Résolution numérique des équations différentielles

1. Les équations différentielles ordinaires (O.D.E)

Sébastien Charnoz & Adrian Daerr

Université Paris 7 Denis Diderot
CEA Saclay

Les systèmes dynamiques

L'évolution des systèmes dynamiques sont régis par des *équations différentielles*

- Chute d'un corps :

$$a_z = \frac{d^2 z}{dt^2} = -g$$

- Mouvement des planètes :

$$\vec{a}_i = \sum_{j \neq i} \frac{G m_j}{r_{ij}^3} (\vec{r}_i - \vec{r}_j)$$

- Transfert de chaleur :

$$\frac{\partial T}{\partial t} - \kappa \frac{\partial^2 T}{\partial x^2} = f(x,t)$$

- Equation d'onde etc… :

$$\frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = f(x,t)$$

- Etc…

BIEN POSER UN PROBLEME

En fonction du système étudié, il y a toujours

Une ou plusieurs **quantités** à déterminer

$$a_z = \frac{d^2 z}{dt^2} = -g$$

X, V et V=da/dt

$$\frac{\partial T}{\partial t} - \kappa \frac{\partial^2 T}{\partial x^2} = f(x,t)$$

Température T

•En fonction de un ou plusieurs **paramètres**

$$a_z = \frac{d^2 z}{dt^2} = -g$$

Le temps t

$$\frac{\partial T}{\partial t} - \kappa \frac{\partial^2 T}{\partial x^2} = f(x,t)$$

Temps t, espace x

Il y a toujours aussi des *conditions limites (ou conditions initiales)*

Chute des corps , mouvement des planètes :

Positions et vitesses initiales

Transfert de la chaleur :

Condition limite Température initiales (répartition spatiale)

Equation d'onde :

Condition limite (d'une corde par exemple)
Etat initial de la corde

Calculer l'évolution du systèmes c'est

CALCULER L'EVOLUTION DES QUANTITES EN FONCTION DES PARAMETRES

Par exemple :

Pour les planètes : X(t) et V(t)

Pour la chaleur : T(x,t)

En d'autres termes : *résoudre l'équation différentielle (« intégrer »)*

Un problème est bien posé si nous avons

- **Une liste de quantités à intégrer en fonction d'une liste de paramètres**
- **Une equa. diff. d'évolution pour *chaque* quantité qui évolue**
- Les conditions initiales (ou cond. Limites)

EXEMPLES

Propagation d'une onde

Paramètres : X (position) et t (le temps)

Quantité : U(x,t) : Amplitude à la position X et à l'instant t

Equation : $$\frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = f(x,t)$$

forcage

Condition initiale : Dépend du problème bien sur,
Profil u(x) à t=0

loadedstring\index.html

Mouvement des planètes

Quantités : Position et Vitesse : (X,Y,Z, Vx, Vy, Vz)

Paramètre : Temps

Equations différentielles

$$\frac{dx}{dt} = V_x$$

$$\frac{dV_x}{dt} = \sum_{j \neq i} \frac{Gm_j}{r_{ij}^{\;3}} (\vec{r}_i - \vec{r}_j)$$

Idem pout Y et Z

6 au total

Conditions initiales : Position et vitesses initiales

kepler.htm

Propagation de la chaleur

Quantité : température T

Paramètre : X (espace)

équation :

$$\frac{\partial T}{\partial t} - \kappa \frac{\partial^2 T}{\partial x^2} = f(x,t)$$

forcage

Condition initiale : Profil T(x) à t=0

RESOLUTION : LES METHODES

Les outils de résolution des équations sont différents.

Cependant elles se basent toutes sur une même idée , imposée par les limites de l'ordinateur :

La **Discrétisation du problème**

Les paramètres sont discrétisés :

Exemple :

le temps s'écrit $\mathbf{t=t_n = n*dt}$ où **dt** sera le pas de temps

L'espace (à 1D) s'écrira : $\mathbf{x=n*dx}$ où **dx** sera le pas d'espace

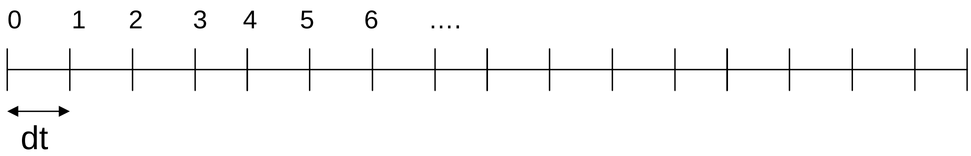
On résout donc le problème sur une **grille** de paramètres.

Une **grille de temps, une grille d'espace etc…**

Plus la grille est fine (dt ou dx) plus la résolution est proche de la solution exacte

---

1D (temps, espace 1D etc…) t_n=n x dt

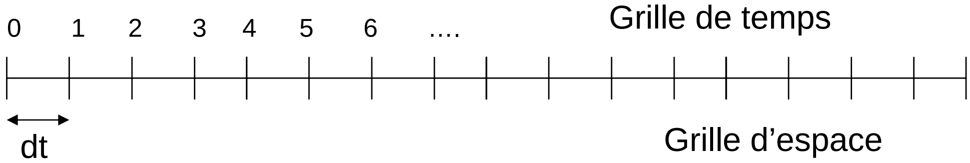


Dans le cas 1D (1 seul paramètre, ex : le temps)

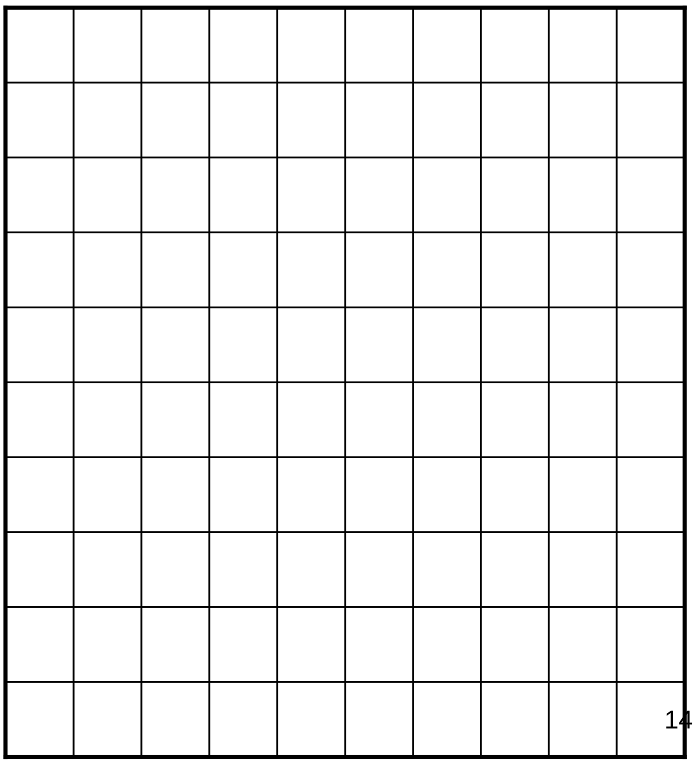
La résolution consiste à calculer une SUITE :

U_{n+1}= F(U_n,t_n) où U_n est la quantité à l'instant t_n

Grille de temps



Grille d'espace



À 2D ou plus

Ex :

$$\frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = f(x,t)$$

En fait on en fait un grille
à 3D (2D espace + 1D temps)

$U_{i,j,k}$=F(Cases voisines)

Pour l'instant on ne s'intéresse qu'aux O.D.E :

Les équations différentielles à 1 seul paramètre

Pour les ODE

$U_{n+1}= F(U_n,t_n)$ où Un est la quantité à l'instant tn

Donc l'équation différentielle se résout de proche en proche
à partir d'un point initial (= condition limite) où
On connaît l'état du système à t=0

La fonction F est appelée « intégrateur » (ou « solver »)

Tout le problème consiste à trouver une fonction
F précise , rapide et robuste.

La précision de la solution dépend de la finesse de la grille
et du nombre d'étapes de calcul.

ILLUSTRATION

Exemple : Système à 1 paramètre : Mouvement d'un ressort

Quantités : X et Vx

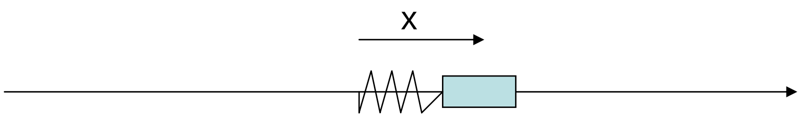
Paramètre : t

Equations : $f = ma = -kx \Rightarrow$

$$\frac{dV_x}{dt} = \frac{-kx}{m}$$

Est- ce tout ???

k : coef. de raideur
m : masse



Non car il nous manque l'équation d'évolution de X :

Equations d'évolutions du système

$$\frac{dx}{dt} = V$$

$$\frac{dVx}{dt} = \frac{-kx}{m}$$

Dans un système à 1 seul paramètre (ODE) on peut TOUJOURS ramener l'équation à un ENSEMBLE d'équations du 1ER ORDRE

On a alors :

Quantités : X et Vx

Conditions initiales : X(t=0) = 10. m
V(t=0)=0. m/s

Paramètre : t

Equations :

$$\frac{dx}{dt} = V$$

$$\frac{dVx}{dt} = \frac{-kx}{m}$$

k : coef. de raideur
m : masse

SOLUTION ANALYTIQUE :
*** démontrer**

$$x(t) = A\cos(\omega t + \varphi)$$

$$v(t) = -A\omega\sin(\omega t + \varphi)$$

ω=sqrt(k/m), A =X0

On prend un grille pour le paramètre t : dt=0.01 seconde

X_{n+1} = f(X_n,tn)
V_{n+1} = f (V_n,tn) , où t_n = n * dt

F est l'intégrateur. Nous prendrons ici la méthode de Euler (nous verrons)

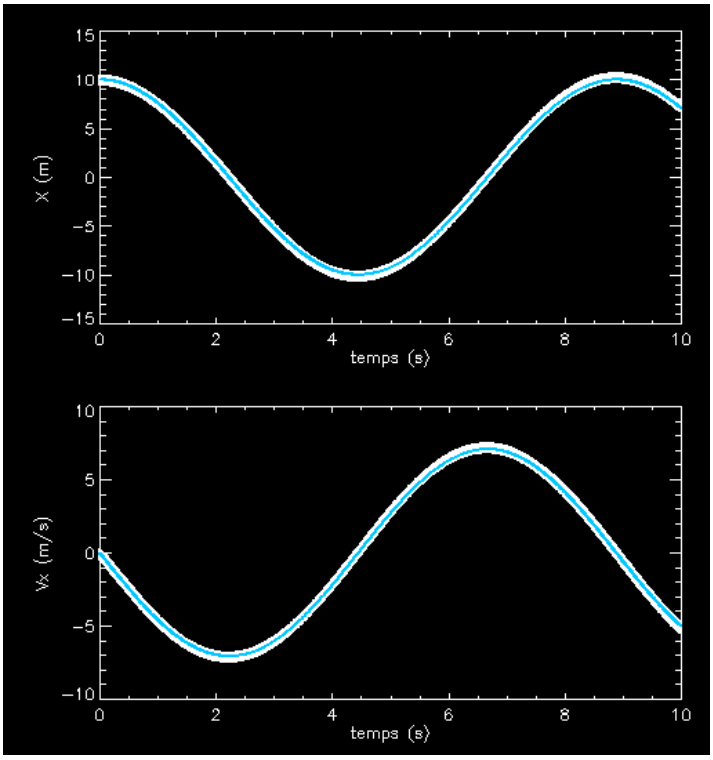
ALGORITHME
1. Initialiser X_0 et V_0
2. Initialiser dt
3. Calculer X_{n+1} = f(Xn,tn) et V_{n+1} = f (Vn,tn)
4. Incrémenter le temps : T=T+dt
5. Revenir en 3.

