



# TP d'équations différentielles ordinaires

## 1 Introduction

Il est rappelé que les programmes doivent respecter les **bonnes pratiques de la programmation** et les interfaces définies dans le notebook.

On trouvera à l'adresse suivante

<https://gitlab.irit.fr/toc/mathn7/calcul-diff-and-edo-modia/etudiants/>  
les fichiers contenant le présent document ainsi que 3 notebooks à modifier.

La date limite pour le rendu est le dimanche ??? 24h00.

## 2 Méthodes de Runge-Kutta

L'objectif de cette partie est de programmer les méthodes de Runge-Kutta explicites et implicites et de visualiser la notion d'ordre de ces méthodes. On réalisera pour ce dernier point les graphiques de la figure 1 (seulement pour la première coordonnée). Ces graphiques seront complétés avec les résultats obtenus pour le schéma implicite de Gauß à 2 étages (cf. cours sur les schémas implicites).

### 2.1 Rappels

#### 2.1.1 Schémas de Runge-Kutta

On rappelle les schémas classiques

$\begin{array}{c c} 0 & \\ \hline & 1 \end{array}$ <p>Euler</p>	$\begin{array}{c cc} 0 & & \\ \hline 1/2 & 1/2 & \\ \hline & 0 & 1 \end{array}$ <p>Runge</p>	$\begin{array}{c ccc} 0 & & & \\ \hline 1/3 & 1/3 & & \\ 2/3 & 0 & 2/3 & \\ \hline & 1/4 & 0 & 3/4 \end{array}$ <p>Heun</p>
$\begin{array}{c ccc} 0 & & & \\ \hline 1/2 & 1/2 & & \\ 1/2 & 0 & 1/2 & \\ 1 & 0 & 0 & 1 \end{array}$ <p>La méthode RK4</p>	$\begin{array}{c cc} 1/2 - \sqrt{3}/6 & 1/4 & 1/4 - \sqrt{3}/6 \\ \hline 1/2 + \sqrt{3}/6 & 1/4 + \sqrt{3}/6 & 1/4 \\ \hline & 1/2 & 1/2 \end{array}$ <p>Gauss d'ordre 4</p>	

