Résolution numérique d'EDO

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Fonction MATLAB pour évaluer $g(q,t)$ pour le problème à une dimension décrit plus haut

```
function res=g(q0,t0)
% dvx(t)/dt=-(vx(t))^2/k
% dx(t)/dt=vx(t)
% q0(1)=vx(t0)
% q0(2)=x(t0)
k=2.0;
res=[-q0(1)^2/k q0(1));
```

Résolution numérique d'EDO

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Méthode de Runge-Kutta d'ordre 4.

Dans ce cas, on procède à quatre évaluations successives de $\vec{g}(\vec{q}(t'), t')$

$$\vec{k}_1 = \vec{g}(\vec{q}(t_{n-1}), t_{n-1})$$

$$\vec{k}_2 = \vec{g}\left(\vec{q}(t_{n-1}) + \frac{\Delta t}{2} \vec{k}_1, t_{n-1} + \frac{\Delta t}{2}\right)$$

$$\vec{k}_3 = \vec{g}\left(\vec{q}(t_{n-1}) + \frac{\Delta t}{2} \vec{k}_2, t_{n-1} + \frac{\Delta t}{2}\right)$$

$$\vec{k}_4 = \vec{g}(\vec{q}(t_{n-1}) + \Delta t \vec{k}_3, t_{n-1} + \Delta t)$$

Résolution numérique d'EDO

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

On utilise ensuite

$$\vec{q}(t_n) = \vec{q}(t_{n-1}) + \frac{\Delta t}{6} \left[\vec{k}_1 + 2\vec{k}_2 + 2\vec{k}_3 + \vec{k}_4 \right] + O(\Delta t^5)$$

- L'erreur à chaque étape de l'algorithme est proportionnelle à $(\Delta t)^5$.
- L'erreur globale après n étapes (égales) est proportionnelle à $(\Delta t)^4$.

Cette méthode de résolution est généralement très précise. Une liste des méthodes de Runge-Kutta est disponible sur [Runge-Kutta sur Wikipédia](#).

Résolution numérique d'EDO

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Algorithme Runge-Kutta en MATLAB.

```
function qs=SEDRK4t0(q0,t0,DeltaT,g)
% Solution equations differentielles
% par methode de RK4
% Equation a resoudre : dq/dt=g(q,t)
% avec
%   qs          : solution [q(t0+DeltaT)]
%   q0          : conditions initiales [q(t0)]
%   DeltaT      : intervalle de temps
%   g           : membre de droite de ED.
%                   C'est un m-file de matlab
%                   qui retourne la valeur de g
%                   au temps choisi
k1=feval(g,q0,t0);
k2=feval(g,q0+k1*DeltaT/2,t0+DeltaT/2);
k3=feval(g,q0+k2*DeltaT/2,t0+DeltaT/2);
k4=feval(g,q0+k3*DeltaT,t0+DeltaT);
qs=q0+DeltaT*(k1+2*k2+2*k3+k4)/6;
```

Résolution numérique d'EDO

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Quelques considérations :

- si le but est de déterminer la solution à un temps particulier (quel qu'il soit), il est toujours préférable d'utiliser la solution analytique si elle est disponible ;

- si le but est de tracer la trajectoire d'un solide, il faut de toute façon évaluer la solution à de multiples pas de temps entre t_0 et t . Lorsque la solution analytique est complexe, il est alors souvent plus rapide (en temps de calcul) de déterminer la solution numérique que la solution exacte.

Résolution numérique d'EDO

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Retour sur l'exemple avec accélération dépendante de la vitesse et dont la solution est

$$x(t_i) = k \ln \left[\frac{k + v_0 t_i}{k} \right]$$

Si on désire tracer la trajectoire à tous les pas de temps $\Delta t_0 = 0.01$ s pour $0 < t < 10$ secondes, la solution d'Euler sur chaque pas de temps sera très précise et demandera 2 multiplications, un quotient et 1 additions alors que la solution analytique demandera 2 multiplications, 1 addition, 1 quotient et un logarithme. La solution d'Euler est alors plus rapide tout en étant à peu près aussi précise. De plus, la solution d'Euler inclut déjà la vitesse alors que des calculs additionnels sont requis pour la méthode analytique (2 multiplications, 1 addition et un quotient).

Résolution numérique d'EDO

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Exemple: particule accélérée en 3D

Ce problème est une généralisation de l'exemple avec accélération dépendante de la vitesse pour une trajectoire 3D. On considère des accélérations

$$(a_x(t), a_y(t), a_z(t)) = (-v_x^2(t)/2, -v_y^2(t), -v_z^2(t)/3)$$

La vitesse initiale de la particule est $\vec{v}(t_0) = (1, 1, 1)^T$ m/s et la position initiale est $\vec{r}(t_0) = (0, 0, 0)^T$ m. Il s'agit de tracer la trajectoire de cette particule à toutes les secondes pour les 10 premières secondes.

Résolution numérique d'EDO

Introduction
Résolution des équations de la cinématique des particules
Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations
Solutions numériques et collisions
Conclusion

Solution par algorithme d'Euler.

```
function [qsol]=SolExEuler(tlimits ,nbi)
% Exemple en 3-D.
% Solutions Euler
% Utilise la fonction Matlab : fgEx.m
% De tlimits(1)=t_i à tlimits(2)=t_f
% avec nbi étapes
qsol=zeros(nbi+1,6);
qsol(1,:)=[1 1 1 0 0 0];
DeltaT=(tlimits(2)-tlimits(1))/nbi;
t0=tlimits(1);
% Solution
for i=1:nbi
    qsol(i+1,:)=SEDEuler(t0,qsol(i,:),DeltaT, ...
        'fgEx');
    t0=t0+DeltaT;
end;
```

Solution par algorithme Runge-Kutta.

```

function [qsol]=SoleExRK4(tlimits ,nbi)
% Exemple en 3-D.
% Solutions Runge - Kutta
% Utilise la fonction Matlab : fgEx.m
% De tlimits(1)=t_i à tlimits(1)=t_f
% avec nbi étapes
qsol=zeros(nbi+1 ,6);
qsol(1 ,: )=[1 1 1 0 0 0];
DeltaT=(tlimits(2)-tlimits(1))/nbi;
t0=tlimits(1);
% Solution
for i=1:nbi
    qsol(i+1 ,: )=SEDRK4t0(qsol(i ,: ),t0,DeltaT ,...
        'fgEx ');
    t0=t0+DeltaT;
end;

```

Résolution numérique d'EDO

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Fonction externe (fgEx) pour algorithme d'Euler et de Runge-Kutta en MATLAB.

```
function g=fgEx(q0,t0)
%
% Exemple en 3-D.
% Valeur differente de k dans chaque direction.
% Acceleration et vitesse pour resolution
% approximative des equations de la cinematique
% q0: vecteur initial vitesse , position
% g : vecteur initial accélération , vitesse
%
g=[ -q0(1)*q0(1)/2 -q0(2)*q0(2)/1 -q0(3)*q0(3)/3 ...
     q0(1)           q0(2)           q0(2)      ];
```

