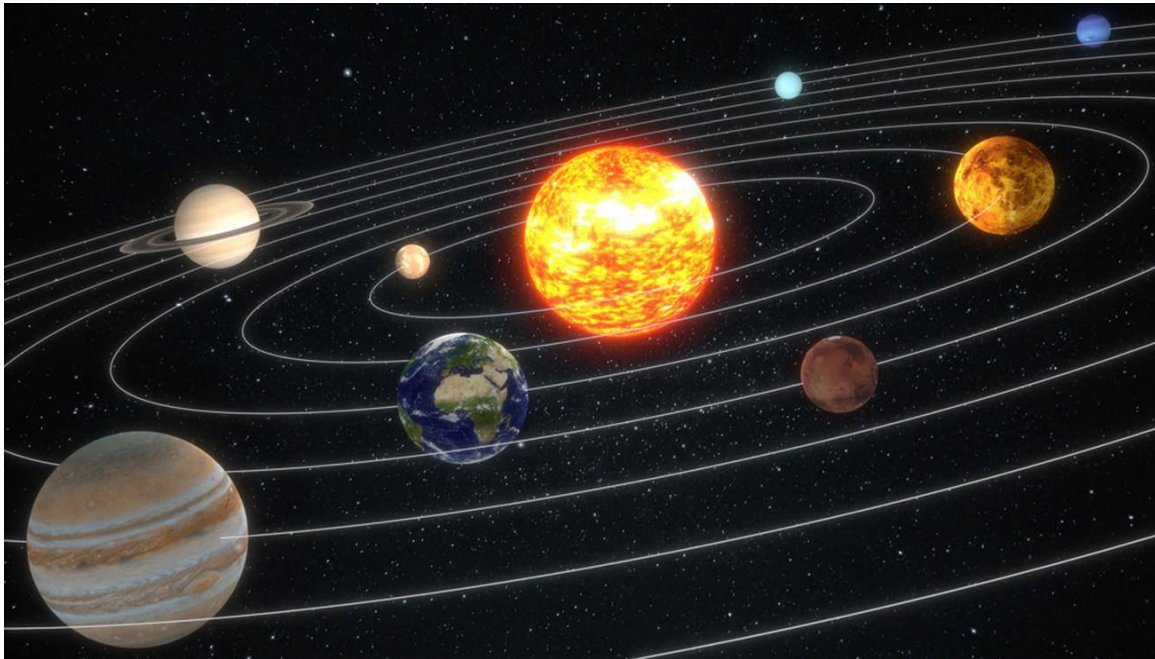


RAPPORT DE PROJET :

Simulation du système solaire.



Etudiants :

- Touzari Lisa.
- Lekhmamra Chihab Eddine.
- Fellah Abd Elmalek.
- Snaoui Yacine.

1. Description du problème posé et la méthode de résolution :

- Notre projet consiste à modéliser la rotation des planètes autour du soleil.
- Ce dernier est constitué de 4 étapes essentielles :
 - a. Modélisation de la rotation de la lune autour de la terre en considérant la terre comme étant fixe.
 - b. Modélisation de la rotation de la lune autour de la terre en considérant la rotation de la terre autour du soleil.
 - c. Modélisation de la rotation des planètes du système solaire autour du soleil.
 - d. Modélisation de la rotation des planètes du système solaire autour du soleil en considérant les interactions entre les planètes.
- La méthode de résolution :
Afin de résoudre le problème posé, il nous suffira de trouver la solution des équations différentielles de deuxième ordre qui modélisent le mouvement de l'ensemble des planètes.

2. Descriptions des méthodes numériques utilisées :

- On va utiliser la méthode de : Euler, Range Kutta 2, Range Kutta 4, différences finies.
- a. Euler :
 - exemple d'équation à résoudre :

$$M_L \frac{d^2}{dt^2} \mathbf{r}_L(t) = \mathbf{F}_g = \frac{GM_T M_L}{r^2(t)} \left(-\frac{\mathbf{r}_L(t)}{r(t)} \right)$$