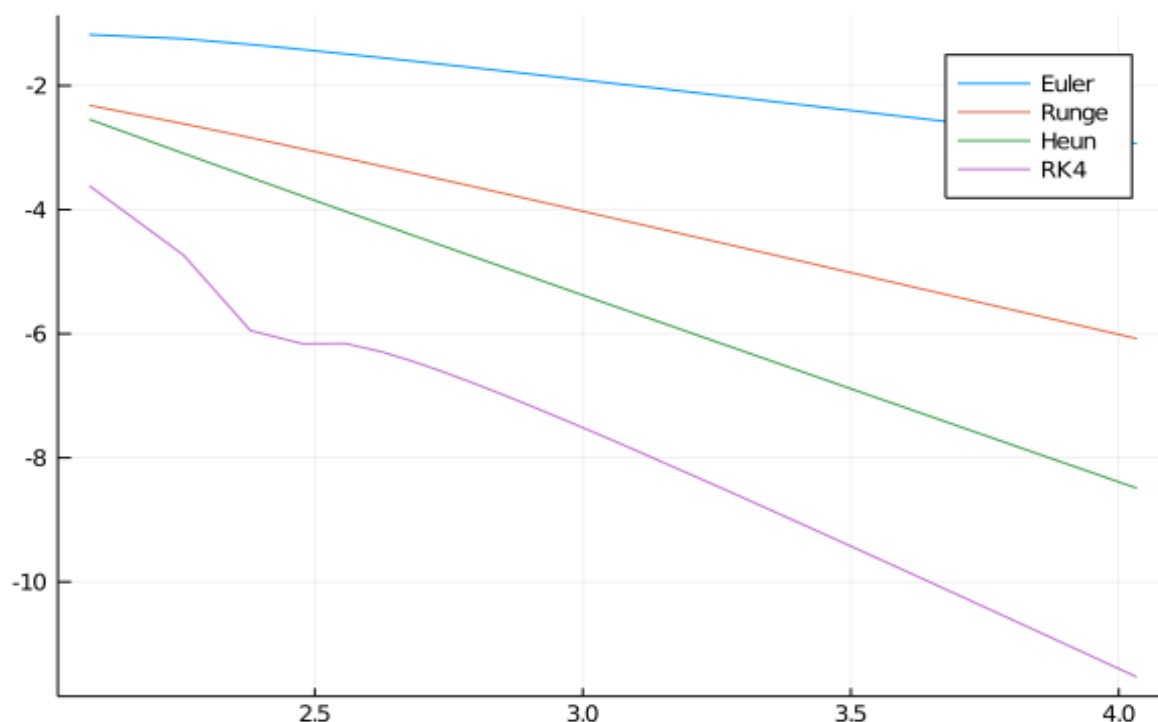


FIGURE 1 – *Erreur globale en fonction du nombre d'évaluations*, E. Hairer, S.P. Nørsett and G. Wanner, Tome I, page 140,  $\log_{10}(\text{err}) = C_1 - p \log_{10}(nfe)$ .

## 2.2 Travail demandé

### 2.2.1 Introduction

L'équation différentielle considérée est l'équation de Van der Pol

$$(IVP) \begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = (1 - x_1^2(t))x_2(t) - x_1(t) \\ x_1(0) = 2.00861986087484313650940188 \\ x_2(0) = 0 \end{cases}$$

$$t_f = T = 6.6632868593231301896996820305$$

La solution de ce problème de Cauchy est périodique de période  $T$ .

Les programmes seront effectués en JULIA. On demande de respecter les signatures des fonctions qui sont donnés dans le notebook.

### 2.2.2 Les méthodes de Runge-Kutta explicites

Les programmes d'intégration numérique par les méthodes explicites ne devront comporter qu'une seule boucle.

On demande pour cette équation :

- de réaliser les graphiques de la figure 3 qui tracent les deux composantes de la solution et le plan de phase pour  $N = 25$  ;
- de réaliser les graphiques de la figure 1. On mettra en abscisse le vecteur (en notation JULIA)  
`N0=[collect(120:60:1080); collect(1200:600:10800)]`.

### 2.2.3 Runge Kutta implicites : schéma de Gauß avec le point fixe

On prendra l'algorithme du point fixe pour résoudre le système d'équations non linéaire sur chaque pas.

1. De rajouter sur les graphiques les résultats obtenus pour le schéma implicite de Gauß. On prendra et comme valeurs pour les paramètres `nb_itmax = 15` et `K_eps = 1.e-6`. Pour la figure d'ordre on prendra comme nombre de pas le vecteur `Int([120:60:1080 1200:600:10800]/4)`.
2. On fera une deuxième figure avec les résultats correspondant au schéma implicite de Gauß pour
  - (a) `nb_itmax = 15` et `K_eps = 1.e-12`;
  - (b) `nb_itmax = 2` et `K_eps = 1.e-12`;
  - (c) `nb_itmax = 15` et `K_eps = 1.e-6`.

### 2.2.4 Runge Kutta implicites : schéma de Gauß avec Newton

On prendra l'algorithme de Newton pour résoudre le système d'équations non linéaire sur chaque pas. On demande de réaliser les mêmes graphiques que pour le cas du schéma de Gauß avec l'algorithme du point fixe.