Solution numérique : ressort résolut par Euler

N	t	X	V
0.000000	0.000000	10.0000	0.000000
1.00000	0.0100000	10.0000	-0.0500000
2.00000	0.0200000	9.99950	-0.100000
3.00000	0.0300000	9.99850	-0.149998
4.00000	0.0400000	9.99700	-0.199990
5.00000	0.0500000	9.99500	-0.249975
6.00000	0.0600000	9.99250	-0.299950
7.00000	0.0700000	9.98950	-0.349912
8.00000	0.0800000	9.98600	-0.399860
9.00000	0.0900000	9.98200	-0.449790
10.0000	0.100000	9.97751	-0.499700
11.0000	0.110000	9.97251	-0.549588
12.0000	0.120000	9.96701	-0.599450
13.0000	0.130000	9.96102	-0.649285
14.0000	0.140000	9.95452	-0.699090
15.0000	0.150000	9.94753	-0.748863
16.0000	0.160000	9.94005	-0.798601
17.0000	0.170000	9.93206	-0.848301
18.0000	0.180000	9.92358	-0.897961
19.0000	0.190000	9.91460	-0.947579
20.0000	0.200000	9.90512	-0.997152
21.0000	0.210000	9.89515	-1.04668
22.0000	0.220000	9.88468	-1.09615
23.0000	0.230000	9.87372	-1.14558
24.0000	0.240000	9.86227	-1.19495
25.0000	0.250000	9.85032	-1.24426
26.0000	0.260000	9.83787	-1.29351
27.0000	0.270000	9.82494	-1.34270
28.0000	0.280000	9.81151	-1.39182
29.0000	0.290000	9.79760	-1.44088

etc..

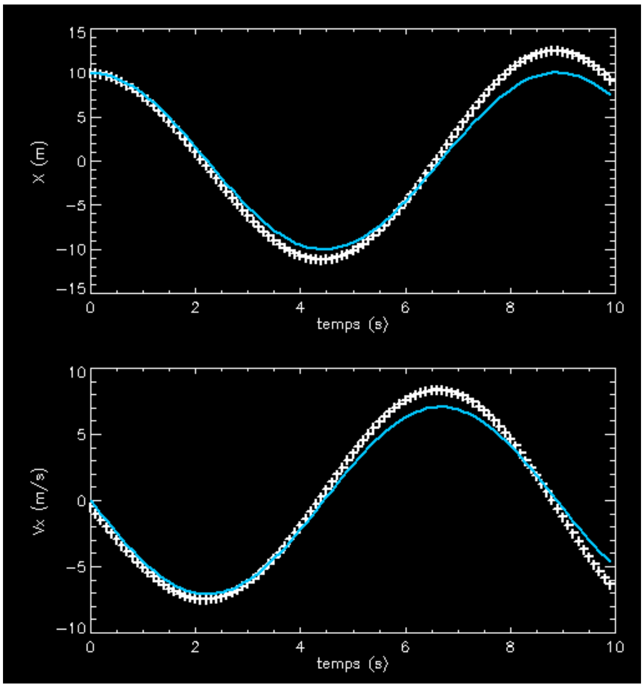
dt=0.01 s



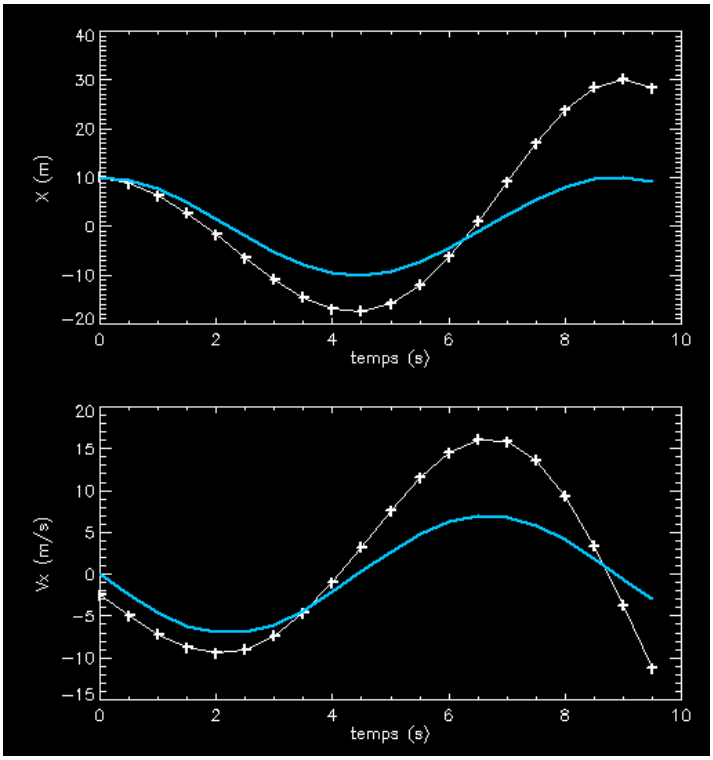
Blanc: sol numérique
bleu : sol analytique

Mais la solution numérique dépend du pas de temps

dt=0.1 s



Plus dt est grand plus l'erreur augmente !!



dt=0.5 s

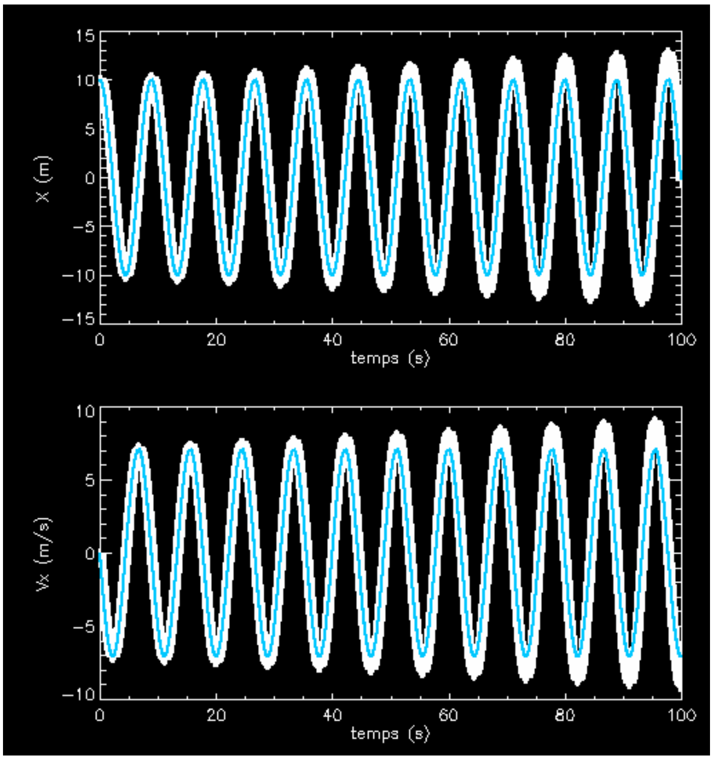
POUR TOUT INTEGRATEUR :

Toute solution numérique
n'est qu'approximative

La précision dépend du pas
d'intégration

Plus le pas est grand, plus
le calcul est rapide,
MAIS moins il est précis

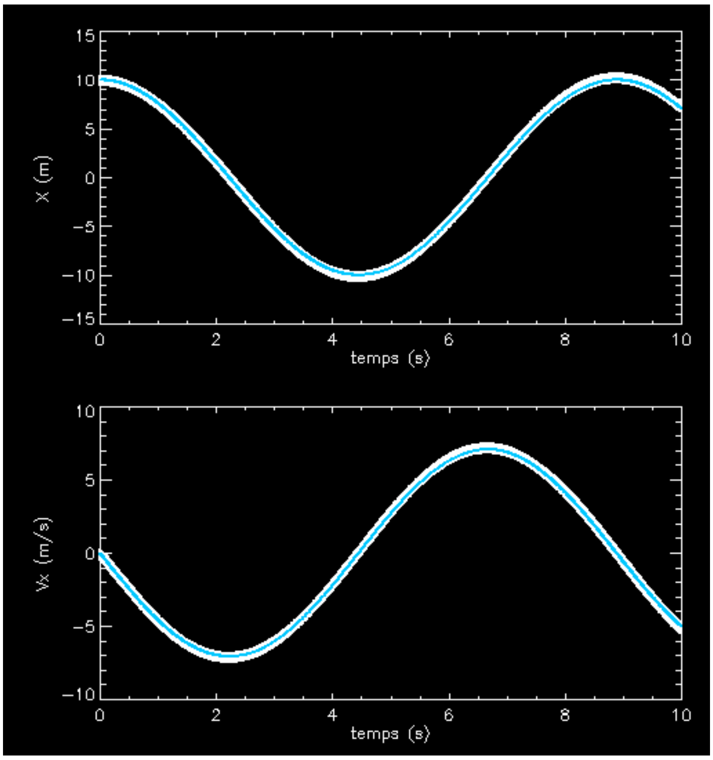
Et inversement …



Dt=0.01 s

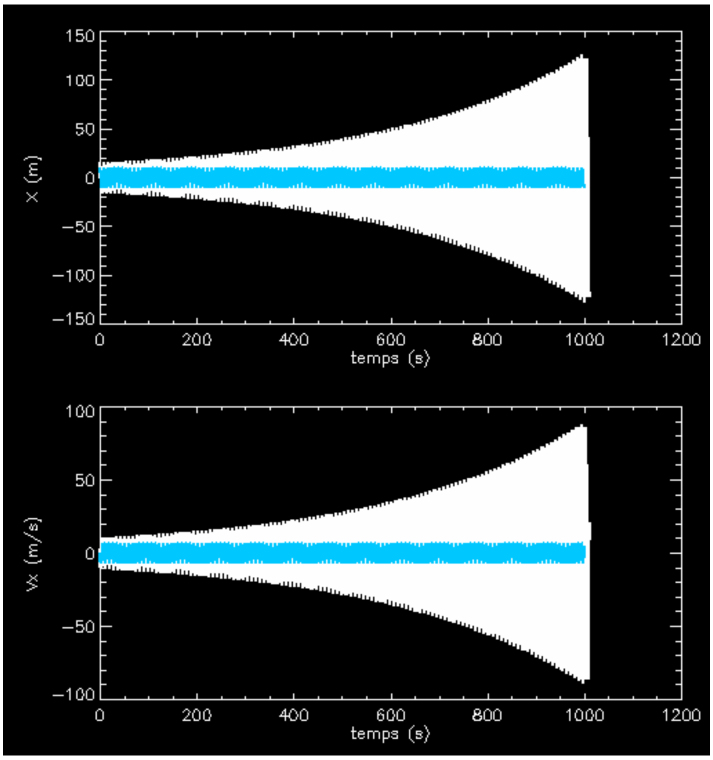
10000 étapes

La précision diminue avec le nombre d'étapes de calculs :



Dt=0.01 s

1000 étapes de calcul



Dt=0.01s

100000 étapes

Tout intégrateur finit
par s'éloigner de la solutions
quand le nombre d'étapes
de calcul augmente

Comment construire un intégrateur ?