Avec :

$$\mathbf{r}(t) = \begin{pmatrix} x_L(t) \\ y_L(t) \end{pmatrix}$$

On obtient alors le système d'équations :

suivant x : $d^2 X_L(t)/dt^2 = -GMT/R^3(t) * X_L(t)$

suivant y : $d^2 Y_L(t)/dt^2 = -GMT/R^3(t) * Y_L(t)$

-en appliquant euler, on calcule pour chaque itération à l'instant t
x(t) et y(t) :

suivant y :

$$Y(t+1) = y(t) + h.f(y(t), t) = [V_y(t) ; Y(t)] + h.[-G.M_t.Y_L(t)/R(t) ; V_y(t)]$$

suivant x :

$$X(t+1) = x(t) + h.f(x(t), t) = [V_x(t) ; X(t)] + h.[-G.M_t.X_L(t)/R(t) ; V_x(t)]$$

- b. Range Kutta 2 : on utilise le même principe que la méthode d'euler, on aura le système suivant :

Suivant y :

$$y(t+h)=y(t)+h.(f(t,y)+h.f(t+1,y(t)+f(t,y)))/2$$

Suivant x :

$$x(t+h)=x(t)+h.(f(t,x)+h.f(t+1,x(t)+f(t,x)))/2$$

- c. Range Kutta 4 :

Suivant y :

$$k1=h.f(t,y)$$

$$k2=h.f(t+h/2, y+k1/2)$$

$$k3 =h.f(t+h/2, y+k2/2)$$

$$k4 =h.f(t+h, y+k3)$$

$$y(t+h)=y(t)+(k1+2.k2+2.k3+k4)/6$$

Suivant x :

$$k1= h.f(t,x)$$

$$k2= h.f(t+h/2, x+k1/2)$$

$$k3 = h.f(t+h/2, x+k2/2)$$

$$k4= h.f(t+h, x+k3)$$

$$x(t+h)= x(t)+(k1+2.k2+2.k3+k4)/6$$

- d. Différences finies :

Suivant y $y(t+h)=h.h.f(t)/M +2.y(t) - y(t-h).$
Avec : $y(0)-y(-1)= h.v0$

Suivant x : $x(t+h)=h.h.f(t)/M +2.x(t) - x(t-h).$
Avec : $x(0)-x(-1)= h.v0$

3. Le principe de l'algorithme programmé et la structuration de données :

- Le fichier header : « projet.h »
Ce fichier contient les prototypes de toutes les fonctions utilisées, la déclaration des constantes ainsi que les différentes structures.
- Le fichier source : « projet.c »
Ce fichier contient le code décrivant nos fonctions.

- Le main : « main.c » :

Dans ce fichier on écrit notre programme principale, on fait appel à nos fonctions, afin de déterminer la trajectoire des différentes planètes du système solaire.

- Le principe de fonctionnement :

- dans le fichier header : on déclare l'ensemble des masses des différentes planètes comme des constantes ; chaque planète est représentée par une structure ayant 5 champs :
 1. indice : pour différencier les différentes planètes.
 2. x0,y0 : position initiale.
 3. Vx0,vy0 : vitesse initiale.

- Le prototype des fonctions :

```
double ** position1(void);
```

Cette dernière nous permettra de calculer la position de la lune par rapport à la terre, pour l'étape 1.

```
double ** position2(void);
```

Cette dernière nous permettra de calculer la position de la lune par rapport à la terre, pour l'étape 2.

```
double ** position3( planete p);
```

Cette dernière nous permettra de calculer la position de chaque planète par rapport au soleil, pour l'étape 3.

```
void **interaction(void);
```

Cette dernière nous permettra de réaliser l'étape 4.

```
planete init_planete();
```

Cette fonction permettra d'attribuer à chaque planète sa masse correspondante, ainsi que les conditions initiales.

- Dans le fichier source : on décrit les différentes fonctions :

```
double ** position1(void);
```

- Cette fonction ne reçoit aucun paramètre en entrée, et en sortie on aura une matrice de type double. Notre but est de calculer la position de la lune par rapport à la terre (fixe), pour cela on aura le choix d'appliquer 4 méthodes de calcul : euler, range kutta 2, range kutta 4, différences finies.

- A partir des équations mathématiques décrivant les méthodes, on pourra écrire un programme c nous permettant de déterminer la position à des instants t déterminés, on récupère à la fin notre résultat dans deux fichiers.

```
double ** position2(void);
```

-Cette fonction ne reçoit aucun paramètre en entrée, et en sortie on aura une matrice de type double. Notre but est de calculer la position de la lune par rapport à la terre (en mouvement), la méthode des différences finies étant la plus stable, on appliquera cette dernière.

```
double ** position3( planete p);
```

-Cette fonction reçoit comme paramètre d'entrée « planète p », et en sortie une matrice de type double qui représente les coordonnées de notre planète. On l'utilisera pour déterminer la trajectoire de p autour du soleil.
On appliquera la méthode des différences finies.

```
void **interaction(void);
```

-Cette fonction n'a ni de paramètre d'entrée ni de sortie, on l'utilisera pour déterminer l'ensemble des trajectoires des planètes en prenant en compte les interactions entre celle-ci, en récupèrera à la fin 8 fichier textes contenant les coordonnées. On utilisera la méthode des différences finies.