Résolution numérique d'EDO

Introduction
Résolution des équations de la cinématique des particules

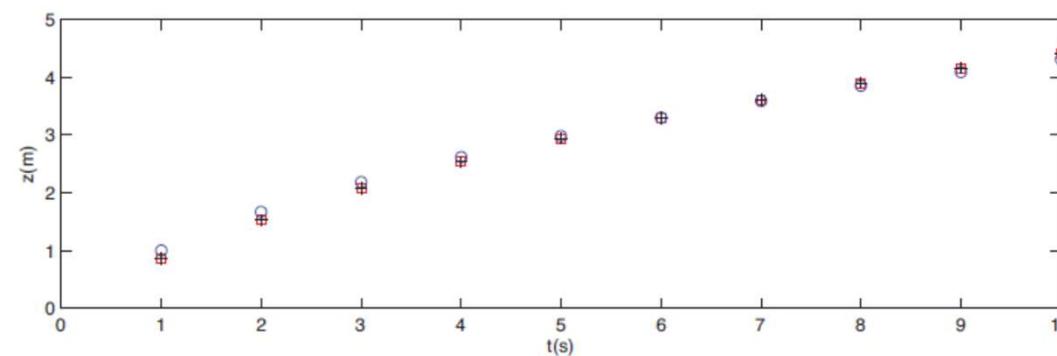
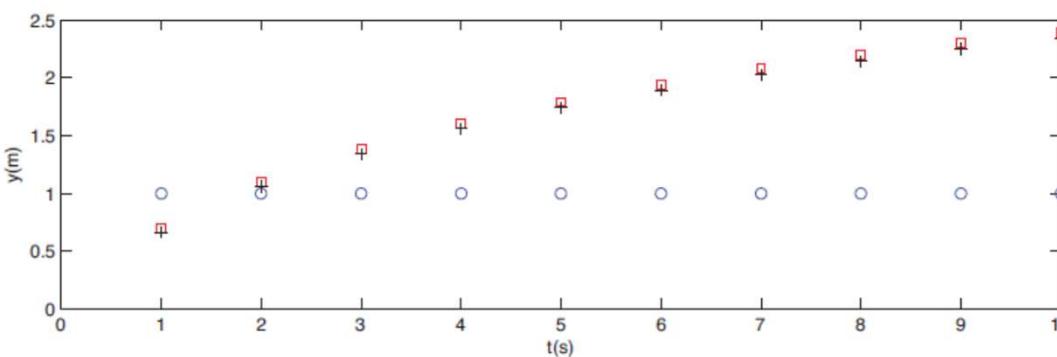
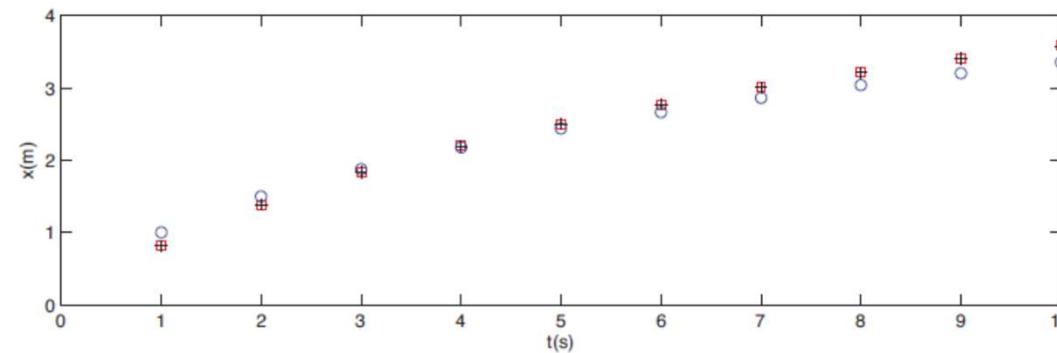
Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion



(exact : rouge, Euler: bleu et RK4 : noir).

$N = 10$ intervalles de temps.

Résolution numérique d'EDO

Introduction
Résolution des équations de la cinématique des particules

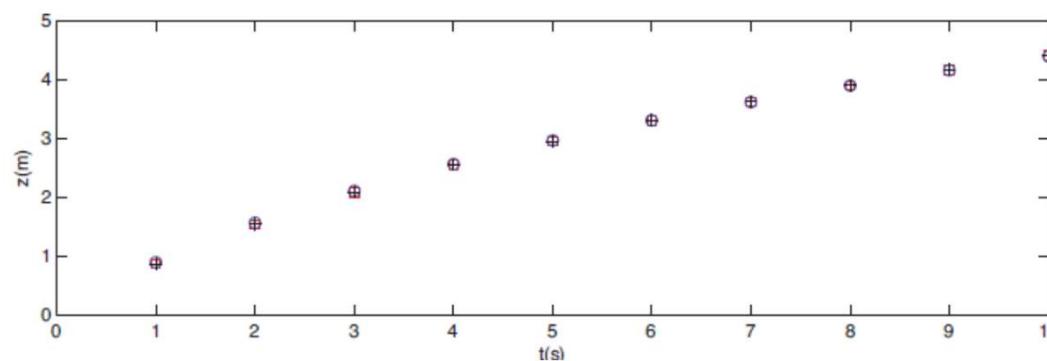
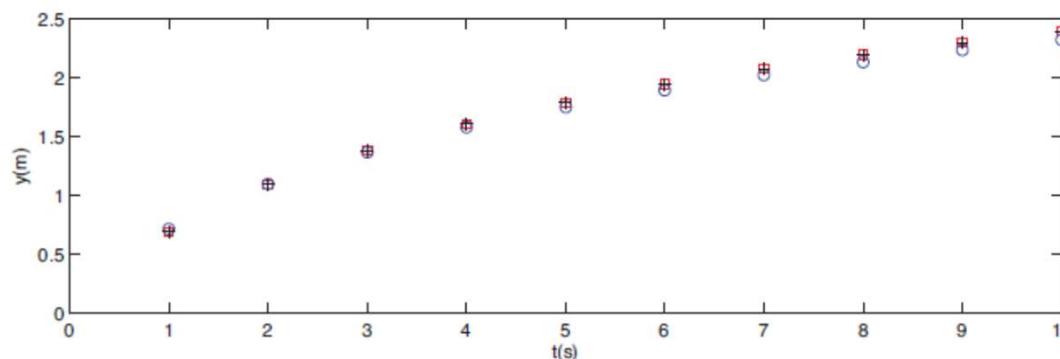
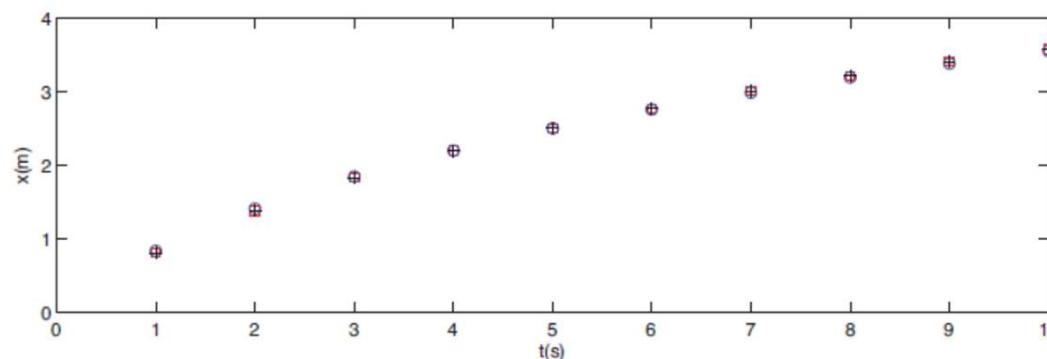
Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion



$N = 40$ intervalles de temps.

Résolution numérique d'EDO

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Solution numérique avec contrôle d'erreur.

Si on désire des solutions numériques à intervalles fixes et qu'on ne veut pas se préoccuper des sous-intervalles de calcul requis par la solution numérique tout en demandant un degré de précision minimal ϵ , il faut alors programmer dans notre routine de résolution numérique un algorithme de contrôle d'erreur.

Résolution numérique d'EDO

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Exemple d'algorithme de contrôle d'erreur simple.