Contraintes : - précis
- stable
- rapides

Mauvaise nouvelle : pas de solution universelle!

Construction d'un intégrateur : méthodes de base

L'intégrateur dépend du problème envisagé

Nous étudierons dans un premiers temps les équations à un seul paramètres (le temps souvent)

On les appelle les **équations différentielles ordinaires (ODE)**

Une équation différentielle ordinaire peut toujours s'écrire comme un ensemble d'équations différentielles du premier ordre

quantités

$$\frac{d(x,y,z,u,w....)}{dt} = f(t,x,y,z,u,w....)$$

Paramètre (unique)

Ecriture vectorielle

$$\frac{d}{dt}\begin{pmatrix} x \\ y \\ z \\ u \\ ... \end{pmatrix} = \begin{pmatrix} f_x(t,x,y,z,u,....) \\ f_y(t,x,y,z,u,....) \\ f_z(t,x,y,z,u,....) \\ f_u(t,x,y,z,u,....) \\ ... \end{pmatrix}$$

Exemple : le ressort (calculer*)

$$\frac{d}{dt}\begin{pmatrix} x \\ v \end{pmatrix} = \begin{pmatrix} v \\ -kx/m \end{pmatrix}$$

Note : ici le paramètre t n'intervient pas explicitement dans f car la force ne dépend pas explicitement du temps

F_x=v et F_v= -kx/m

Véritable équation :

$$\frac{d}{dt}\begin{pmatrix} x \\ y \\ z \\ u \\ ... \end{pmatrix} = \underbrace{\begin{pmatrix} f_x(t,x,y,z,u,....) \\ f_y(t,x,y,z,u,....) \\ f_z(t,x,y,z,u,....) \\ f_u(t,x,y,z,u,....) \\ ... \end{pmatrix}}_{\textit{fonction } f}$$

Résolution numérique

X_{n+1}= X à l'instant t_{n+1}

où t_{n+1} = (n+1) x dt

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \\ z_{n+1} \\ u_{n+1} \\ ... \end{pmatrix} = \underbrace{\begin{pmatrix} F_x(t_n, x_n, y_n, z_n, u_n,....) \\ F_y(t_n, x_n, y_n, z_n, u_n,....) \\ F_z(t_n, x_n, y_n, z_n, u_n,....) \\ F_u(t_n, x_n, y_n, z_n, u_n,....) \\ ... \end{pmatrix}}_{\textit{fonction } F}$$

f est la dérivée, F est l'intégrateur

L'outil de base pour construire **F** est le développement de Taylor :

$$X(t+dt) = X(t) + dt \cdot f(x,t) + \frac{dt^2}{2!} f'(x,t) + \frac{dt^3}{3!} f''(x,t) + ...$$

En pratique on ne connaît que **f**. Le but de tout intégrateur est d'estimer le mieux possible le développement de X en ne connaissant que **f**…

C'est possible !

Soit une quantitté X.

En utilisant le développement de taylor on a :

$$X_{n+1} = X(t+dt)$$

$$X(t+dt) \approx X(t) + dt\frac{dX}{dt} + \frac{dt^2}{2}\frac{d^2X}{dt^2} + ...$$

Fonction f, équa. diff du système

En s'inspirant de ce développement, la fonction F sera :

$$X_{n+1} = X_n + dt\ D(t, X_n)$$

Où D sera un approximation numérique de la dérivée !!

Comment construire D :

Cas le plus simple :

METHODE d'EULER : D(x,t) = f(x,t)

La fonction F (x,t) est alors : $\mathbf{X_{n+1}=X_n+dt\ f(x,t)}$

Exemple : Cas du ressort avec Euler

$$\begin{pmatrix} x_{n+1} \\ v_{n+1} \end{pmatrix} = \begin{pmatrix} F_x(t_n, x_n, v_n) \\ F_v(t_n, x_n, v_n) \end{pmatrix} = \begin{pmatrix} x_n + dt\ D_x(t_n, x_n, v_n) \\ v_n + dt\ D_v(t_n, x_n, v_n) \end{pmatrix} = \begin{pmatrix} x_n + dt\ v \\ v_n + dt\ \dfrac{-kx}{m} \end{pmatrix}$$

Le schéma d’intégration est donc tout simplement

$$\begin{pmatrix} x_{n+1} \\ v_{n+1} \end{pmatrix} = \begin{pmatrix} x_n + dt\ v \\ v_n + dt \dfrac{-kx}{m} \end{pmatrix}$$

Avec t_n= n dt

Le shéma d’Euler est le plus simple possible.

$D(t_n , X_n)=f(t_n ,X_n)$

C’est un intégrateur d’ordre 1 (car entre t et t+dt l’erreur est un *$o(dt^1)$*

C’est un schéma rapide car il y a un seul appel à la dérivée f

En pratique : jamais utilisé

Mais on peut faire beaucoup mieux !

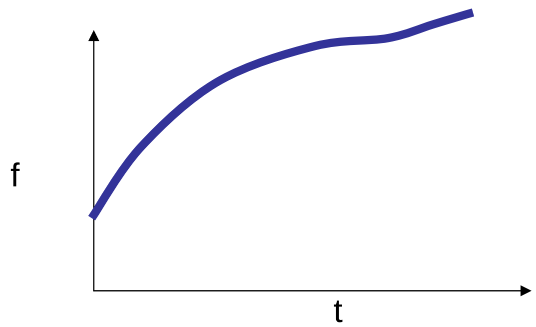
Construire un meilleur schémas numérique en s’inspirant de l’intégration

On sait que la résolution *exacte* de X_{n+1} est :

$$X_{n+1} = X_n + dt\ D(t,x) = X_n + \int_t^{t+dt} f(t, X(t),...)dt$$

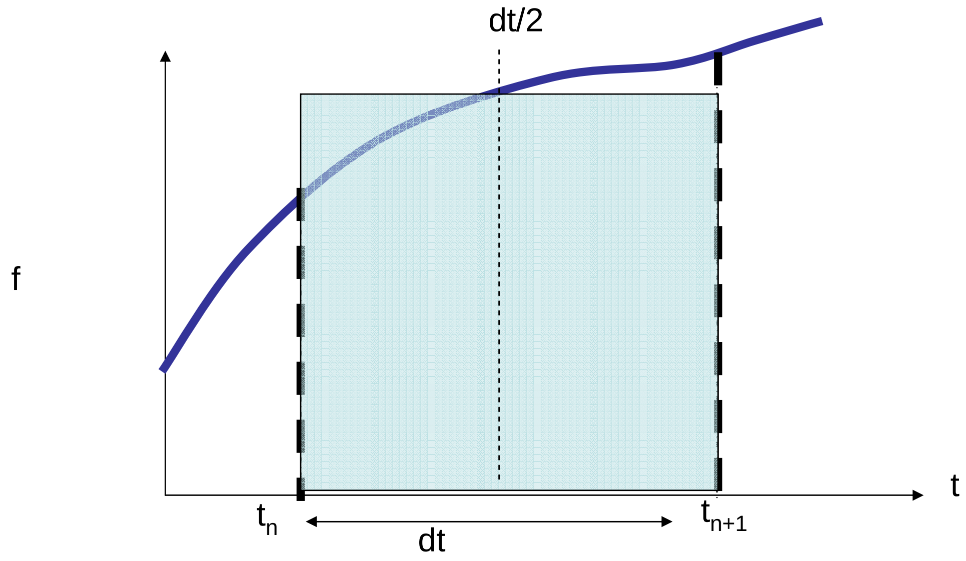
donc

$$D(t,x) = \frac{\int_t^{t+dt} f(t, X(t),...)dt}{dt}$$



D est l’aire sous la courbe divisée par dt

Idée : approximer D par la méthode des trapèzes, méthode du point milieu



D ~ (Aire bleue)/dt

$$\sim \frac{dt * f\left(t+\frac{dt}{2}, X(t+\frac{dt}{2})\right)}{dt}$$

$$D(t,x) \sim \frac{dt}{dt}\, f\left(t+\frac{dt}{2}, X(t+\frac{dt}{2})\right)$$

Mais on peut encore approximer X(t+dt/2) par :

$$X(t+dt/2) \sim X(t) + dt/2 * f(t,x)$$

On obtient alors un nouveau schéma d’intégration : ***Euler modifié***

$$X_{n+1} = X_n + dt\ D(X_n, t)$$
$$= X_n + dt\ f\left(t+\frac{dt}{2}, X_n + \frac{dt}{2} f(t, X_n)\right)$$

Qu’on peut encore écrire

$$X_{n+1} = X_n + dt\ f\left(t+\frac{dt}{2}, k_1\right)$$

où $k_1 = X_n + \dfrac{dt}{2} f(t, X_n)$

Euler modifié est un intégrateur plus précis !!