## 3 Problèmes raides

cf. . le notebook `ode-raide-etudiant.ipynb`

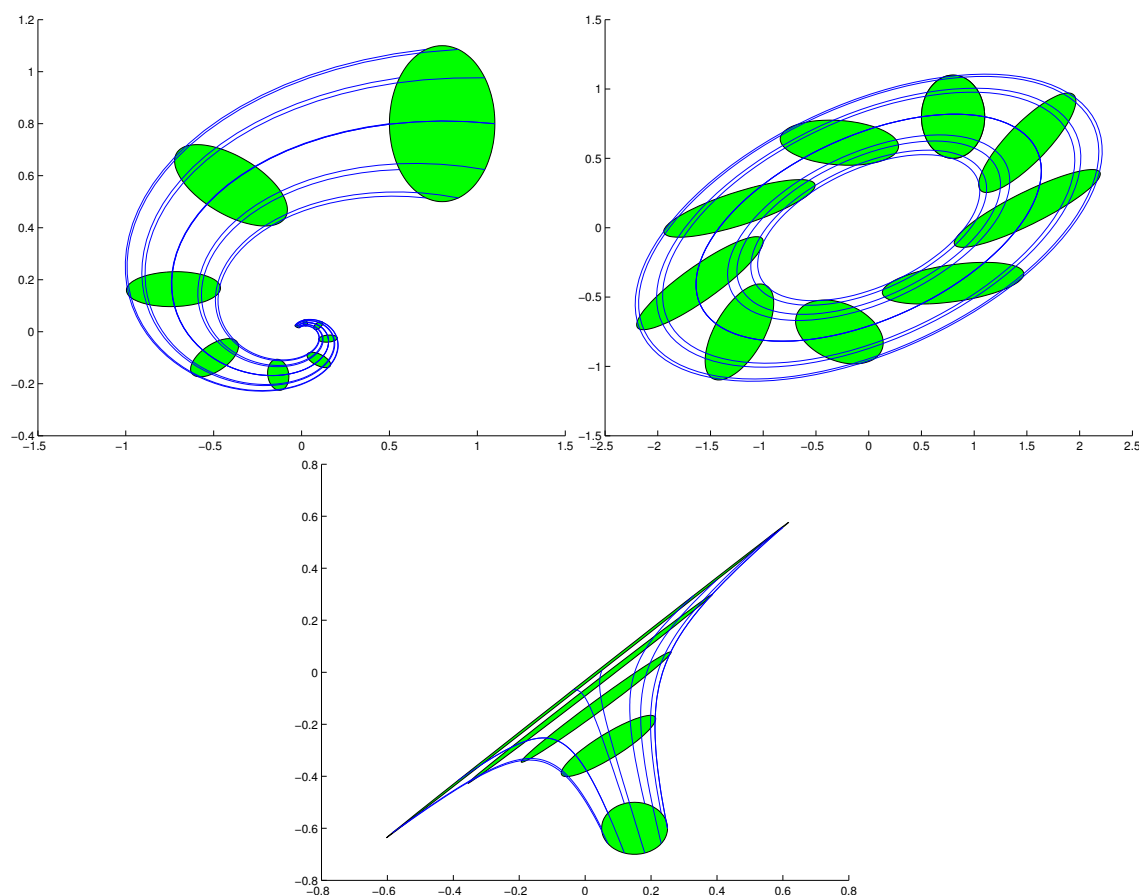
## 4 Stabilité

L'objectif de ce projet est de réaliser les graphiques de la figure 2 qui illustrent les propriétés de stabilité des edo linéaires homogènes et autonomes

$$(IVP) \begin{cases} \dot{y}(t) = Ay(t) \\ y(0) = y_0 \end{cases}$$

avec

$$A = A_1 = \begin{pmatrix} -1 & -4 \\ 1 & -1 \end{pmatrix} \quad A = A_2 = \begin{pmatrix} 1 & -4 \\ 1 & -1 \end{pmatrix} \quad A = A_3 = \begin{pmatrix} 1 & 1 \\ 4 & -2 \end{pmatrix}.$$

FIGURE 2 – Visualisation des flot pour  $A = A_1, A_2$  et  $A_3$ .

## 5 Calcul de dérivée

cf. le notebook `calcul_derivees.ipynb`

## 6 Données utiles

- On prendra 10 points sur le cercle de départ pour tracer les trajectoires (en bleu sur les graphique) et 100 points sur les cercles pour tracer le flot, c'est-à-dire les autres cercles).
- On utilisera la fonction `exp` de JULIA qui calcule l'exponentielle d'une matrice pour calculer les solutions de l'edo (*IVP*).
- Le temps final est
  - pour  $A = A_1$  et  $A = A_2$ ,  $t_f = 2 * \pi / \sqrt{3}$ ;
  - pour  $A = A_3$ ,  $t_f = 1$ .
- Le cercle des points de départ est
  - pour  $A = A_1$  et  $A = A_2$  le cercle de centre  $(0.8, 0.8)$  et de rayon  $0.3$ ;
  - pour  $A = A_3$  le cercle de centre  $(0.15, -0.6)$  et de rayon  $0.1$ .