- On spécifie ΔT , le pas de temps associé à la solution finale ($t = t_0 + \Delta T$).
- On spécifie ϵ , la précision absolue de la solution.
- Pour chaque essai m le pas de temps sera

$$\Delta t_m = \frac{\Delta t_{m-1}}{2} = \frac{\Delta T}{2^{m-1}}$$

Résolution numérique d'EDO

Introduction
Résolution des
équations de la
cinématique des
particules

Résolution des
équations de la
cinématique des
solides

Résolution
numérique d'EDO

Solutions numériques
et rotations

Solutions numériques
et collisions

Conclusion

La procédure de contrôle d'erreur est alors (on démarre avec $m = 1$) :

1. résoudre le problème avec Δt_m et d'obtenir $\vec{q}_a(t)$.
2. résoudre le problème avec Δt_{m+1} et d'obtenir $\vec{q}_b(t)$.
3. évaluer le vecteur d'erreur

$$\vec{E}^m = \vec{q}_b(t_f) - \vec{q}_a(t_f).$$

4. si $|E_i^m| > \epsilon_i$ quelque soit i , redéfinir $\vec{q}_a(t) = \vec{q}_b(t)$ et $m = m + 1$
5. sinon $|E_i^m| \leq \epsilon_i$ pour tous les i et la solution désirée est $\vec{q}_b(t)$.

Résolution numérique d'EDO

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Amélioration de la solution (interpolation de Richardson).

Pour la méthode de Runge-Kutta d'ordre 4 on sait que l'erreur globale est d'ordre $(\Delta t)^4$, on supposera donc que

$$\vec{q}(t) = \vec{q}_a(t, \Delta t_m) + \vec{c}(\Delta t_m)^4$$

$$\vec{q}(t) = \vec{q}_b(t, \Delta t_{m+1}) + \vec{c}(\Delta t_{m+1})^4 = \vec{q}_b(t, \Delta t_{m+1}) + \vec{c}(\Delta t_m/2)^4$$

$$\Rightarrow \vec{c} = \frac{16}{15(\Delta t_m)^4} (\vec{q}_b(t, \Delta t_{m+1}) - \vec{q}_a(t, \Delta t_m))$$

Une solution améliorée sera alors donnée par

$$\vec{q}_c(t) = \vec{q}_b(t, \Delta t_{m+1}) + \frac{1}{15} (\vec{q}_b(t, \Delta t_{m+1}) - \vec{q}_a(t, \Delta t_m)) (\Delta t_m)^4$$

Résolution numérique d'EDO

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

On peut aussi utiliser cette information afin de déterminer de façon mieux adaptée le prochain intervalle de temps à considérer.

$$\Delta t < \left(\frac{\epsilon}{|c|} \right)^{\frac{1}{4}}$$

Résolution numérique d'EDO

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Matlab possède aussi des fonctions intrinsèques pour résoudre numériquement les équations différentielles ordinaires :

ode15s, ode23s, ode23t, ode23tb, ode45, ode23, ode113.

Table des matières

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

- Introduction
- Résolution des équations de la cinématique des particules
- Résolution des équations de la cinématique des solides
- Résolution numérique d'EDO
- **Solutions numériques et rotations**
- Solutions numériques et collisions
- Conclusion

Solutions numériques et rotations

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Exemple d'un parallélépipède tournant sur lui même en 3D.

Considérons un parallélépipède de $1.0 \times 2.0 \times 3.0 \text{ m}^3$ dont le centre de masse est initialement $\vec{r}_c = (0, 0, 10)^T \text{ m}$.

1. La vitesse initiale de son centre de masse est de $\vec{v}_c = (1, 0, 0)^T \text{ m/s}$.
2. Il subit une accélération linéaire de $\vec{a} = (0, 0, -1)^T \text{ m/s}^2$.
3. Il subit un mouvement de rotation uniforme sur lui-même autour de l'axe x avec une vitesse angulaire de $\pi/10 \text{ rad/s}$. Il ne subit aucune accélération angulaire.

Tracer la trajectoire du parallélépipède pour une période de $t = 10 \text{ s}$.

Solutions numériques et rotations

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Résolution.

1. Mouvement de centre de masse résolu directement

$$\vec{r}(t) = \begin{pmatrix} 0 + t \\ 0 \\ 10 - t^2/2 \end{pmatrix}$$

2. Pas d'accélération angulaire donc

$$\vec{\omega}(t) = \begin{pmatrix} \pi/10 \\ 0 \\ 0 \end{pmatrix} \text{ rad/s}$$

3. Pour la matrice de rotation ($\mathbf{R}(0) = \mathbf{I}$), on utilise la méthode d'Euler avec intervalle de temps de $\Delta t = 0.1$ s.

On trace le parallélépipède à chaque seconde ($\Delta t = 1.0$ s).

Solutions numériques et rotations

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

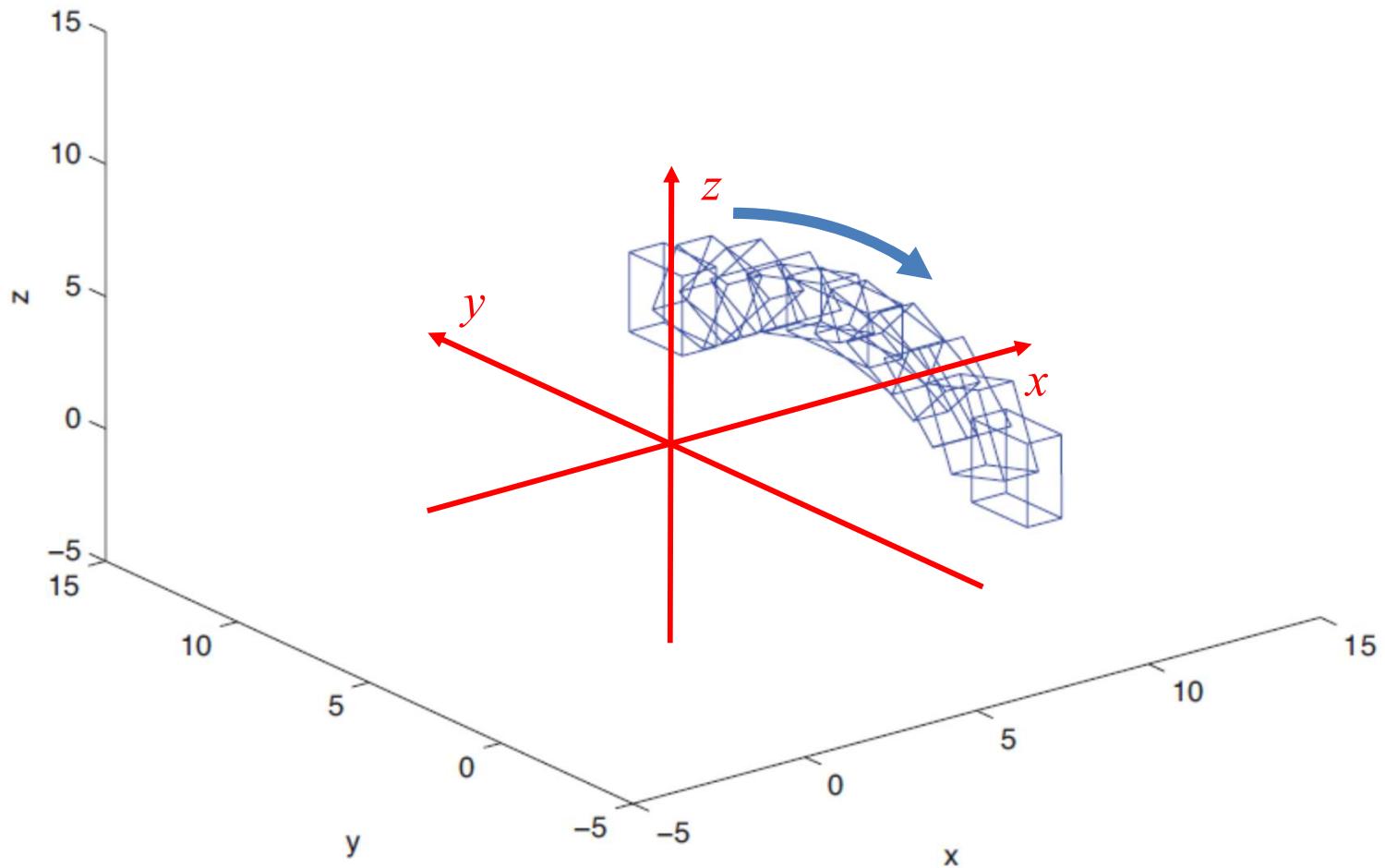
Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

On peut observer le comportement du parallélépipède pour les 10 premières secondes dans la figure suivante (méthode d'Euler avec $\Delta t_E = 0.1$ s).