Montrons que Euler modifié est un intégrateur d'ordre 2.
On ne considère que la dépendance en temps t

Le schéma est : $$X_{n+1} = X_n + dt\ f\left(t_n + \frac{dt}{2}\right)$$

$$Or\ f\left(t_n + \frac{dt}{2}\right) \approx f(t_n) + \frac{dt}{2} f'(t_n) + \frac{dt^2}{8} f''(t_n) + \ldots$$

$$Donc$$

$$X_{n+1} = X_n + dt\ f(t) + \frac{dt^2}{2} f'(t) + \frac{dt^3}{8} f''(t_n) + \ldots$$

On calcule donc

$$X_{n+1} = X_n + dt\ f(t) + \frac{dt^2}{2} f'(t) + \frac{dt^3}{8} f''(t_n) + \ldots$$

Or le vrai développement de Taylor de X_n est (sachant que dX/dt=f)

$$X_{n+1} = X_n + dt\ f(t) + \frac{dt^2}{2} f'(t) + \frac{dt^3}{6} f''(t_n) + \ldots$$

On voit donc que Euler Modifié est juste jusqu'à l'ordre 2

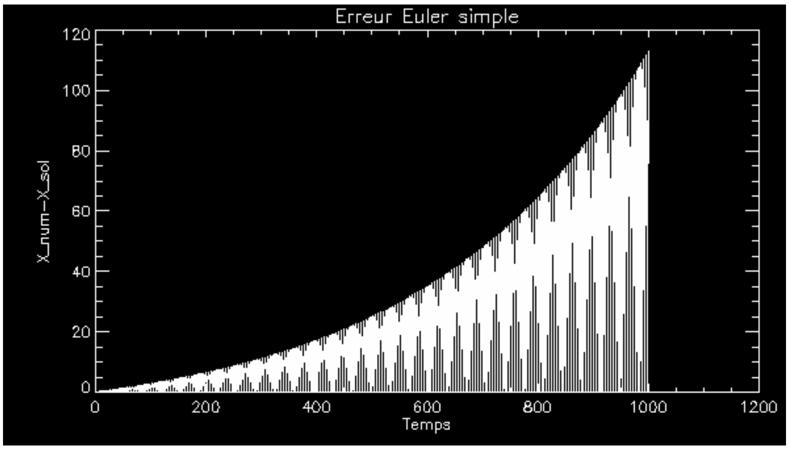
La méthode d'Euler modifiée est précise jusqu'à l'ordre 2

Application :

Exemple du ressort

Abs(X_vrai-X_approx)

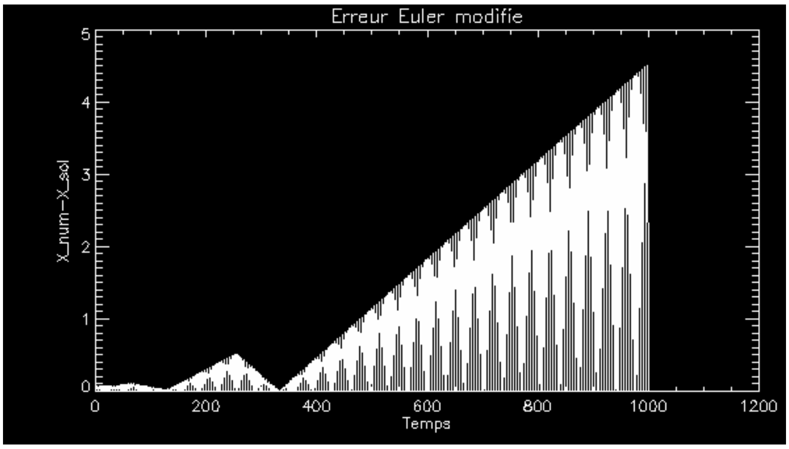
Euler simple :
Dt=0.01



Euler modifié :

Gain de précision :

Un facteur 20 !!



De nombreux autres schémas d'intégration existent !!

Les schémas que nous avons vus sont dit **intégrateurs explicites** :

Euler : $X_{n+1}= X_n+ dt\ f(t, X_n)$

Euler modifié : Xn+1= Xn+ dt f(t+dt/2, f(Xn+dt/2))

Car **Xn+1 ne dépend que de des indices précédents (n, n-1, etc..).** Ils se calculent de manière directe.

Une famille d'intégrateur importante est celle des **intégrateurs implicites** :

EXEMPLE : Cranck Nicholson

Exemple : Cranck Nicolson

$$X_{n+1} = X_n + \frac{dt}{2}\left[f(t_n, X_n) + f(t_{n+1}, X_{n+1}) \right]$$

Xn+1 dépend de Xn+1 et Tn+1 => Schéma implicite.

Il faut résoudre une équation non linéaire pour déterminer Xn+1

Les intégrateures implicites sont souvent moins précis
‘’’ plus compliqués à utiliser

MAIS

Ils sont plus stables que les intégrateurs explicites : ils divergent moins rapidement.

En d’autres termes : moins précis à court terme, plus précis à long termes

Catalogue des intégrateurs d’ODE les plus courants.

L’ordre de l’intégrateur est entre parenthèses

Euler explicite (1)	$U_{n+1} = U_n + dt \cdot f(t_n, U_n)$
Euler implicite (1)	$U_{n+1} = U_n + dt \cdot f(t_{n+1}, U_{n+1})$
Leap Frog (2)	$U_{n+1} = U_{n-1} + 2dt \cdot f(t_n, U_n)$
Euler modifié (2)	$U_{n+1} = U_n + dt \cdot f\left(t_n + \frac{dt}{2}, U_n + \frac{dt}{2} f(t_n, U_n) \right)$
Cranck Nicholson(2)	$U_{n+1} = U_n + \frac{dt}{2} \cdot \left(f(t_n, U_n) + f(t_{n+1}, U_{n+1}) \right)$
Adam Bashfort (2)	$U_{n+1} = U_n + dt \cdot \left(\frac{3}{2} f(t_n, U_n) - \frac{1}{2} f(t_{n-1}, U_{n-1}) \right)$
Adam Bashfort (3)	$U_{n+1} = U_n + dt \cdot \left(\frac{23}{12} f(t_n, U_n) - \frac{16}{12} f(t_{n-1}, U_{n-1}) + \frac{5}{12} f(t_{n-2}, U_{n-2}) \right)$
Adam Moulton (3)	$U_{n+1} = U_n + dt \cdot \left(\frac{5}{12} f(t_{n+1}, U_{n+1}) + \frac{8}{12} f(t_n, U_n) - \frac{1}{12} f(t_{n-1}, U_{n-1}) \right)$

Runge Kutta (3)

$$\begin{cases} k_1 = dt \cdot f(t_n, U_n) \\ k_2 = dt \cdot f(t_n + dt, U_n + k1) \\ U_{n+1} = U_n + \frac{1}{2}(k_1 + k_2) \end{cases}$$

Runge Kutta (4)

$$\begin{cases} k_1 = dt \cdot f(t_n, U_n) \\ k_2 = dt \cdot f(t_n + \frac{dt}{2}, U_n + \frac{k_1}{2}) \\ k_3 = dt \cdot f(t_n + \frac{dt}{2}, U_n + \frac{k_2}{2}) \\ k_4 = dt \cdot f(t_n + dt, U_n + k_3) \\ U_{n+1} = U_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{cases}$$

Stabilité d'un intégrateur

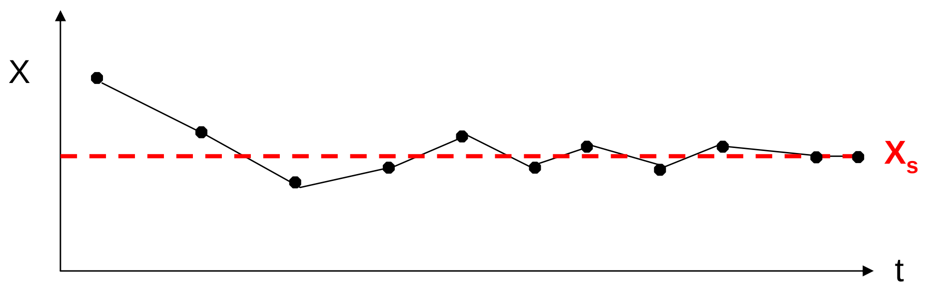
Pour un intégrateur donné, il est important de **quantifier sa stabilité**, en d'autres termes ses conditions de convergence.

Soit un un intégrateur type : $\mathbf{X_{n+1}=X_n+dt\ F\ (t_n,X_n)}$

Pour quantifier la **stabilité**, **on se place au voisinage d'un point stationnaire Xs**

Imaginons que X soit proche de **Xs.**

Mesurons la vitesse à laquelle on *s'éloigne/approche* de la solution.



Soit $\mathbf{e_{n+1}=X_{n+1}-X_n}$ l'erreur sur la solution à l'étape n. On suppose e_n<<Xn

Le schéma numérique va-t-il amplifier l'erreur ??

$X_{n+1} = F(X_n)$ que l'on essaye d'écrire = X_n+dt x F'x X_n= A X_n
(F'=linéarise autour de X_s)

Or

$$X_{n+1} = F(X_n) = (1 + dt \cdot F')X_n = AX_n$$
$$e_{n+1} = X_{n+1} - X_n \quad \Rightarrow$$
$$e_{n+1} = AX_n - AX_{n-1} \quad \Rightarrow$$
$$e_{n+1} = A(X_n - X_{n-1}) \quad \Rightarrow$$
$$e_{n+1} = Ae_n$$

$$\boxed{1 + F'}$$

Est la **matrice d'amplification (A)**

Conditions de stabilité ?

Dans la base propre

$$\begin{pmatrix} ex'_{n+1} \\ ey'_{n+1} \\ ez'_{n+1} \end{pmatrix} = \begin{pmatrix} v_1 & & 0 \\ & v_2 & \\ 0 & & v_3 \end{pmatrix} \cdot \begin{pmatrix} ex'_n \\ ey'_n \\ ez'_n \end{pmatrix} \Rightarrow$$
$$\begin{pmatrix} ex'_{n+1} \\ ey'_{n+1} \\ ez'_{n+1} \end{pmatrix} = \begin{pmatrix} v_1 \cdot ex'_n \\ v_2 \cdot ey'_n \\ v_3 \cdot ez'_n \end{pmatrix}$$

Donc pour qu'il y ait STABILITE (⇔ pas d'amplification des erreurs) il faut que toutes les valeurs propres de la matrice d'amplification A soient <1 en valeur absolue.

Soit V=MAX[abs(v1), abs(v2), abs(v3)]. V dépend du pas de temps dt
- Si V< 1 pour dt<dt_{max} alors le schéma est *conditionnellement stable*
- *S*i V<1 pour tout dt, alors le schéma est *inconditionnellement stable*
- *S*i V>1 pour tout dt, alors le schéma est *inconditionnellement instable*

Ecriture matricielle. Exemple avec un système à 3 variables : X, Y et Z. Soient ex, ey et ez les erreurs en X, Y et Z

$$\begin{pmatrix} ex_{n+1} \\ ey_{n+1} \\ ez_{n+1} \end{pmatrix} = A \begin{pmatrix} ex_{n} \\ ey_{n} \\ ez_{n} \end{pmatrix} \Rightarrow$$

Si A est diagonalisable
$A=m^{-1}$ D m,
où m est la matrice de passage
et D la matrice diagonale

$$\begin{pmatrix} ex_{n+1} \\ ey_{n+1} \\ ez_{n+1} \end{pmatrix} = M^{-1} D M \begin{pmatrix} ex_{n} \\ ey_{n} \\ ez_{n} \end{pmatrix} \Rightarrow$$

$$M \underbrace{\begin{pmatrix} ex_{n+1} \\ ey_{n+1} \\ ez_{n+1} \end{pmatrix}} = DM \underbrace{\begin{pmatrix} ex_{n} \\ ey_{n} \\ ez_{n} \end{pmatrix}}$$

Vecteurs dans la base propre de A

Exemple avec le ressort + méthode d'Euler EXPLICITE

$$\begin{cases} x_{n+1} = x_n + dt\, v_n \\ v_{n+1} = v_n + dt \dfrac{-kx_n}{m} \end{cases} \Rightarrow$$