- **Le main :** en exécutant notre programme, en rentrera en premier lieu l'indice de l'étape qu'on veut exécuter, après chaque exécution on récupèrera des fichiers textes contenant les coordonnées voulues.
- On utilisera un programme MATLAB afin de dessiner les différentes trajectoires.

4. Les différentes étapes du projet :

- **Données nécessaires :** en utilisant la relation $T=n.h$; T étant la période de rotation et n le nombre de point de calcul, on détermine de le pas h pour chaque planète.

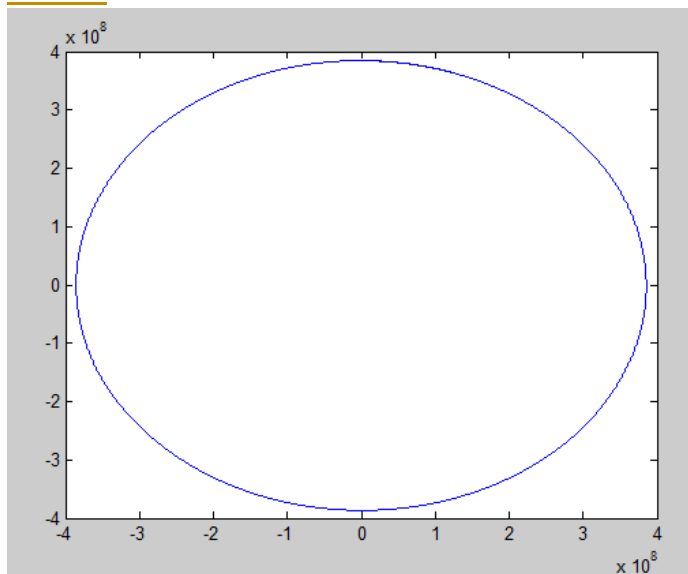
planète	Vitesse (m/s)	Distance(m)	Pas h
Soleil	0	0	-
Mercure	47367.93	$5.791 \cdot 10^{10}$	76
Venus	35025.71	$1.082 \cdot 10^{10}$	194
Terre	29780	$1.496 \cdot 10^{10}$	316
Mars	24080.2	$2.279 \cdot 10^{10}$	590
Jupiter	13058.5	$7.785 \cdot 10^{10}$	3785
Saturne	9640.7	$1.424 \cdot 10^{11}$	9600
Uranus	6796.7	$2.871 \cdot 10^{11}$	27594
Neptune	5432.48	$4.495 \cdot 10^{11}$	52034.4
lune	1022	$3.844 \cdot 10^8$	25

a. Etape 1 : (Système Terre-Lune simplifié) :

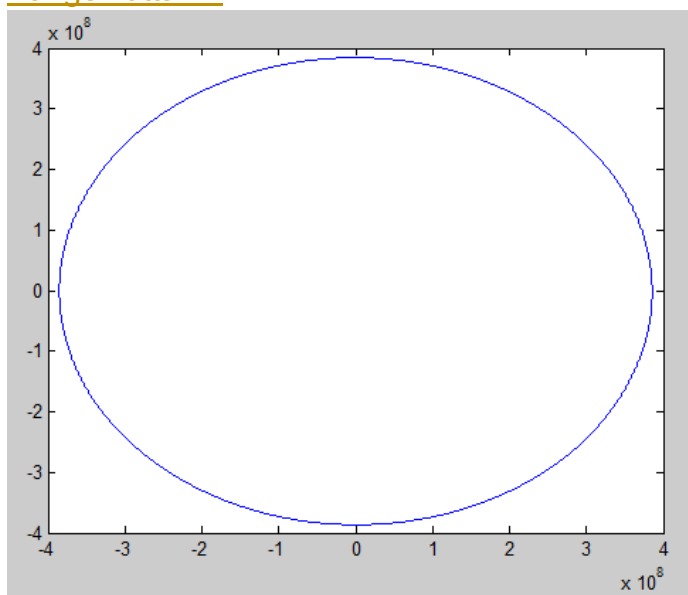
-on utilise la fonction « position 1 », les conditions initiales de la lune et la terre étant prédéfinie, on fait directement appel à notre fonction, et on rentre comme donnée le pas de discrétisation h .