Solutions numériques et rotations

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

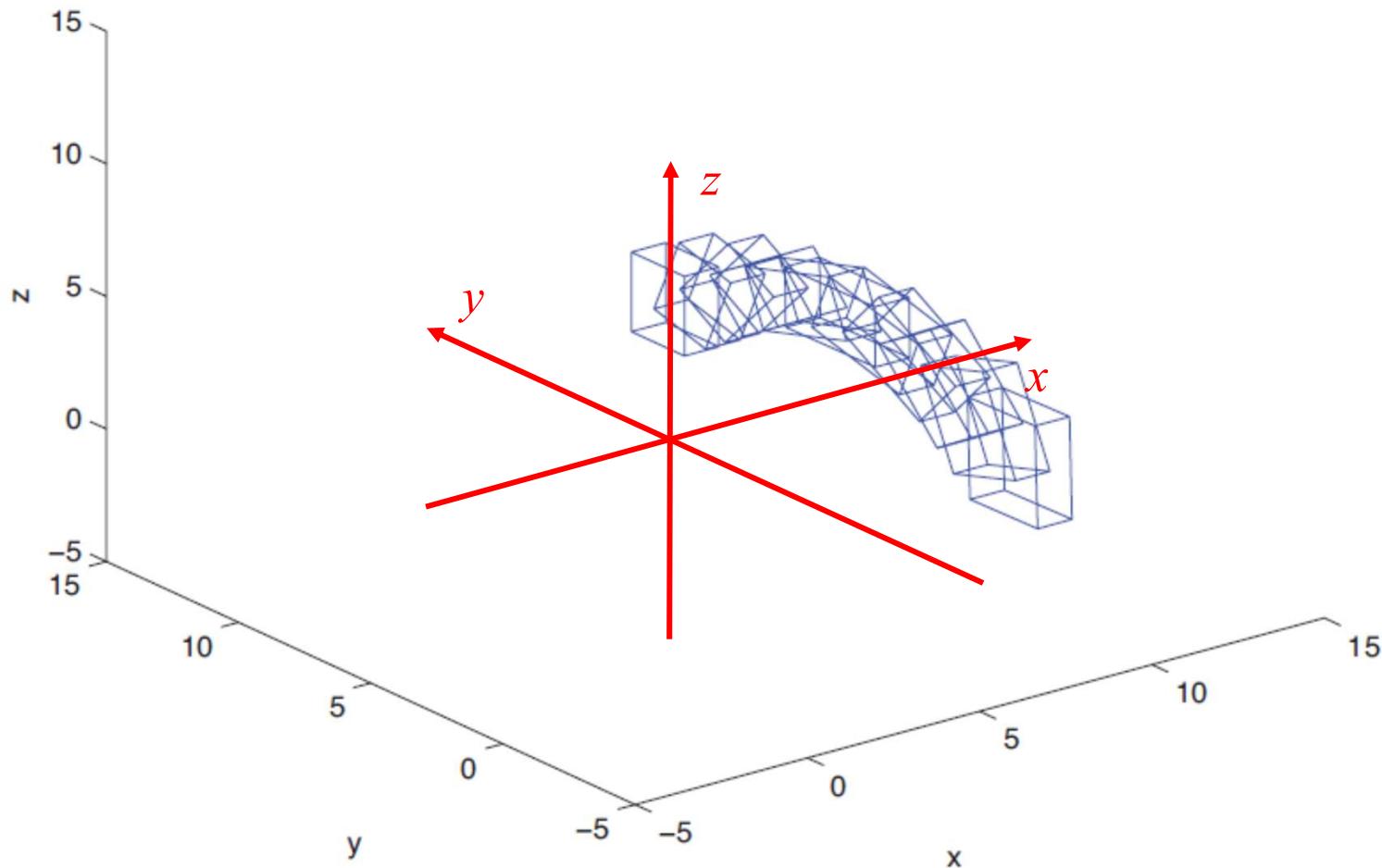
Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Si on utilise $\Delta t_E = 0.5$ s, on obtient.



La trajectoire diffère très peu.

Le parallélépipède change cependant de dimensions (il devient plus grand).

Solutions numériques et rotations

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Solution

$$\mathbf{R}(0.5) = \begin{pmatrix} 1.00000 & 0 & 0 \\ 0 & 1.00000 & -0.15708 \\ 0 & 0.15708 & 1.00000 \end{pmatrix}$$

$$\det(\mathbf{R}(0.5)) = 1.0247$$

$$\mathbf{R}(1.0) = \begin{pmatrix} 1.00000 & 0 & 0 \\ 0 & 0.97533 & -0.31416 \\ 0 & 0.31416 & 0.97533 \end{pmatrix}$$

$$\det(\mathbf{R}(0.5)) = 1.04996$$

Solutions numériques et rotations

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Solution

$$R(0.5) = \begin{pmatrix} 1.00000 & 0 & 0 \\ 0 & 1.00000 & -0.15708 \\ 0 & 0.15708 & 1.00000 \end{pmatrix}$$

$$R(t) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\omega_x t) & -\sin(\omega_x t) \\ 0 & \sin(\omega_x t) & \cos(\omega_x t) \end{pmatrix}$$

Solutions numériques et rotations

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

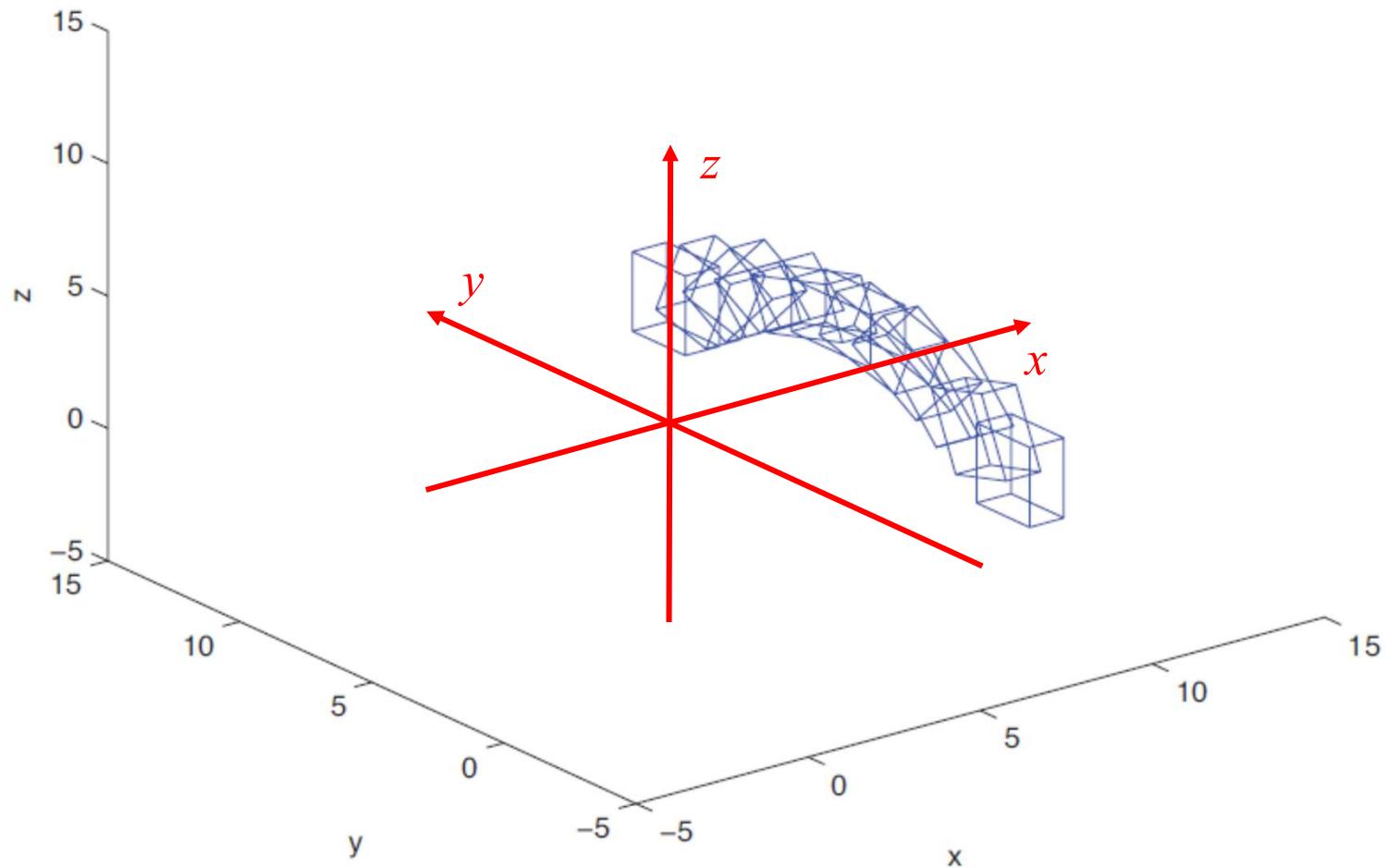
Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Si on utilise $\Delta t_E = 0.01\text{s}$ on obtient.