Forme : $X_{n+1}=AX_n$

$$\begin{pmatrix} x_{n+1} \\ v_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & dt \\ \dfrac{-kdt}{m} & 1 \end{pmatrix} \cdot \begin{pmatrix} x_{n} \\ v_{n} \end{pmatrix} \Rightarrow$$

$$A = \begin{pmatrix} 1 & dt \\ \dfrac{-kdt}{m} & 1 \end{pmatrix}$$

$$\boxed{\begin{pmatrix} 1 & dt \\ \dfrac{-kdt}{m} & 1 \end{pmatrix}}$$

Matrice d'amplification

CALCULER LES VALEURS PROPRES

Valeurs propres λ : λ^2- λ trace + déterminant = 0 =>

$$\lambda^2 - 2\lambda + \left(1 - \frac{kdt^2}{m}\right) = 0 \Rightarrow$$

$$\Delta = 4 - 4\left(1 - \frac{kdt^2}{m}\right) = \frac{4kdt}{m} \Rightarrow$$

$$\lambda = \frac{2 \pm \sqrt{\frac{4kdt^2}{m}}}{2} = 1 \pm dt\sqrt{\frac{k}{m}}$$

La plus grande des 2 valeurs propres est toujours $1 + dt\sqrt{\frac{k}{m}}$

Donc Euler appliqué au ressort est *inconditionnellement instable (dt>0)*

Exemple avec le ressort + méthode d'Euler IMPLICITE

$$\begin{cases} x_{n+1} = x_n + dt\, v_{n+1} \\ v_{n+1} = v_n + dt\, \dfrac{-kx_{n+1}}{m} \end{cases} \Rightarrow$$

Substitution

$$\begin{cases} x_{n+1} = x_n + dt\left(v_n + dt\, \dfrac{-kx_{n+1}}{m}\right) \\ v_{n+1} = v_n + dt\, \dfrac{-k}{m}\left(x_n + dt\, v_{n+1}\right) \end{cases} \Rightarrow$$

$$\begin{cases} x_{n+1} = \left(x_n + dt\, v_{n+1}\right) \cdot \dfrac{1}{\alpha} \Rightarrow \\ v_{n+1} = \left(v_n + dt\, \dfrac{-kx_{n+1}}{m}\right) \cdot \dfrac{1}{\alpha} \end{cases}$$

où $\alpha = 1 + \frac{kdt^2}{m}$

$$\boxed{\frac{1}{\alpha}\begin{pmatrix} 1 & dt \\ -\dfrac{kdt}{m} & 1 \end{pmatrix}}$$

Matrice d'amplification

$$\begin{pmatrix} x_{n+1} \\ v_{n+1} \end{pmatrix} = \frac{1}{\alpha}\begin{pmatrix} 1 & dt \\ -\dfrac{kdt}{m} & 1 \end{pmatrix} \cdot \begin{pmatrix} x_n \\ v_n \end{pmatrix}$$

Forme : $X_{n+1}=AX_n$

CALCULER LES VALEURS PROPRES

Exemple avec le ressort + méthode d'Euler IMPLICITE

$$\begin{cases} x_{n+1} = x_n + dt\ v_{n+1} \\ v_{n+1} = v_n + dt\ \dfrac{-kx_{n+1}}{m} \end{cases}$$

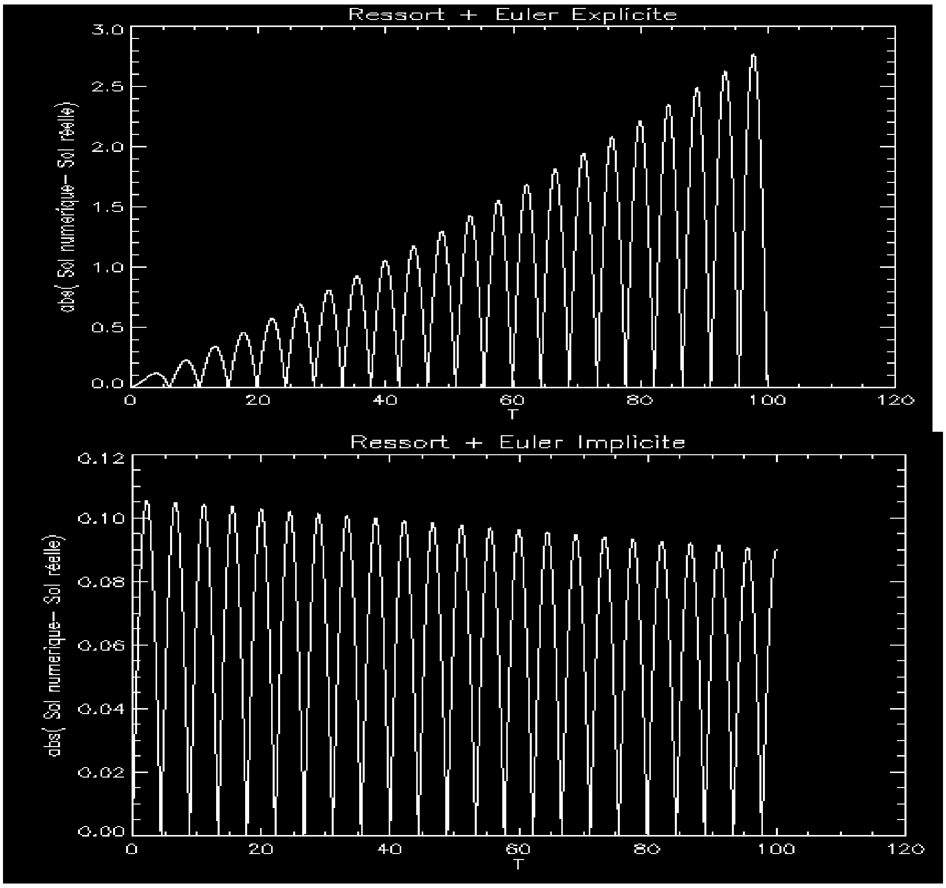
Calculer la matric d'amplification A
(A telle que $X_{n+1}=AX_{n}$)

Les valeurs propres sont : $$\frac{1}{\alpha}\left(1 \pm dt\sqrt{\frac{k}{m}}\right) = \frac{1 \pm dt\sqrt{\frac{k}{m}}}{1 + \frac{kdt^2}{m}}$$

Elles sont toujours < 1 DONC

Avec le problème du ressort, Euler Implicite est :
Inconditionnellement stabe

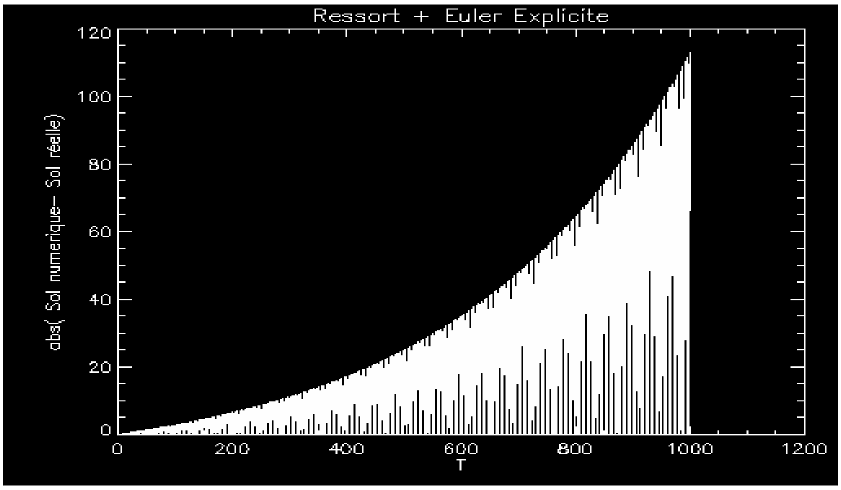
Comparaison Euler Explicite Vs. Implicite : Cas du ressort

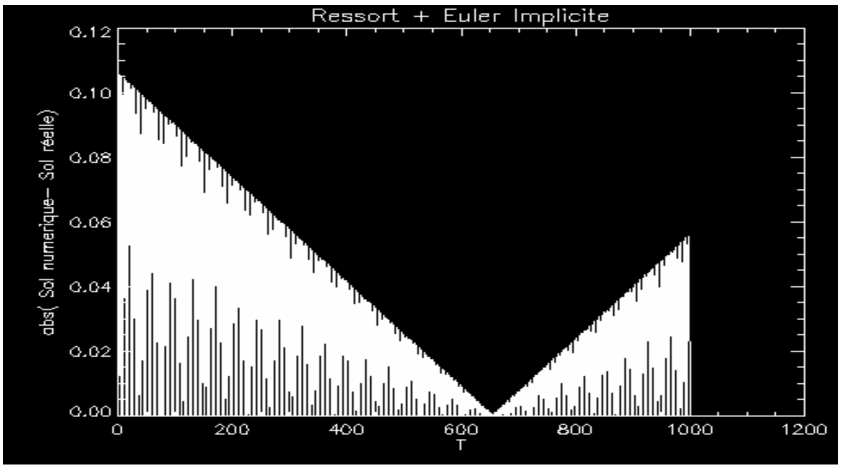


Erreur Explicite

Dt=0.01
10000 itérations

Erreur Implicite



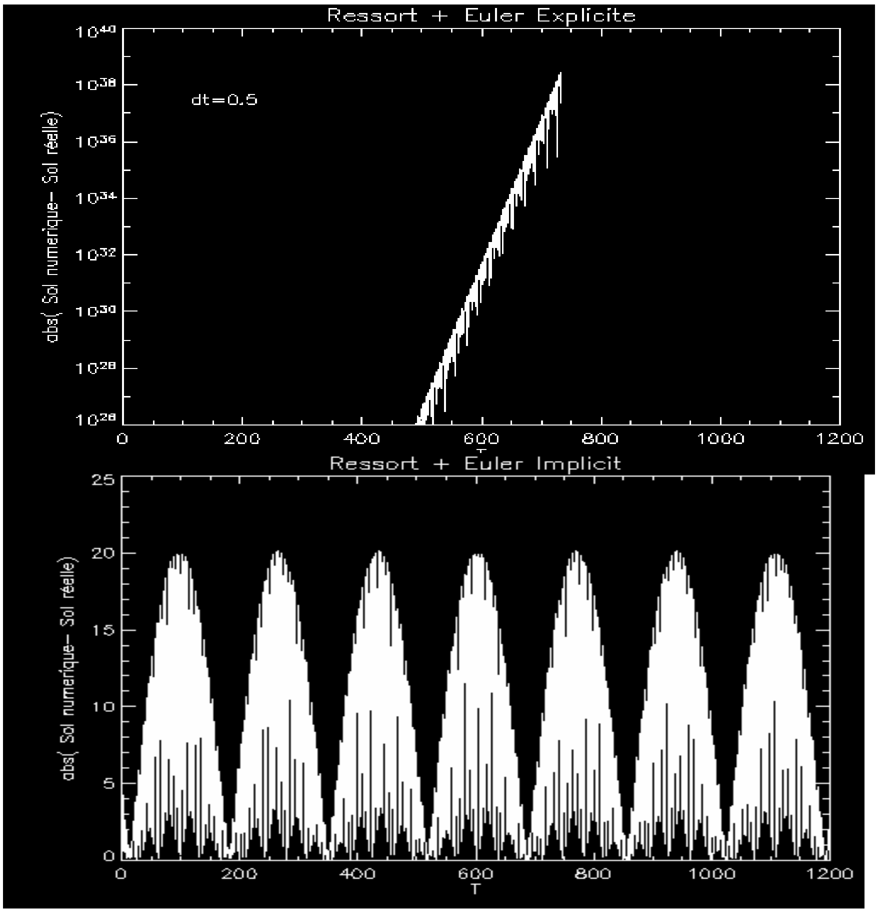


Erreur Explicite

Dt=0.01
100000 itérations

Erreur Implicite

Pas de temps très très grand : dt=0.5 (au lieu de 0.01)



Erreur Euler Explicite

…. dans les choux…

100000 itérations

Erreur Euler Implicite

L'erreur n'augmente pas
… mais le résultat est
quand même faux

L'intégrateur implicite est un peu moins précis au tout début

MAIS

A long terme , il ne s'éloigne pas trop de la solution

DONC

Quand la stabilité est importante, il est intéressant d'utiliser un intégrateu
implicité.

