M 3 : Applications EDO avec *Mathematica*

Exercices du Préambule

Le fichier doit être renommé exactement comme ceci :
M1VotrenomPrenom.nb
Cela permet à vos enseignants de s'y retrouver et de ne pas avoir à renommer tous les fichiers. Merci.
Votrenom avec une seule majuscule
Prenom avec une seule majuscule
Pas de blanc, ni souligné, ni accent, ni symbole particulier
Ce fichier doit être déposé sur MOODLE à la fin de la séance, et/ou avant la fin de la semaine de la séance.

I. Objectif général et Présentation

II. Calcul matriciel

Tous les calculs sont effectuées avec des matrices réelles carrées d'ordre 2 sans aucune perte de généralité, car toutes les opérations ci-dessous sont vérifiées pour n'importe quel ordre, et n'importe quelles matrices.

■ → B. Opérations de base sur des matrices *

Pour saisir une matrice, le plus simple est d'utiliser la suite de menus :
`Insert ▹ Create Table/Matrix/ ▹ New`
ou encore les raccourcis clavier indiqués dans ce dernier menu.

On peut aussi utiliser la palette : `Palettes ▹ Basic Math Assistant`

■ → 1) Saisir les matrices suivantes :

$$\texttt{matM} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \text{ et } \texttt{matN} = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}$$

```
matM =

matN =
```

■ → 2) Calculez la somme matricielle : matM + 2 matN

matM + 2 matN

Utilisez **MatrixForm[]** pour retrouver le bel affichage.

Comparer les deux calculs suivants, et observer lesquels fonctionnent ou non :

(matM // MatrixForm) + (2 matN // MatrixForm)

matM + 2 matN // MatrixForm

Comment peut on faire aussi ?

■ → 3) Calculer le produit matriciel : matM.matN

Dot[matM, matN]==matM.matN
À quoi correspondent ces opérations ?

- **→ 4) Calculer avec *Mathematica* : matM^2 et matM.matM**

  Calculons avec *Mathematica* : **matM^2**

  Calculons avec *Mathematica* : **matM.matM**

  Expliquez à quoi correspondent ces deux calculs.

- **→ 5) Conclusions**

III. Programmation graphique

Relire M2_4M062UtilisationMMaTxt.nb, section VI.B Programmation graphique, que nous allons mettre en pratique ci-dessous.

- **→ A. Opérations graphiques et rotations ... ***

- **→ 1) Tracer le cercle unité (Circle[])**

  Tracer le cercle unité (`Circle[]`), afficher-le en rouge avec des axes, et avec une taille ImageSize→200

  Rappel : construction du type :
  Show[
  Graphics[
  { vos directives (couleur) et primitives graphiques **Line[], Arrow[], ...** }
  ],
  Axes->True, ImageSize→200
  ]

- **→ 2) Tracer le vecteur unité sur l'axe des *x* (réels), et superposer**

  (optionnel : rajout de texte $\overrightarrow{e_1}$ à l'extrémité du vecteur, avec la fonction **Text[]**)

- **→ 3) Ajouter l'affichage du produit $\begin{pmatrix} \mathrm{Cos}[\theta] & -\mathrm{Sin}[\theta] \\ \mathrm{Sin}[\theta] & \mathrm{Cos}[\theta] \end{pmatrix} . \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ en bleu (prendre une rotation d'angle 60°)**

  (**Flatten[]** pourrait vous être utile ...)

- **→ 4) En faire un Manipulate[], cadrer, jouer avec, et conclure.**

  NB : On peut s'amuser à ouvrir la réglette sous le curseur, et à faire tourner automatiquement le vecteur bleu comme une pendule, et aussi jouer avec la stroboscopie...