Solutions numériques et rotations

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Au lieu de passer par les matrices de rotation, on peut utiliser les quaternions de rotation

4 équations au lieu de 9.

$$\frac{d\vec{\tilde{R}}(t)}{dt} = \frac{1}{2} \vec{\tilde{R}}(t) \vec{\omega}(t)$$

Avec $\vec{\omega}(t) = (0, \vec{\omega}(t))^T$ et $\vec{\tilde{R}}(t)$ le quaternion de rotation associé à la matrice de rotation $\mathbf{R}(t)$.

Solutions numériques et rotations

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

La résolution par la méthode d'Euler donnera alors

$$\vec{\tilde{R}}(t_0 + \Delta t) = \vec{\tilde{R}}(t_0) + \frac{1}{2} \vec{\tilde{R}}(t_0) \vec{\omega}(t_0) \Delta t$$

Le problème de changement dans les dimensions du solide observé lorsque l'on utilise \mathbf{R} persiste, car $\left\| \vec{\tilde{R}}(t_0 + \Delta t) \right\| \neq 1$

On peut cependant régler ce problème en utilisant:

$$\vec{\tilde{R}}(t_0 + \Delta t) = \frac{\vec{\tilde{R}}(t_0 + \Delta t)}{|\vec{\tilde{R}}(t_0 + \Delta t)|}$$

Solutions numériques et rotations

Introduction
Résolution des équations de la cinématique des particules
Résolution des équations de la cinématique des solides
Résolution numérique d'EDO
Solutions numériques et rotations
Solutions numériques et collisions
Conclusion

Trajectoire d'une tige de métal très fine dans l'espace

Une tige de métal très fine de longueur $l = 1 \text{ m}$ dont le centre de masse est centré initialement à $\vec{r}_c(0) = (0, 0, 5)^T \text{ m}$ se déplace dans l'air. Cette tige qui est initialement alignée avec l'axe des z et possède une vitesse linéaire initiale $\vec{v}_c(0) = (1, 1, 1)^T \text{ m/s}$ et une vitesse angulaire initiale $\vec{\omega}(0) = (0, 1, 1)^T \text{ rads/s}$. Elle subit une accélération linéaire constante $\vec{a}(t) = \vec{a}(0) = (0, 0, -9,8)^T \text{ m/s}^2$ et une accélération angulaire nulle. Ici, on vous demande de déterminer la position de cette tige et sa longueur après 2 secondes en utilisant

- la solution exacte ;
- la solution par méthode de Euler avec quaternions non normalisés ;
- la solution par méthode de Euler avec quaternions normalisés.

Solutions numériques et rotations

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

La solution pour la position du centre de masse est obtenue analytiquement

$$\vec{r}_c(2) = \vec{r}_c(0) + 2\vec{v}(0) + 2\vec{a}_c(0) = (2, 2, -12.6)^T,$$

La solution pour les quaternions de rotation requiert une condition initiale. Ici, nous supposerons une rotation de $\theta = 0$ rad autour de l'axe \hat{z} . Ceci correspond au quaternion de rotation initial suivant

$$R(0) = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

et à la position angulaire initiale $\vec{\Omega}(0) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$.

Solutions numériques et rotations

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

La solution exacte pour le quaternion de rotation est simple à obtenir, car l'accélération angulaire est nulle. On résout donc les équations du mouvement pour la position angulaire

$$\vec{\Omega}(2) = \vec{\Omega}(0) + 2\vec{\omega}(0) = (0, 2, 2)^T = 2\sqrt{2} \left(0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)^T.$$

Cette nouvelle position angulaire correspond à la matrice de rotation

$$\mathbf{R}(2) = \frac{1}{\sqrt{2}} \begin{pmatrix} \cos(\sqrt{8}) & -\frac{1}{\sqrt{2}} \sin(\sqrt{8}) & \frac{1}{\sqrt{2}} \sin(\sqrt{8}) \\ \frac{1}{\sqrt{2}} \sin(\sqrt{8}) & \frac{1}{2}(1 + \cos(\sqrt{8})) & \frac{1}{2}(1 - \cos(\sqrt{8})) \\ -\frac{1}{\sqrt{2}} \sin(\sqrt{8}) & \frac{1}{2}(1 - \cos(\sqrt{8})) & \frac{1}{2}(1 + \cos(\sqrt{8})) \end{pmatrix}$$

ou au quaternion de rotation

$$\vec{\vec{R}}(2) = \left(\cos(\sqrt{2}), 0, \frac{\sin(\sqrt{2})}{\sqrt{2}}, \frac{\sin(\sqrt{2})}{\sqrt{2}} \right)^T$$

Solutions numériques et rotations

Introduction
Résolution des équations de la cinématique des particules
Résolution des équations de la cinématique des solides
Résolution numérique d'EDO
Solutions numériques et rotations
Solutions numériques et collisions
Conclusion

$$\mathbf{R}(t) = \begin{pmatrix} u_x^2 + (u_y^2 + u_z^2) \cos \theta & u_x u_y (1 - \cos \theta) - u_z \sin \theta & u_x u_z (1 - \cos \theta) + u_y \sin \theta \\ u_x u_y (1 - \cos \theta) + u_z \sin \theta & u_y^2 + (u_z^2 + u_x^2) \cos \theta & u_y u_z (1 - \cos \theta) - u_x \sin \theta \\ u_x u_z (1 - \cos \theta) - u_y \sin \theta & u_y u_z (1 - \cos \theta) + u_x \sin \theta & u_z^2 + (u_x^2 + u_y^2) \cos \theta \end{pmatrix}$$

Solutions numériques et rotations

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Les extrémités des tiges qui se trouvaient initialement à la position

$$\vec{r}_{b,c}(0) = (0, 0, -0.5)^T \text{ m} ,$$

$$\vec{r}_{t,c}(0) = (0, 0, 0.5)^T \text{ m} ,$$

par rapport au centre de masse seront maintenant à la position

$$\vec{r}_b(2) = \vec{r}_c(2) + \mathbf{R}(2)(0, 0, -0.5)^T = (1.8911, 1.5122, -12.6121)^T \text{ m} ,$$

$$\vec{r}_t(2) = \vec{r}_c(2) + \mathbf{R}(2)(0, 0, 0.5)^T = (2.1089, 2.4878, -12.5878)^T \text{ m} .$$

On voit donc que la longueur de la tige, qui est donnée par $l = |\vec{r}_t(2) - \vec{r}_b(2)| = 1 \text{ m}$, n'a pas changé.