-on récupère le fichier contenant la position de la lune, avec MATLAB on trace notre trajectoire.

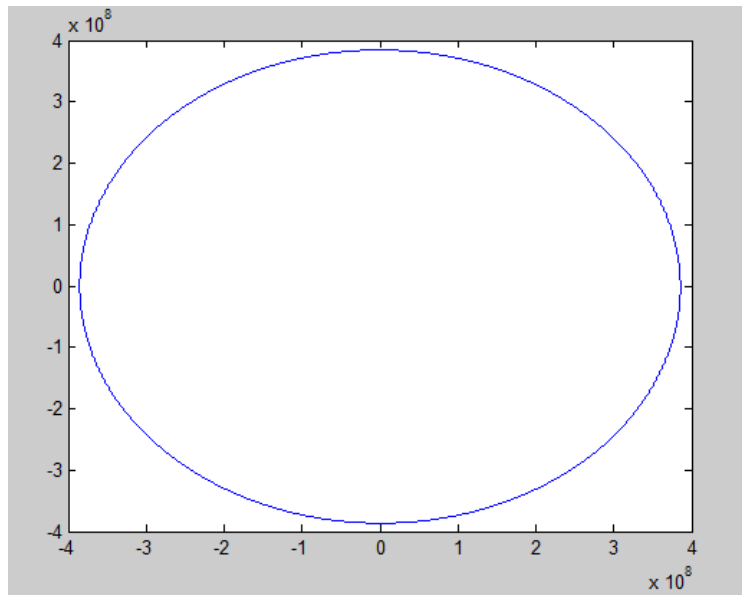
-euler :



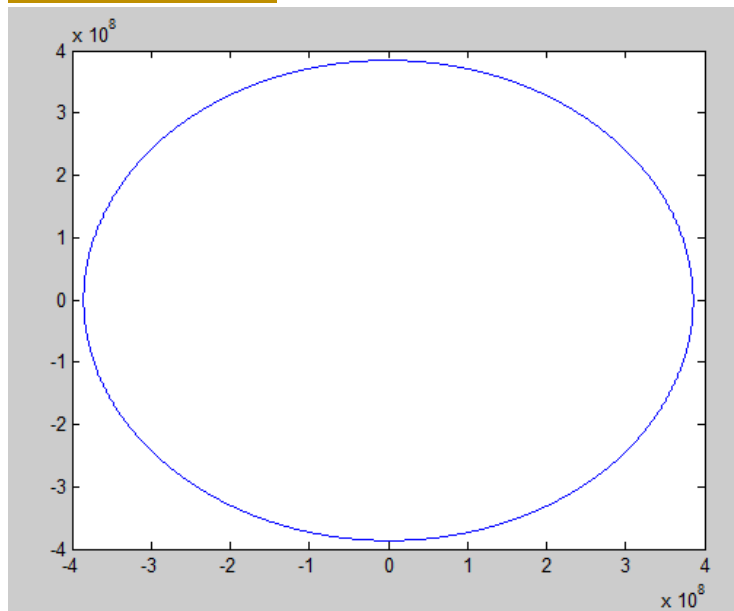
-range kutta 2 :



-range kutta 4 :



-différences finies :