## 7 Rendu

On mettra dans un même répertoire qui s'appellera de votre nom :

- le fichier notebook `ode-rk-etudiant.ipynb`;
- le fichier notebook `ode-raide-etudiant.ipynb`;
- un fichier notebook `stabilite.ipynb` que vous créerez pour la partie stabilité;
- le fichier notebook `calcul_derivees.ipynb`
- le fichier contenant l'archive (`<noms>.tar`), me sera envoyé par mail à l'adresse : `gergaud@enseeiht.fr`

## 8 Résultats pour tests

Voici ci-après les résultats pour  $N = 10$  pour les différents schémas. La première colonne est  $T$  et les deux suivantes  $X$ .

Euler

0.0000000000000000	2.008619860874843	0.0000000000000000
0.666328685932313	2.008619860874843	-1.338401032434280
1.332657371864626	1.116804859682459	0.029458487293659
1.998986057796939	1.136433894810396	-0.719553966980973
2.665314743729252	0.656974445534582	-1.337038672872153
3.331643429661565	-0.233932776401004	-2.281177517157184
3.997972115593878	-1.753946793786687	-3.562133341262534
4.664300801526191	-4.127498422185830	2.534846643721279
5.330629487458504	-2.438457389035097	-21.800755581293878
5.996958173390817	-16.964926207850183	51.673054627620850
6.663286859323130	17.466312380281039	-9812.202238480942469

Runge

0.0000000000000000	2.008619860874843	0.0000000000000000
0.666328685932313	1.562712360278651	0.014729243646829
1.332657371864626	1.220894137875029	-0.531644111154892
1.998986057796939	0.653509832528317	-1.176382554058143
2.665314743729252	-0.425046060567530	-2.355437605902746
3.331643429661565	-2.328613474497056	-0.666821025659383
3.997972115593878	-1.601326538420963	-2.971936242570924
4.664300801526191	-2.194094633102912	2.142928059459987
5.330629487458504	-2.093551719985328	3.009462583470000
5.996958173390817	-1.883624518031323	3.696462462835581
6.663286859323130	-1.093338066861612	4.585206668061812

Heun

0.0000000000000000	2.008619860874843	0.0000000000000000
0.666328685932313	1.863255533197435	-0.908266060321282
1.332657371864626	1.211064582800380	-0.970739550986403
1.998986057796939	0.340951280354486	-1.805621923056305
2.665314743729252	-1.343388532295027	-2.597289061805764

3.331643429661565	-1.553155419631253	-1.762194037001645
3.997972115593878	-1.761923667266568	-0.354168010449823
4.664300801526191	-1.696763283602980	0.676119846604791
5.330629487458504	-1.141093089779297	1.054354969329856
5.996958173390817	-0.188465573737059	1.992096663893664
6.663286859323130	1.593565087325988	2.154895921996044

RK4

0.0000000000000000	2.008619860874843	0.0000000000000000
0.666328685932313	1.728289565064685	-0.434733098442855
1.332657371864626	1.281792525355265	-0.876907363239775
1.998986057796939	0.486631604430130	-1.622466234766066
2.665314743729252	-1.024410786018538	-2.556512804455775
3.331643429661565	-1.953005683961636	-0.209070573804531
3.997972115593878	-1.742817815413673	0.341277278614284
4.664300801526191	-1.337915627756918	0.814186086166159
5.330629487458504	-0.600769643339283	1.494502898570029
5.996958173390817	0.821446001908530	2.589914527582617
6.663286859323130	1.886292589030273	0.455334065676449