Résolution des équations de la cinématique des solides

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

$$\vec{q}(t) = \begin{pmatrix} v_{c,x}(t) \\ v_{c,y}(t) \\ v_{c,z}(t) \\ x_c(t) \\ y_c(t) \\ z_c(t) \\ \omega_x(t) \\ \omega_y(t) \\ \omega_z(t) \\ \vec{R}_0(t) \\ \vec{R}_1(t) \\ \vec{R}_2(t) \\ \vec{R}_3(t) \end{pmatrix} \quad \vec{g}(\vec{q}, t) = \begin{pmatrix} a_{c,x}(\vec{q}, t) \\ a_{c,y}(\vec{q}, t) \\ a_{c,z}(\vec{q}, t) \\ q_1(t) \\ q_2(t) \\ q_3(t) \\ \alpha_x(\vec{q}, t) \\ \alpha_y(\vec{q}, t) \\ \alpha_z(\vec{q}, t) \\ (-q_{11}(t)q_7(t) - q_{12}(t)q_8(t) - q_{13}(t)q_9(t))/2 \\ (q_{10}(t)q_7(t) + q_{12}(t)q_9(t) - q_{13}(t)q_8(t))/2 \\ (q_{10}(t)q_8(t) + q_{13}(t)q_7(t) - q_{11}(t)q_9(t))/2 \\ (q_{10}(t)q_9(t) + q_{11}(t)q_8(t) - q_{12}(t)q_7(t))/2 \end{pmatrix}$$

$\frac{1}{2} \vec{R}(t) \vec{\omega}(t)$

Résolution des équations de la cinématique des solides

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Pour la solution par la méthode d'Euler, on peut procéder de la façon suivante ($\Delta t=2$ s).

- Pour la position du centre de masse, on utilise

$$\vec{r}_c(t) = \text{SEDEuler}(t_0, \vec{r}_c(t_0), \Delta t, 'g1').$$

avec

```
function res=g1(q0,t0)
% dr(t)/dt=v_0+a_0*t
res=[1 1 1]+[0 0 -9.8]*t0;
```

car $\vec{v}(t) = \vec{v}(0) + \vec{a}(0)t$.

Résolution des équations de la cinématique des solides

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

- Pour le quaternion de rotation (position angulaire), on utilise.

$$\vec{R}(t_0) = (1, 0, 0, 0)^T ,$$

$$\vec{R}(t) = \text{SEDEuler0}(\vec{R}(t_0), t, \Delta t, 'g2') ,$$

avec

```
function res=g2(q0,t0)
% dR(t)/dt=(R w)/2
ww=[0 0 1 1];
res=QProduit(q0,ww)/2;
```

la fonction QProduit étant définie au chapitre 2.

- Normalisation du quaternion, si nécessaire.

$$\vec{R}_n(t) = \vec{R}(t) / |\vec{R}(t)| .$$

Résolution des équations de la cinématique des solides

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Si le quaternion de rotation n'est pas normalisé, les extrémités de la tige sont:

$$\vec{r}_{E,b}(2) = (1.8596, 1.4071, -12.6166)^T \text{ m},$$

$$\vec{r}_{E,t}(2) = (2.1404, 2.5929, -12.5834)^T \text{ m},$$

$$l_E = 1.2190 \text{ m}$$

Valeurs analytiques:

$$\vec{r}_b(2) = \vec{r}_c(2) + \mathbf{R}(2)(0, 0, -0.5)^T = (1.8911, 1.5122, -12.6121)^T \text{ m},$$

$$\vec{r}_t(2) = \vec{r}_c(2) + \mathbf{R}(2)(0, 0, 0.5)^T = (2.1089, 2.4878, -12.5878)^T \text{ m}.$$

Résolution des équations de la cinématique des solides

Introduction
Résolution des
équations de la
cinématique des
particules

Résolution des
équations de la
cinématique des
solides

Résolution
numérique d'EDO

Solutions numériques
et rotations

Solutions numériques
et collisions

Conclusion

Si le quaternion de rotation est normalisé:

$$\vec{r}_{N,b}(2) = (1.8848, 1.5136, -12.6136)^T \text{ m},$$

$$\vec{r}_{N,t}(2) = (2.1152, 2.4864, -12.5864)^T \text{ m},$$

$$l_N = 1,00008$$

Valeurs analytiques:

$$\vec{r}_b(2) = \vec{r}_c(2) + \mathbf{R}(2)(0, 0, -0.5)^T = (1.8911, 1.5122, -12.6121)^T \text{ m},$$

$$\vec{r}_t(2) = \vec{r}_c(2) + \mathbf{R}(2)(0, 0, 0.5)^T = (2.1089, 2.4878, -12.5878)^T \text{ m}.$$

Table des matières

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

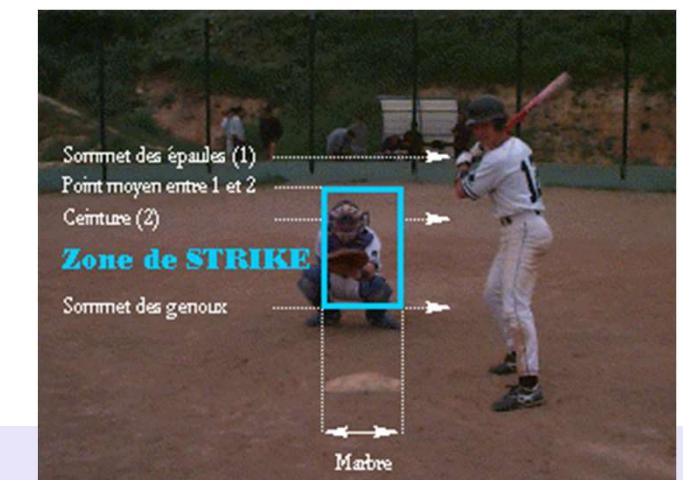
- Introduction
- Résolution des équations de la cinématique des particules
- Résolution des équations de la cinématique des solides
- Résolution numérique d'EDO
- Solutions numériques et rotations
- Solutions numériques et collisions**
- Conclusion

Solutions numériques et collisions

Introduction
 Résolution des équations de la cinématique des particules
 Résolution des équations de la cinématique des solides
 Résolution numérique d'EDO
 Solutions numériques et rotations
Solutions numériques et collisions
 Conclusion

Trajectoire d'une balle au baseball

Un lanceur mesurant 2.1 m lance une balle de baseball (diamètre $d_b = 7.3$ cm et masse $m_b = 145$ g) vers le marbre situé à 18.44 m (60 pieds 6 pouces). Cette balle doit atteindre une région virtuelle de l'espace, la zone des prises, qui se situe au centre du marbre et a une largeur correspondant à celle du marbre (30.48 cm ou 12 pouces) et une hauteur d'un mètre, le bas de la zone se situant à 0.8 m (entre les genoux du frappeur et un point qui se situe à mi-distance entre la ceinture et les épaules du frappeur). La surface de la zone des prises est perpendiculaire à la ligne joignant le lanceur au centre du marbre.