  Commentaires à faire :

IV. Champ de Vecteurs

▪ B. VectorPlot de l'identité

▪ → 1. Tracé brut du champ de vecteurs de l'identité

Tracer le champ de vecteurs (**`VectorPlot[]`**) de l'identité {x,y}, afficher-le entre –1 et 1 pour chaque dimension avec une taille ImageSize→150

```
VectorPlot[

]
```

▪ → 2. Tracé normalisé du champ de vecteurs de l'identité

Même tracé avec l'option **`VectorScale`** pour avoir de petites flèches de même longueur

```
VectorPlot[

]
```

▪ → 3. Idem + 1 ligne de flot

Même tracé avec l'option **`StreamPoints`** pour avoir de flot à partir du point {0.3,0.2}

```
VectorPlot[

]
```

▪ → C. VectorPlot d'applications linéaires dans un Manipulate

▪ → 1. VectorPlot d'une application linéaire simple dans un Manipulate

Dans le code ci-dessous, définir les coefficients a, b, c, et d à partir du tableau de matrices données ci dessous pour étudier le champ de vecteurs d'une application linéaire de votre choix :

Fonction	Identité	Symétrie				
		Centrale	/X	/Y	/Y = X	Orthogonale/axe[θ]
Matrice	$\begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}$	$\begin{pmatrix} -\lambda & 0 \\ 0 & -\lambda \end{pmatrix}$	$\begin{pmatrix} \lambda & 0 \\ 0 & -\lambda \end{pmatrix}$	$\begin{pmatrix} -\lambda & 0 \\ 0 & \lambda \end{pmatrix}$	$\begin{pmatrix} 0 & \lambda \\ \lambda & 0 \end{pmatrix}$	$\begin{pmatrix} \lambda \,\mathrm{Cos}[\theta] & \lambda \,\mathrm{Sin}[\theta] \\ \lambda \,\mathrm{Sin}[\theta] & -\lambda \,\mathrm{Cos}[\theta] \end{pmatrix}$
Fonction	Cisaillement[λ]	Rotation[θ]				Projection[θ]
Matrice	$\begin{pmatrix} 1 & \lambda \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} \lambda \,\mathrm{Cos}[\theta] & -\lambda \,\mathrm{Sin}[\theta] \\ \lambda \,\mathrm{Sin}[\theta] & \lambda \,\mathrm{Cos}[\theta] \end{pmatrix}$				$\begin{pmatrix} \lambda \,\mathrm{Cos}[\theta]^2 & \lambda \,\mathrm{Sin}[\theta]\,\mathrm{Cos}[\theta] \\ \lambda \,\mathrm{Sin}[\theta]\,\mathrm{Cos}[\theta] & \lambda \,\mathrm{Sin}[\theta]^2 \end{pmatrix}$

Le code utilise les 2 options de **VectorPlot[]**, **`VectorScale`** et **`StreamPoints`** vues précédemment.
Il comporte d'autres éléments de code intéressants : :

l'objet original **Locator** pour déplacer interactivement un objet dans un graphique,
des aspects d'affichage un peu techniques comme **Control[]**, ou **Style[]**, faciles à comprendre.

```
Manipulate[
   (* I. -------Definitions et calcul des variables ----------------- *)
  With[{a = ? , b = ? ,c=?,d=?},
        With[{mat = (a b
                     c d)},
          (* II. -------Affichages ----------------------------------- *)
          VectorPlot[
           {a * x + b * y, c * x + d * y}, {x, -1, 1}, {y, -1, 1},
           StreamPoints → {{p[[1]], p[[2]]}},
           StreamStyle → {Red, Thick}, VectorScale → {Tiny, Automatic, None},
           ImageSize → {200, 200}, FrameLabel → {x, y}
          ]
        ]],
 (* III. ------Contrôles des variables du Manipulate------------------ *)
 Style["Matrice de Symétrie Orthogonale[λ, θ] :", Bold],
 Control[{{λ, 1, "λ"}, -5, 5, .01, Appearance → "Labeled"}],
 Control[{{θ, 0, "θ"}, -π, π, .01, Appearance → "Labeled"}],
 {{p, {.5, .5}}, {-1, -1}, {1, 1}, Locator}
 ]
```

- **→ 2. Texte et VectorPlot dans deux colonnes adjacentes**

Nous reprenons le code précédent en rajoutant au moyen de **`Eigensystem[]`** le calcul des valeurs propres, et des vecteurs propres de la matrice ainsi que de son déterminant au moyen de **`Det[]`**.