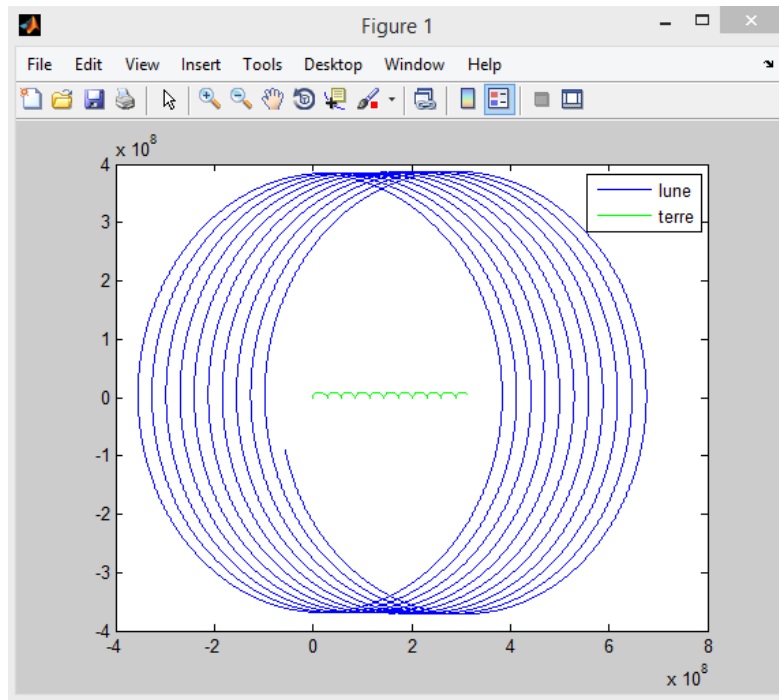


*la méthode des différences finies est la plus stable et la plus précise, de ce fait dans les étapes suivantes elle sera la privilégiée.

b. Etape 2 : (Système Terre-Lune) :

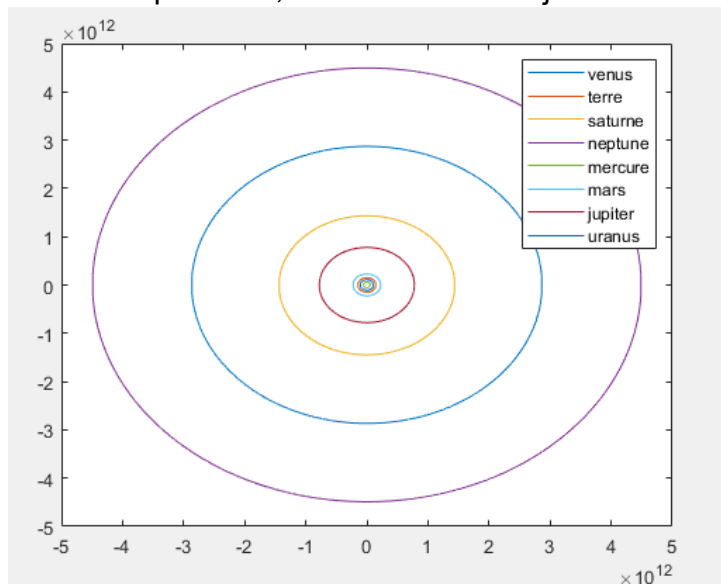
-on déterminera la rotation de la lune autour du soleil, en prenant en considération la rotation de la terre autour du soleil, on utilise la fonction « position 2 », les conditions initiales de la lune et la terre étant prédéfinie, on fait directement appel à notre fonction, et on rentre comme donnée le pas de discrétisation h .

--on récupère le fichier contenant la position de la lune, avec MATLAB on trace notre trajectoire.



c. **Etape 3 : (Système solaire simplifié) :**

- On déterminera la trajectoire des différentes planètes autour du soleil, en considérant ce dernier comme étant fixe, donc les conditions initiales du soleil sont toutes nulles.
- on utilise la fonction « position 3 », en rentre en donnée l'indice de notre planète, la position initiale, la vitesse initiale, le pas de discrétisation h.
- on exécute notre programme pour récupérer les coordonnées de toutes les planètes, on dessine les trajectoires avec MATLAB.



d. **Etape 4 : (système solaire) :**

- cette fois ci, on déterminera la trajectoire des différentes planètes, en prenant en considération les forces appliquées sur chacune d'elles.

-la fonction utilisée est « interaction », les conditions initiales étant prédéfinies, on fait directement appel à cette fonction, les fichiers textes récupérés à la fin de l'exécution contiennent les coordonnées des planètes, on utilisera MATLAB pour dessiner les trajectoires. La méthode utilisée est les différences finies.

5. Les suggestions d'amélioration du projet :

- utilisation d'une base de données pour stocker les résultats au lieu des fichiers texte.
- choix du pas de discrétisation h plus rigoureux.
- utilisation de listes chaînées pour la minimisation de l'espace mémoire perdu.

6. Conclusion :

- A l'aide de ce projet nous avons pu comprendre et expérimenter les différentes méthodes de calcul numériques (euler, range kutta 2, range kutta 4, différences finies).
- mise en pratique des notions apprises dans le cours de programmation c.

7. Bibliographie :

- site internet : google, wikipedia.
- cours programmation c et méthodes numériques L3.