Solutions numériques et collisions

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

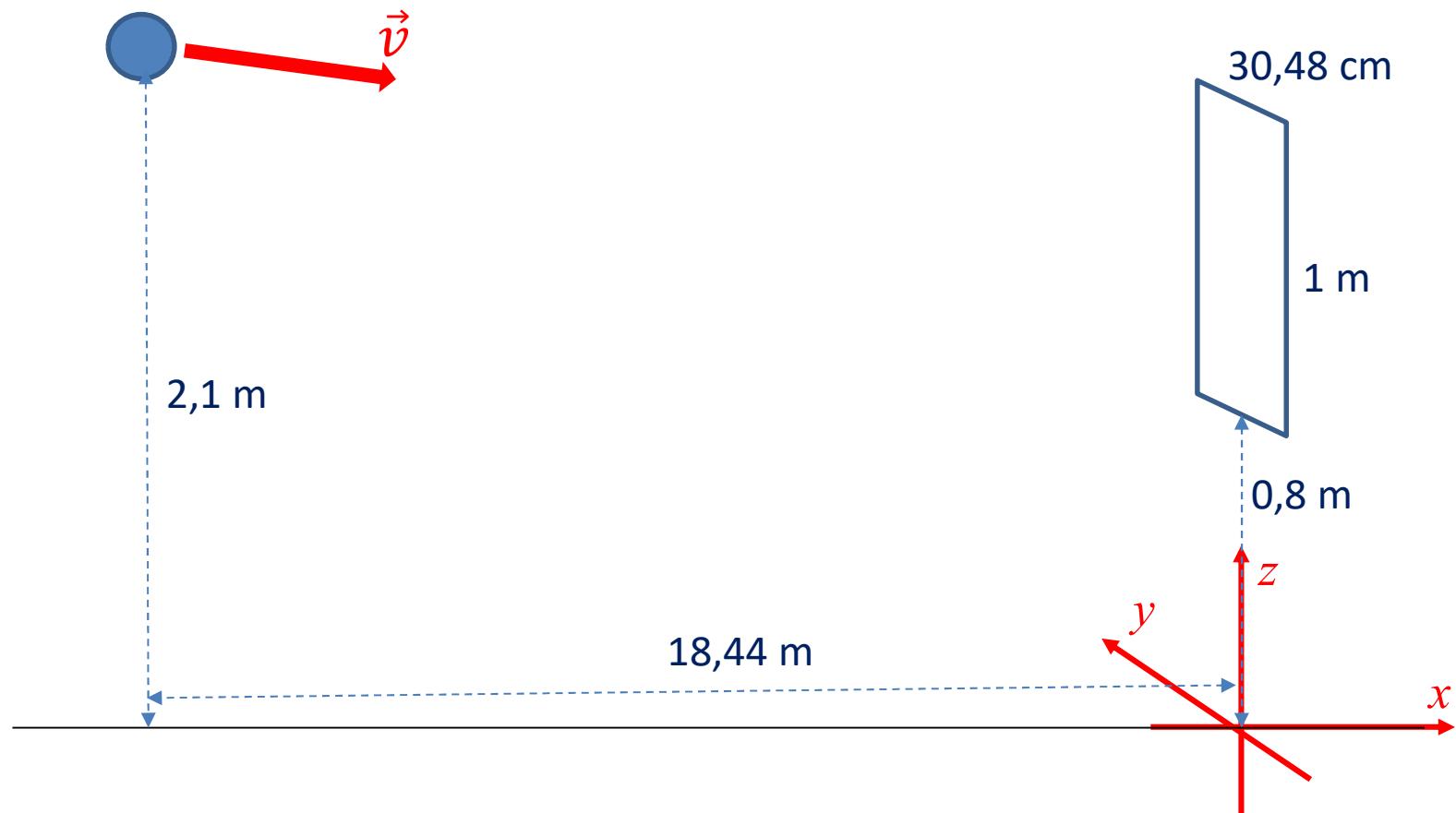
Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Trajectoire d'une balle au baseball



Solutions numériques et collisions

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Le but est de simuler et de tracer la trajectoire de la balle de son point de départ (position du lanceur et hauteur $z = 2.1 \text{ m}$) jusqu'à ce qu'elle touche le sol, ou atteigne la zone des prises. Il faut aussi déterminer si une «prise» a été lancée. La balle subit une accélération constante due à la gravité donnée par $\vec{a} = (0, 0, -g)^T$ avec $g = 9.8 \text{ m/s}^2$.

Solutions numériques et collisions

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Deux obstacles peuvent se retrouver le long de la trajectoire de la balle (considérée comme ponctuelle) :

1. la zone des prises (surface virtuelle) avec $x_p = 0$ et

$$y_{\min,p} + d_b/2 \leq y(t_p) \leq y_{\max,p} - d_b/2$$

$$z_{\min,p} + d_b/2 \leq z(t_p) \leq z_{\max,p} - d_b/2$$

car la balle de diamètre d_b doit se retrouver totalement dans la zone définie par $y_{\pm,p} = \pm 0.1524$ m et $z_{\pm,p} = 1.3 \pm 0.5$ m.

2. le sol situé à $z_s = 0$.

La simulation dynamique se termine lorsque la balle touche un de ces deux obstacles.

Solutions numériques et collisions

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Dans ce cas simple, une résolution analytique est possible. La façon de procéder est la suivante.

1. Déterminer les trajectoires exactes.

$$\vec{r}(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} = \begin{pmatrix} x(0) \\ y(0) \\ z(0) \end{pmatrix} + \begin{pmatrix} v_x(0) \\ v_y(0) \\ v_z(0) - \frac{g}{2}t \end{pmatrix} t$$

2. Déterminer le temps requis t_s pour que la balle touche le plan $z_s = 0$ correspondant au sol ce qui correspond à résoudre:

$$z(t_s) = d_b/2$$

C'est-à-dire :

$$4.9t_s^2 - v_z(0)t_s + \frac{d_b}{2} - z(0) = 0$$

Solutions numériques et collisions

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO
Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

- Déterminer le temps requis t_p pour que la balle traverse le plan $x(t_p) = 0$ correspondant à la zone des prises

$$t_p = \frac{-x(0)}{v_0}$$

- Si $t_s < t_p$ le sol est touché en premier, la simulation se termine au temps $t_f = t_s$.
- Si $t_p < t_s$, la zone des prises pourrait être traversée. Il faut alors déterminer les positions $y(t_p)$ et $z(t_p)$ où se trouve le projectile au temps t_p

$$y(t_p) = y(0) + v_y(0)t_p$$

$$z(t_p) = z(0) + v_z(0)t_p - 4.9t_p^2$$

Solutions numériques et collisions

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

6. Si $y(t_p)$ et $z(t_p)$ sont dans la zone des prises, le lanceur à tiré une prise et la simulation s'arrête au temps $t_f = t_p$.
7. Sinon la balle n'atteint pas la zone des prises et la trajectoire se termine lorsqu'elle touche le sol à $t_f = t_s$.

Solutions numériques et collisions

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Si on utilise une solution numérique, la procédure est la suivante.

1. Choisir une méthode de résolution numérique et un intervalle de temps initial Δt pour la simulation de la trajectoire. Ici, on choisit Δt suffisamment court pour :
 - ◆ donner une bonne précision de simulation pour la trajectoire sans contrainte ;
 - ◆ s'assurer que la distance parcourue par l'objet ne soit pas trop grande, car celle-ci est utilisée pour la détection des collisions.

On choisit préférablement la méthode Runge-Kutta à cause de sa haute précision.

Solutions numériques et collisions

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

2. Déterminer la position atteinte par le projectile

$(x_i, y_i, z_i)^T = (x(t_i), y(t_i), z(t_i))^T$ au temps $t_i = t_0 + i\Delta t$ en utilisant la méthode de résolution numérique choisie.

Après chaque intervalle de calcul on vérifie si la balle :

- ◆ traverse le sol ($z_{i-1} > d_b/2 \geq z_i$);
- ◆ traverse la position x_p de la zone des prises ($x_{i-1} < x_p \leq x_i$).

Sinon, on passe au prochain intervalle de temps.