Gauss point fixe avec N = 10, nb\_itmax = 15 et K\_eps = 1.0e-6

0.0000000000000000	2.008619860874843	0.0000000000000000
0.666328685932313	1.748077038315635	-0.614452193067304
1.332657371864626	1.227167526678123	-0.975617349879985
1.998986057796939	0.345219364241332	-1.811976134160198
2.665314743729252	-1.241597780608769	-2.475124551330471
3.331643429661565	-2.013913987826556	-0.001168903518536
3.997972115593877	-1.753872199839338	0.612809197750127
4.664300801526190	-1.235039630038149	0.970480377872889
5.330629487458503	-0.359183505874149	1.797419504046430
5.996958173390816	1.222176651788525	2.495490845811687
6.663286859323128	2.014631389666057	0.015750010678870

nfun\_Gauss\_fp = 296

ifail\_Gauss\_fp = [-1, 13, 11, 15, -1, -1, 13, 11, 15, -1]

Gauss Newton avec N = 10, nb\_itmax = 15 et K\_eps = 1.0e-6

0.0000000000000000	2.008619860874843	0.0000000000000000
0.666328685932313	1.748103353412420	-0.614524376585803
1.332657371864626	1.227167371665737	-0.975642346688062
1.998986057796939	0.345201034491682	-1.812011679746201
2.665314743729252	-1.241634876414838	-2.475102844185241
3.331643429661565	-2.013908638346081	-0.001153761617750
3.997972115593877	-1.753891509975080	0.612895411728274
4.664300801526190	-1.235024728110440	0.970518388811259
5.330629487458503	-0.359136669330347	1.797486858502427
5.996958173390816	1.222255698893968	2.495430848815410
6.663286859323128	2.014631461651370	0.015689664812240

```
nfun_Gauss_Newton = 96  
ndfun_Gauss_Newton = 76  
ifail = [4, 3, 3, 4, 5, 4, 3, 3, 4, 5]
```

On trouveras aussi ci-après les figures des solutions pour  $N = 25$  pas.

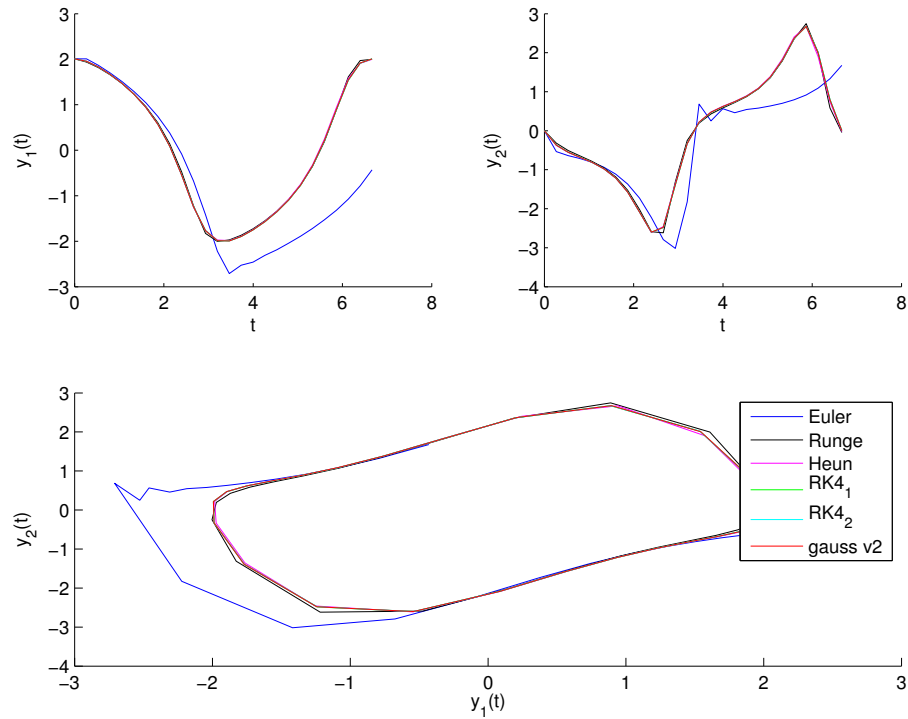


FIGURE 3 – Solution de l'équation de Van der Pol, composante 1 et 2 et plan de phase, pour les schémas de Runge-Kutta avec  $N = 25$ , pour Gauss  $nb\_itmax = 15$  et  $f\_eps = 1.e-12$ .