Quel intégrateur choisir ?

⇒Equilibre entre le temps de calcul, précision demandée et la stabilité

Ordre faible (1,2) explicite
Rapide, peu stable, peu précis

Ordre faible (1,2) implicite
Moyennement rapide, très stable, précis

Ordre élevé (3,4, +) explicite
Lent, stable, assez précis

Ordre élevé (3,4,+) implicite
Tres lent, tres stable, très précis MAIS quasiment jamais utilisés….

Quel pas de temps choisir ?

Règle générale : dt << temps dynamique du système

Il faut **toujours** bien tester le pas de temps, en contrôlant certains paramètres.

(Exemple typique : utiliser l'ENERGIE du système, vérifier sa conservation)

Pourquoi ? Prenons un système à 1 variable , X

Supposons que nous avons un point d'équilibre stable tel que $X_{n+1} \sim X_n \sim X_S$

On aura $X_{n+1} \sim X_n + dt \times dXs/dt \Rightarrow (X_{n+1} - X_n) \sim dt \times dXs/dt << 1$

⇒ **$dt << (d\,Xs/dt)^{-1}$**

$(dX/dt)^{-1}$ est un temps caractéristique d'évolution du système
(inverse de la dérivée)

Exemple :

Pour le ressort , le temps caractéristique est la période d'oscillation,

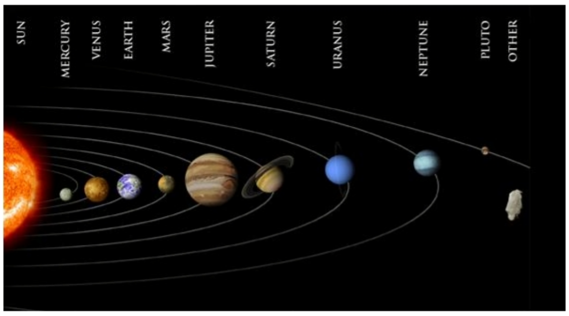
$$T = \frac{2\pi}{\omega} = 2\pi\sqrt{\frac{m}{k}}$$

On prendra donc dt << T

Dans notre exemple : m=1, k=1 => T=6,28 secondes.

Avec dt=0.01 s OK
Avec dt=0.5 s   PB !! (voir figures précédentes)

Que faire quand il y a plusieurs temps caractéristiques très différents ?
=>problème *courant* et très mal résolu….



Exemple : Le Système Solaire :

Période orbitale de
Mercure :   88 jours
Terre :   1 an
Jupiter :   12 ans
Pluton :   248 ans

TOUTES les planètes intéragissent (couplage)

Dans ce système, il a un facteur 1000 entre le temps dynamique le plus court et le plus long…

Que choisir pour dt ??

On a pas le choix : dt << 88 jours…

Conclusion : la majorité du temps de calcul sert juste à intégrer Mercure

Une fausse bonne idée à éviter :

Donner un pas de temps différent à chaque planète …

$\Rightarrow$Le Résultat sera invariablement faux car les différentes variables du systèmes ne seront pas SYNCHRONISEES !!

Par exemple:

Mercure : dt , 2dt, 3dt , 4 dt , 5 dt, 6 dt, 7 dt, 8 dt, 9 dt 10 dt

Terre : 2dt 4dt 6dt 8 dt 10 dt

Jupiter: 3dt 6dt 9dt

Plunto: 5dt 10dt

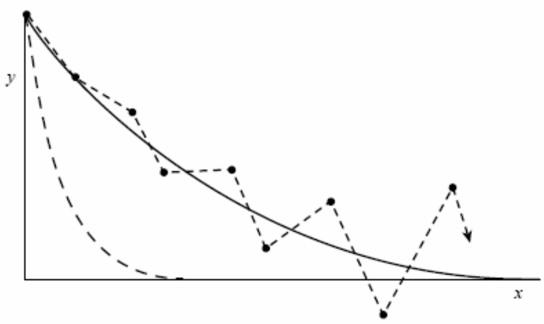
PB: Pour Mercure on a besoin de connaître la position de TOUTES les planètes à chaque valeur de dt…

Autre exemple : Un système à 2 variables

$$\begin{cases} u' = 998u + 1998v \\ v' = -999u - 1999v \end{cases} \Rightarrow \begin{cases} u = 2e^{-t} - e^{-1000t} \\ v = -e^{-t} + e^{-1000t} \end{cases}$$

Avec U(0)=1, V(0)=0

2 echelles de temps : 1 et 1/1000



Une méthode explicite va osciller entre les deux exp., mêmes après que la plus rapide soit~0.

Solution : Méthode implicite
Dont le domaine de stabilité est infini.
Mais peu précise…

Ce type de problème est dit « raide » (STIFF en anglais)

CONCLUSION POUR UN PROBLEME « raide »

Certaines méthodes existent , assez subtiles, pour utiliser un pas de temps indépendant pour chaque variables…
⇒Souvent cela n'est pas très efficace

Dans un système où TOUTES les variables sont couplées les unes aux autres, il faut que dt << plus petit temps dynamique.

Pour éviter de trop grandes instabilités utiliser un intégrateur IMPLICITE : peu précis mais STABLE

Un intégrateur « populaire » : le Runge Kutta 4 (RK4)

$$\begin{cases} k_1 = dt \cdot f(t_n, U_n) \\ k_2 = dt \cdot f(t_n + \frac{dt}{2}, U_n + \frac{k_1}{2}) \\ k_3 = dt \cdot f(t_n + \frac{dt}{2}, U_n + \frac{k_2}{2}) \\ k_4 = dt \cdot f(t_n + dt, U_n + k_3) \\ U_{n+1} = U_n + \frac{1}{6}(k_1 + k_2 + k_3 + k_4) \end{cases}$$

Intégrateur EXPLICITE donc facile à mettre en oeuvre
« relativement ».. Stable

Relativement « lent » car 4 appels à la dérivée.

Principe

RK4 :
4 évaluations à l’intérieur
du pas de temps

Ordre 4

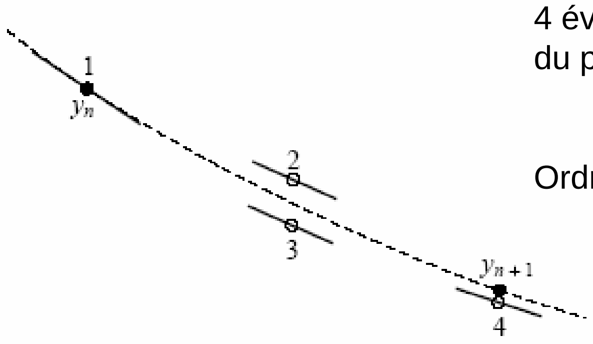
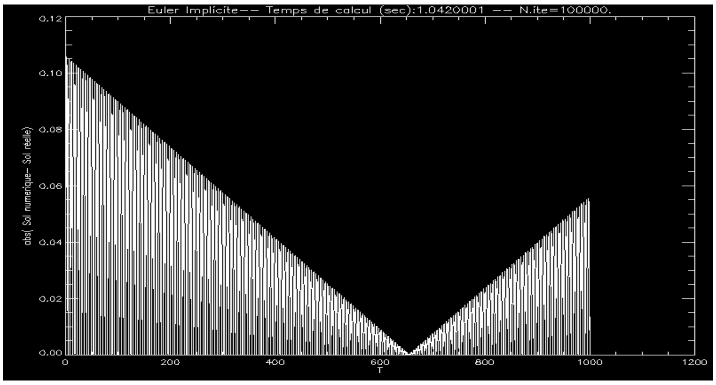
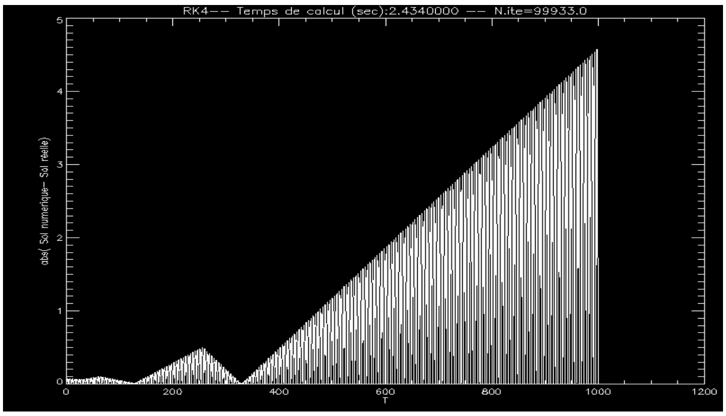


Figure 16.1.3. Fourth-order Runge-Kutta method. In each step the derivative is evaluated four times: once at the initial point, twice at trial midpoints, and once at a trial endpoint. From these derivatives the final function value (shown as a filled dot) is calculated. (See text for details.)

Le Runge Kutta est-il plus précis que Euler implicit ?
A priori OUI, mais pas toujours !!! Ex: ressort



Euler Implicite



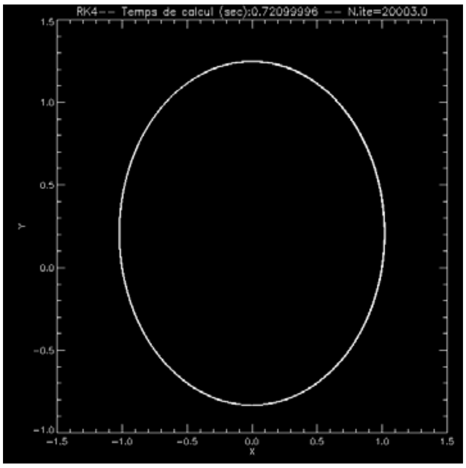
RK4

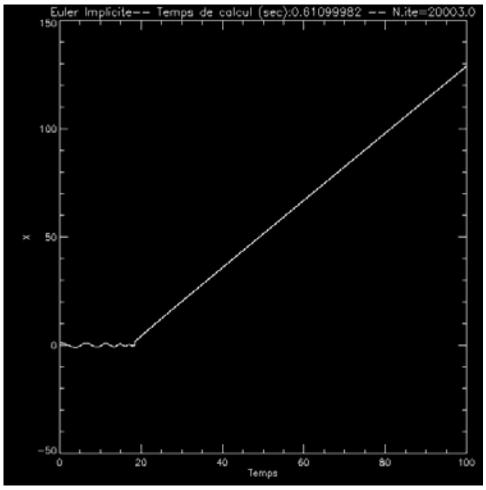
Le RK4 est moins bon

⇒En fait l’équation implicite est mieux adaptée à ce problème simple d’ordre faible.