Solutions numériques et collisions

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

3. Si la balle traverse le sol sans traverser la zone des prises: on suppose que la balle traversera le sol avant la zone des prises (ceci est assuré si la solution est assez précise). On doit alors déterminer à quel temps t_f la balle (cm) traversera le sol ($z_s = d_b/2$).
4. Si le projectile traverse le plan de la cible sans traverser le sol, il pourrait frapper la cible.
5. Si y et z sont aussi dans le bon intervalle pour la cible, on imprime la position de la balle et on termine la simulation. Sinon, on continue la simulation pour le prochain intervalle de temps.

Solutions numériques et collisions

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

6. Si le plan de la zone des prises et le plan du sol ont été croisés par la balle (possible si Δt est trop long):

On diminue le pas de temps sur le dernier intervalle et on résout sur ce pas de temps en retournant à l'étape 2.

Comment déterminer t_f ?

Solutions numériques et collisions

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

1 Méthode d'interpolation.

- a Utiliser une approximation pour la vitesse moyenne $\bar{v}_z(t_{i-1})$ sur l'intervalle $t_{i-1} \leq t \leq t_i$. Ici, on supposera une approximation linéaire:

$$\bar{v}_z(t_{i-1}) = \frac{v_z(t_i) + v_z(t_{i-1})}{2}$$

$$t_f = t_{i-1} + \frac{z_s - z(t_{i-1})}{\bar{v}_z(t_{i-1})} = t_{i-1} - \frac{z(t_{i-1})}{\bar{v}_z(t_{i-1})}$$

- b On résout le problème numériquement pour une dernière fois et on évalue la position en x correspondant à t_f (on suppose que $z(t_f) = z_s$).

Solutions numériques et collisions

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

- 2 Méthode itérative avec ε une distance faible qui est utilisée pour s'assurer de la convergence du processus.
- a Si $|z_s - z(t_{i-1})| < \varepsilon$, on suppose que l'intersection a eu lieu à t_{i-1} et on termine alors la simulation.
 - b Si $|z_s - z(t_i)| < \varepsilon$, on suppose que l'intersection a eu lieu à t_i et on termine alors la simulation.
 - c Sinon on divise l'intervalle $t_{i-1} \leq t \leq t_i$ en J sous intervalles et on résout le problème pour $t_j = t_{i-1} + j \Delta t / J$.
 - d. On vérifie ensuite si l'intersection se situe entre $t_{j-1} \leq t \leq t_j$.
 - e. Si $|z_s - z(t_j)| < \varepsilon$, on suppose que l'intersection a eu lieu à t_j et on termine alors la simulation.
 - f. Sinon on remplace $t_{i-1} = t_{j-1}$ et $t_i = t_j$, $\Delta t = \Delta t / J$ et on retourne à l'étape "c" et on poursuit jusqu'à convergence.

Solutions numériques et collisions

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Notez que :

- si la précision de notre algorithme de solution numérique n'est pas suffisante, il se peut que l'on manque ou que l'on touche la cible alors que la solution analytique (si disponible) nous indique le contraire.

- si les intervalles de temps de simulation sont trop longs (même si la méthode de simulation est précise), il se peut aussi que l'on manque des collisions ou que l'ordre des collisions soit incorrect.

Tout ceci pourrait avoir un impact important sur le reste de la simulation (s'il y a des rebonds par exemple).

Solutions numériques et collisions

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

Résolution numérique d'EDO
Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Si la cible correspond à un cube en 3D posé sur le sol, et que la particule se déplace aussi dans la direction y le problème se complique.

- Il faut vérifier la possibilité d'interaction avec toutes les faces du cube et avec le sol.
- Après avoir déterminé la succession de ces événements dans le temps, on peut procéder au processus itératif en prenant l'intervalle de temps le plus court.
- Ceci nous permet un premier classement des surfaces touchées successivement.
- Comme les vitesses ne sont pas nécessairement constantes, il peut arriver que ce classement soit faux. On doit donc reprendre l'analyse de toutes les surfaces à chaque itération de façon à minimiser les fausses collisions.

Solutions numériques et collisions

Introduction
Résolution des équations de la cinématique des particules

Résolution des équations de la cinématique des solides

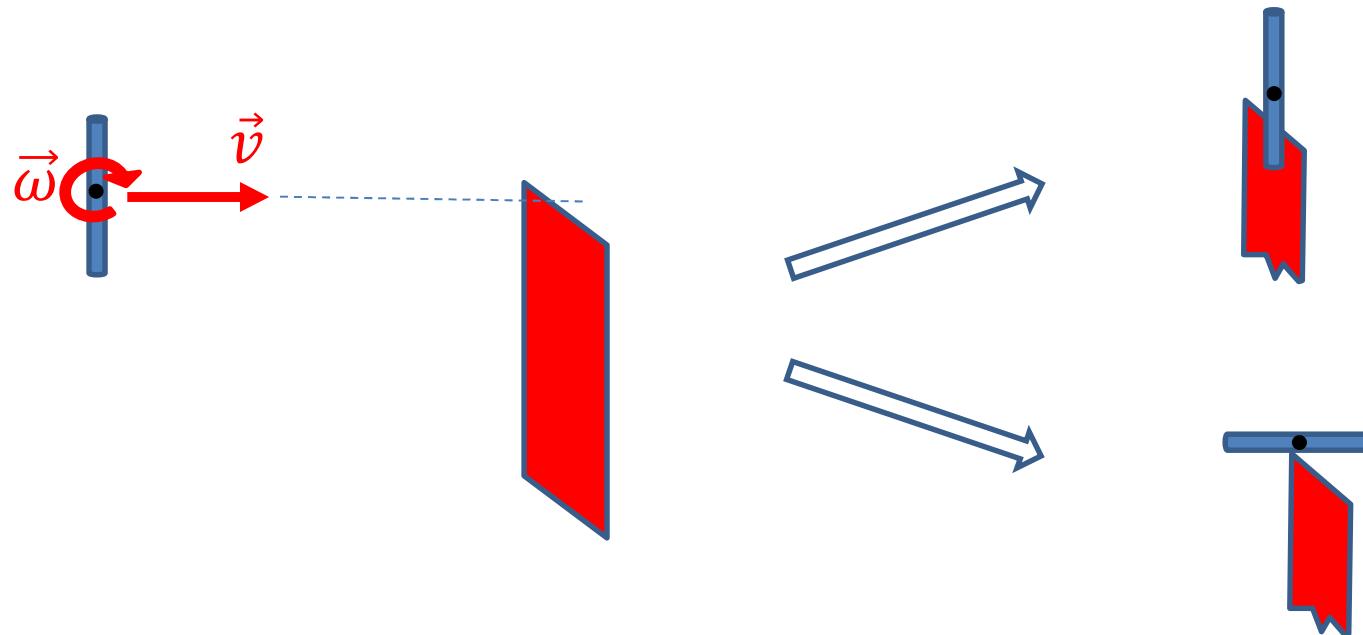
Résolution numérique d'EDO

Solutions numériques et rotations

Solutions numériques et collisions

Conclusion

Finalement, si le projectile a une forme arbitraire il faut généraliser cette analyse à tous les points de l'objet. Pour les solides de forme arbitraire subissant un mouvement de rotation ce problème est très difficile à résoudre et sera considéré au chapitre 5.



Conclusion

Introduction
Résolution des équations de la cinématique des particules
Résolution des équations de la cinématique des solides
Résolution numérique d'EDO
Solutions numériques et rotations
Solutions numériques et collisions
Conclusion

Dans ce chapitre nous avons étudié des méthodes numériques qui peuvent être utilisées pour résoudre les équations pour le mouvement dans l'espace de solides isolés. Nous avons aussi étudié l'impact de telles solutions numériques sur la détection de collisions. Dans le prochain chapitre, nous présenterons les forces et moments de force pouvant influencer le déplacement (en translation et en rotation) des solides.