⇒ Moins d’appels à la dérivée
⇒ moins d’erreurs d’arrondis

Prenons un système plus complique :
Le mouvement d'une planète autour du Soleil:
A=- GM u/r^2
Dt=0.1, Temps Dynamique = 2PI

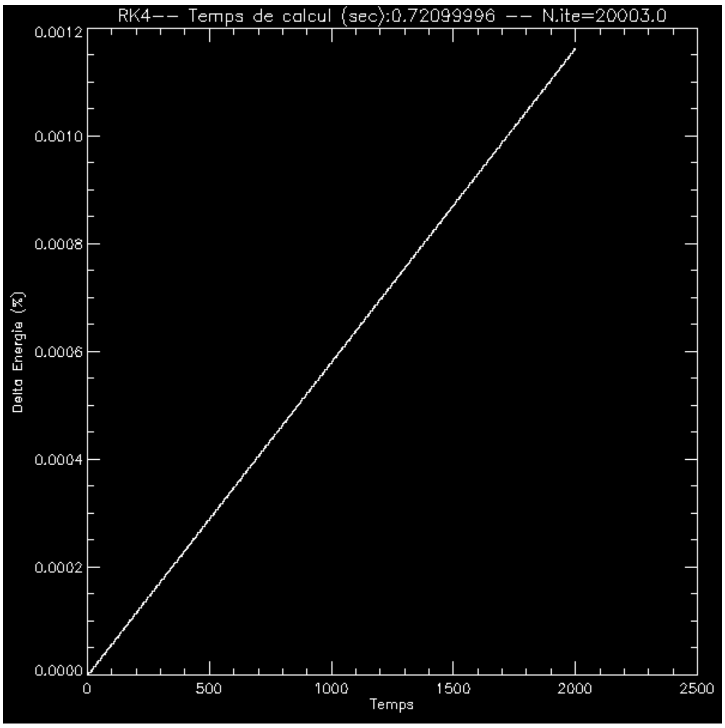




RK4
Tout se passe bien à-priori

Euler Implicite
Instabilité numérique !!
car pas de sol analytique à $U_{n+1}=U_n+dtf(U_{n+1})$

Un paramètre de contrôle simple : L'énergie
L'energie totale doit être conservée : $E=1/2\ mV^2 - Gm/r$



dt=0.1
1 orbite= 2PI

L'énergie a artificiellement augmentée de 0,12 % en 250 orbites (250 temps dynamiques)

⇒On ne pourra croire le résultat au delà de 250000 orbites (1000 fois plus) car $\Delta E \sim E$ au bout de ce temps

Vers les méthodes adaptatives :
Le RK4 adaptatif

Idée : comment contrôler dt pour être sur que l'on l'erreur ne soit pas trop grande

Plusieurs méthodes existent, Une méthode à pas de temps adaptatif est plus complexe à mettre en place , mais souvent plus rapide et plus précise.
Nécéssite de BIEN connaître la physique du système

Difficulté

Comme on ne connaît pas *à priori* la solution exacte, il est difficile d'estimer l'erreur.

Une méthode usuelle est de réaliser que si le calcul est faux, ou très approximé,
La solution trouvée par l'intégrateur devrait dépendre *très* fortement de la valeur de dt.

Pourquoi ? Par ce que : Lim (F(Xn)) = Solution , quand dt -> 0
Donc quand on est loin de la solution (contraposée) F dépend fortement du pas de temps.

Idée : Comparer différentes évaluations de l'intégrateur , soit en fonction du pas de temps, soit en fonction de l'Ordre de l'intégrateur.

Il faut introduire un paramètre de précision, Δ_0 **, la précision désirée**

1ère technique : Faire 2 évaluations du résultat, en prenant dt et dt/2.
(double le temps de calcul) . Si les deux résultats sont égaux plus ou moins Δ_0
Alors la solution est acceptable, sinon il faut diminuer le pas de temps.

Méthode simple mais très couteuse en temps :

Combiens d' évaluations ?
4 pour le pas de temps à dt
8 pour 2 pas de temps à dt/2 MAIS la 1ere à dt/2 est la même que celle à dt
donc 12 au total.

A comparer avec 8 evaluations (on avance avec dt/2)

Donc une augmentation du temps de calcul de 11/8~ 1.4
40% plus lent

2ème méthode : Plus élégante et plus rapide : Runge Kutta 5 adaptatif
Méthode de Fehlberg pour le Runge Kutta

Felhberg a étudié le RK5. Il nécessite 6 appels à la dérivée

Le RK5 sécrit d'une manière générale

$$k_1 = hf(x_n, y_n)$$
$$k_2 = hf(x_n + a_2 h, y_n + b_{21} k_1)$$
$$\cdots$$
$$k_6 = hf(x_n + a_6 h, y_n + b_{61} k_1 + \cdots + b_{65} k_5)$$
$$y_{n+1} = y_n + c_1 k_1 + c_2 k_2 + c_3 k_3 + c_4 k_4 + c_5 k_5 + c_6 k_6 + O(h^6)$$

Le résultat est d'ordre 5

Mais Fehlberg à découvert q'une autre combinaison de coefficients donne un résultat d'ordre 4

Ordre 5

$$y_{n+1} = y_n + c_1 k_1 + c_2 k_2 + c_3 k_3 + c_4 k_4 + c_5 k_5 + c_6 k_6 + O(h^6)$$

Ordre 4

$$y^*_{n+1} = y_n + c^*_1 k_1 + c^*_2 k_2 + c^*_3 k_3 + c^*_4 k_4 + c^*_5 k_5 + c^*_6 k_6 + O(h^5)$$

DONC : En calculant les mêmes quantités k1 à k6, on peut avoir deux évaluations différentes du résultat :

Y_n à l'ordre 5
Y^*_n à l'ordre 4

=> ABS (Y^*_n-Y_n) est une estimation de l'erreur à l'ordre 5

Tableau des coefficients pour le RK5

Cash-Karp Parameters for Embedded Runga-Kutta Method								
i	a_i	b_{ij}					c_i	c_i^*
1							$\frac{37}{378}$	$\frac{2825}{27648}$
2	$\frac{1}{5}$	$\frac{1}{5}$					0	0
3	$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$				$\frac{250}{621}$	$\frac{18575}{48384}$
4	$\frac{3}{5}$	$\frac{3}{10}$	$-\frac{9}{10}$	$\frac{6}{5}$			$\frac{125}{594}$	$\frac{13525}{55296}$
5	1	$-\frac{11}{54}$	$\frac{5}{2}$	$-\frac{70}{27}$	$\frac{35}{27}$		0	$\frac{277}{14336}$
6	$\frac{7}{8}$	$\frac{1631}{55296}$	$\frac{175}{512}$	$\frac{575}{13824}$	$\frac{44275}{110592}$	$\frac{253}{4096}$	$\frac{512}{1771}$	$\frac{1}{4}$
$j =$		1	2	3	4	5		

On peut utiliser un tableau de coefficients aussi pour le RK4

RK4

$$
\begin{aligned}
f_0 &= f(x_i, y_i) \\
f_1 &= f\left(x_i + \alpha_1 h, y_i + h\beta_{10} f_0\right) \\
\cdots & \quad \cdots \\
f_k &= f\left(x_i + \alpha_k h, y_i + h(\beta_{k0} f_0 + \beta_{k1} f_1 + \cdots + \beta_{k,k-1} f_{k-1}\right)
\end{aligned}
$$

$$y_{i+1} = y_i + h\left(c_0 f_0 + c_1 f_1 + \cdots + c_k f_k\right)$$

i	α_i	β_{ij}	c_i
0			$\frac{1}{6}$
1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{3}$
2	$\frac{1}{2}$	$0 \quad \frac{1}{2}$	$\frac{1}{3}$
3	1	$0 \quad 0 \quad 1$	$\frac{1}{6}$

$$
\begin{aligned}
f_0 &= f(x_i, y_i), \\
f_1 &= f(x_i + \frac{h}{2}, y_i + \frac{h}{2} f_0), \\
f_2 &= f(x_i + \frac{h}{2}, y_i + \frac{h}{2} f_1), \\
f_3 &= f(x_i + h, y_i + h f_2) \\
y_{i+1} &= y_i + \frac{h}{6}\left(f_0 + 2f_1 + 2f_2 + f_3\right).
\end{aligned}
$$

Supposons que nous avons deux estimations du résultat,
Y_n et Y^*_n

L'erreur $\Delta \sim \| Y^*_n - Y_n \| \propto dt^5$

Nous cherchons le nouveau pas de temps dt' tel que $\Delta = \Delta_0$ (précision requise)

on a donc $(dt'/dt)^5 = \Delta_0 / \Delta$

=> $dt' = dt\, (\Delta_0 / \Delta)^{1/5}$

Cela sert à deux choses :
1- Si l'erreur est trop grande le pas de temps diminue
2- Si l'erreur est trop petite le pas de temps augmente => on gagne du temps !

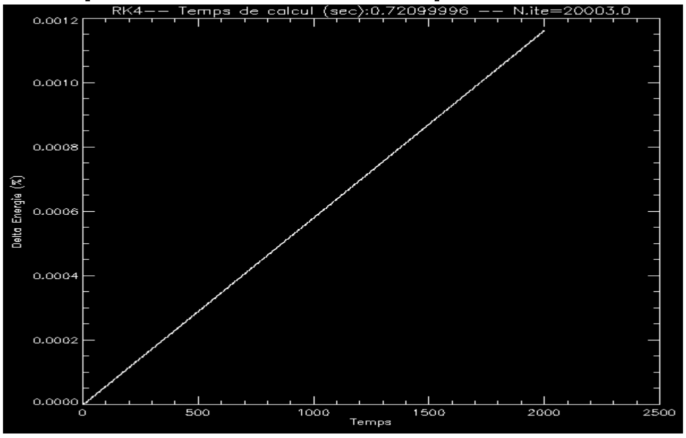
Un pas de temps typique du RK5 adaptatif

1. Evalue Y_n et Y^*_n
2. Calculer Δ
3. Calculer dt'
4. Si $dt' \neq dt$ retourner en 2 pour vérifer
5. Remplacer dt' par dt
6. Retourner en 1 (pas de temps suivant)

ATTENTION : Il est en général dangereux de travailler avec des pas de temps adaptatifs, car 1 pas de temps mal calculé peut fausser tout le calcul. Il faut bien contrôler les instabilités.

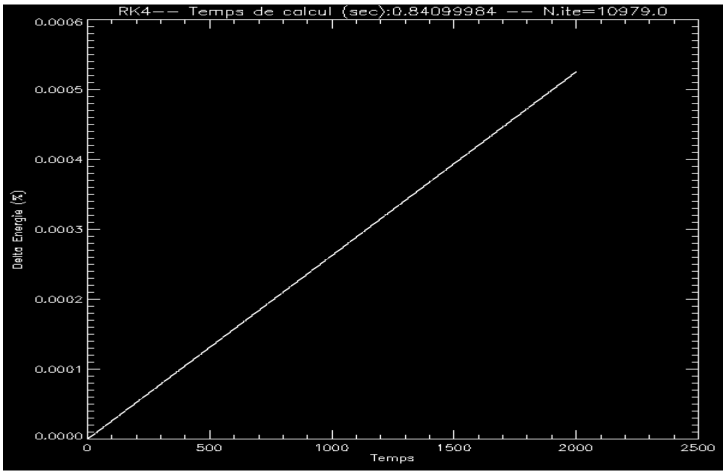
MAIS si ça marche, cela vaut la peine de l'utiliser

Exemple : RK4 Vs. RK5 Adaptatif. Problème de la planète



Energie RK4

Dt=0.2

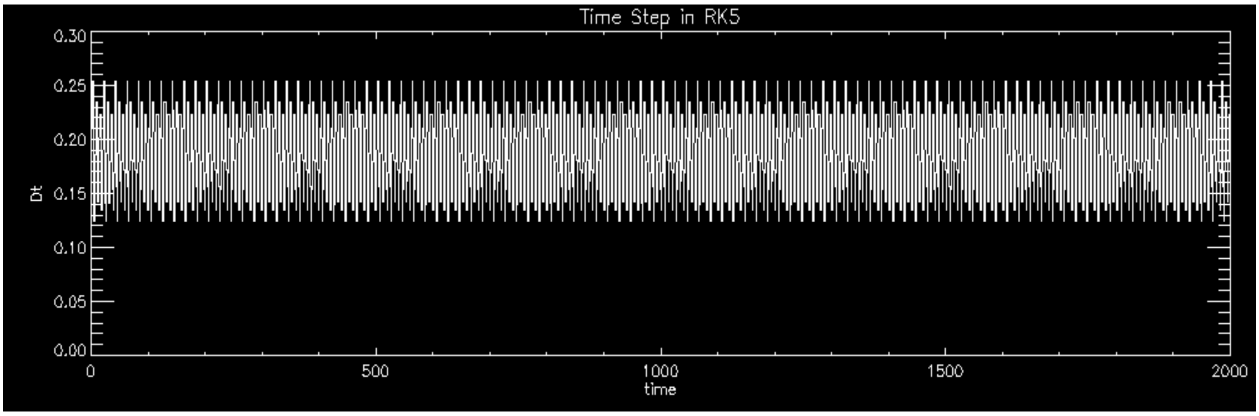


Energie RK5 Adaptatif

L’erreur sur E croit 2 fois moins vite

Temps de calcul comparable

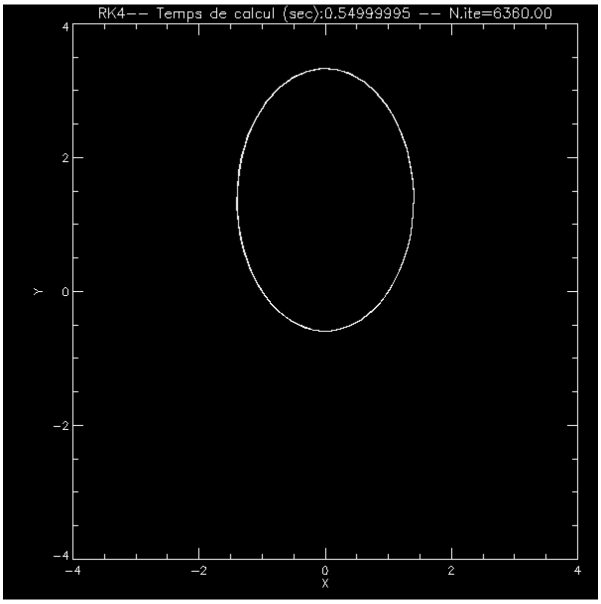
Voici le pas de temps du RK5 Adaptatif



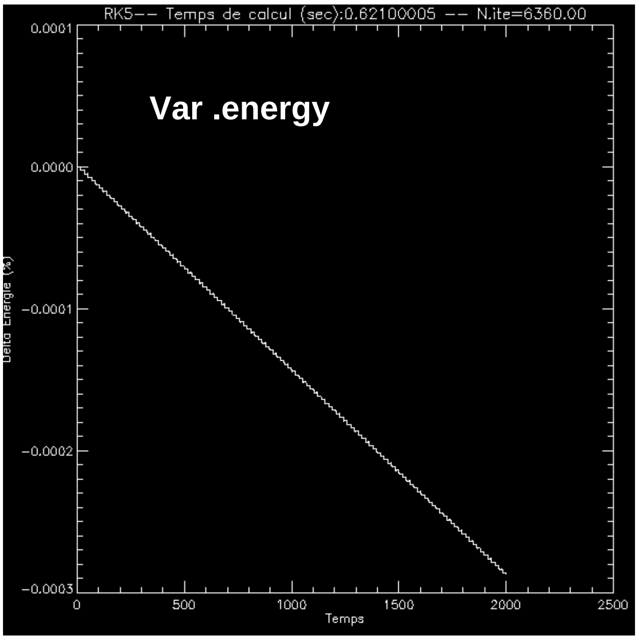
Dt dimininue quand la planète accélère (périhélie)

Dt augmente quand la planète décélère (Aphélie)

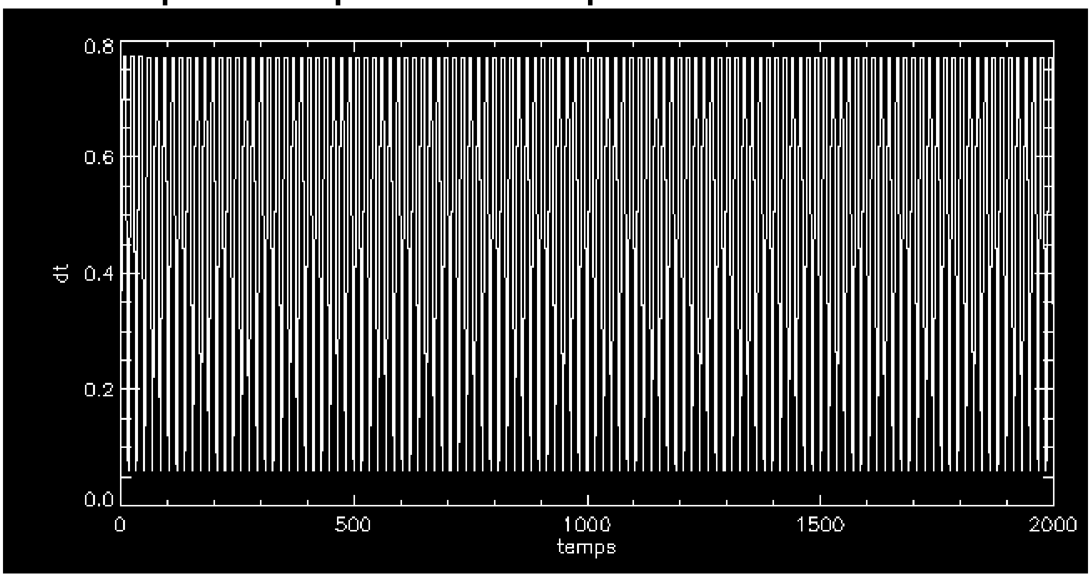
Prenons une orbite TRES allongée (difficile à intégrer)
Avec un gd pas de temps initial:



RK5 adaptatif

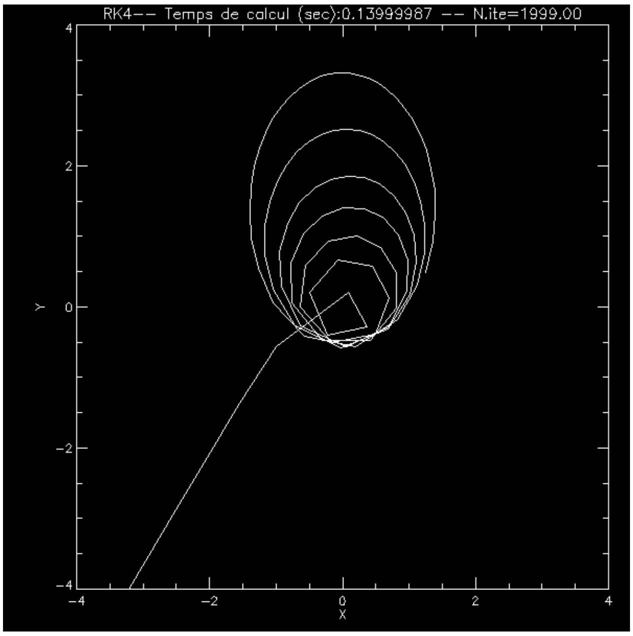


Evolution du pas de temps dur RK5 adaptatif:

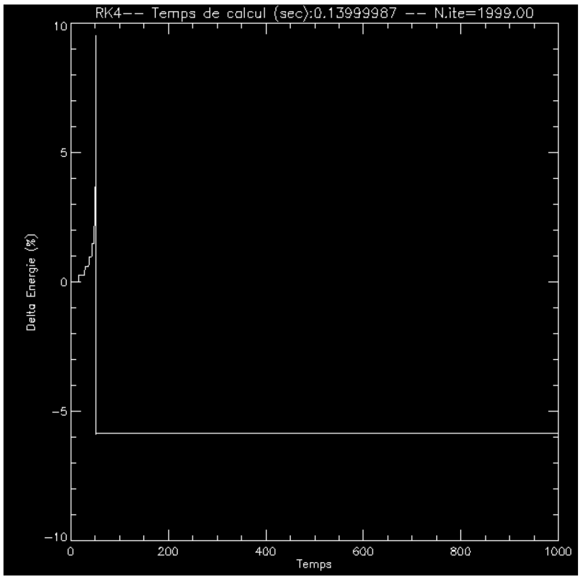


Le pas de temps s'adapte à l'orbite en permanence.
Pas de temps initial : 0.5

RK4 avec DT=0.5, mêmes conditions initiales



Energie ….



Humm…..

EN GUISE DE CONCLUSION

1. **Bien choisir son intégrateur en fonction du problème (Problème simple ? Problème Raide ? etc…)**

2. **Un ordre élevé ne signifie pas TOUJOURS une précision élevée**

3. **Parfois un intégrateur Implicite peut vous simplifier la vie et augmenter la précision**

Ne jamais croire le résultat d'un intégrateur trop vite !!!

4. **Toujours contrôler ce que l'on fait :comparer aux solutions analytiques, Contrôler l'énergie, contrôler le résultat en fonction du pas de temps**

5. **Utiliser un pas de temps adaptatif avec *beaucoup* de précautions**