Les changements de code servent principalement à présenter le texte et le VectorPlot dans deux colonnes au moyen :

des fonctions **`Grid`** et **`Column`** d'organisation graphique en une colonne de texte et le graphique, et de la fonction **`TraditionalForm`** pour un bel affichage de texte.

```
Manipulate[
   (* I. -------Definitions et calcul des variables ----------------- *)
   With[{a = ? , b = ? ,c=?,d=?},
         With[{dvec = (x' ; y'), vec = (x ; y), mat = (a b ; c d)},
          With[(* Valeurs propres et vecteurs propres de la matrice *)
           {esys = Eigensystem[mat], det = Det[mat] },
           With[{
             λ1 = esys[[1, 1]], λ2 = esys[[1, 2]],
             vec1 = (esys[[2, 1]][[1]] ; esys[[2, 1]][[2]]), vec2 = (esys[[2, 2]][[1]] ; esys[[2, 2]][[2]])
            },
            (* II. -------Affichages --------------------------------- *)
            Grid[{{
               (* A. Texte de droite *)
               Style[Text[Column[{
                    Text[TraditionalForm[Row[{"f : ℝ^2 ⟶ ℝ^2"}]]],
                    Text[TraditionalForm[Row[{"f : (x ; y) ↦ (a b ; c d) (x ; y)"}]]],
                    "",
                    Text[TraditionalForm[Row[{dvec , " = " , mat, vec}]]],
                    Text[TraditionalForm[Row[{" déterminant = ", det}]]],
                    "",
                    Text[TraditionalForm[Row[{"λ_1 = ", NumberForm[λ1, {5, 2}]}]]],
                    Text[TraditionalForm[Row[{"λ_2 = ", NumberForm[λ2, {5, 2}]}]]],
                    "",
                    Text[TraditionalForm[
                      Row[{"v_1 = ", NumberForm[vec1, {5, 2}]}]]],
                    Text[TraditionalForm[Row[{"v_2 = ", NumberForm[
                         vec2, {5, 2}]}]]],
                    "                                                   "
                   }]], 12],
               (* B. Figure de gauche *)
               VectorPlot[
                {a * x + b * y, c * x + d * y}, {x, -1, 1}, {y, -1, 1},
                StreamPoints → {{p[[1]], p[[2]]}}, StreamStyle →
                 {Red, Thick}, VectorScale → {Tiny, Automatic, None},
                ImageSize → {250, 250}, FrameLabel → {x, y}
               ]
              }}]
          ]]]],
 (* III. ------Contrôles des variables du Manipulate---------------- *)
 Style["Paramètres de la matrice de Symétrie Orthogonale :", Bold],
 Control[{{λ, 1, "λ"}, -5, 5, .01, Appearance → "Labeled"}],
 Control[{{θ, 0, "θ"}, -π, π, .01, Appearance → "Labeled"}],
 {{p, {.5, .5}}, {-1, -1}, {1, 1}, Locator}
 ]
```

Jouer avec les curseurs et le locateur dans tout l'espace....

→ V. Expressions des Solutions d'EDO avec Mathematica

■ → A. Résolution Formelle

■ → 1. Résolution formelle d'EDO non linéaires

a./ Résoudre l'équation différentielle suivante :

$\frac{dx}{dt} = x\,(a - b\,x) - c$ où $a,\ b,\ c$ sont positives ou nulles

b./ Définir une fonction de $a,\ b,\ c,\ d,$ et t qui mémorise l'expression de la solution :

c./ Tracer la solution pour $a = 1,\ b = 0.02,\ c = 0,\ d = 0.01,\ t = 20$

d./ Tracer la solution générale avec un **Manipulate[]** pour des valeurs raisonnables de $a,\ b,\ c,\ d$ et t :

■ → 2. Résolution formelle d'EDO linéaires

a./ Soit : $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, $X = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$ et $B = \begin{pmatrix} \mathrm{b1} \\ \mathrm{b2} \end{pmatrix}$, résoudre le problème de Cauchy suivant :

$\frac{dX}{dt} = A\,X + B \quad \forall\, t \in \mathbb{R}$

$\mathrm{X0} = \begin{pmatrix} \mathrm{x0} \\ \mathrm{y0} \end{pmatrix}$ à $t = 0$

b./ Tracer la solution pour $a = -1,\ b = 2,\ c = 3,\ d = -4,\ \mathrm{x0} = 1,\ \mathrm{y0} = 0,\ t$ alllant de 0 à 2.

→ VI. Résolution pratique d'EDO numériques

■ A. Résolution du système d'EDO pour $\frac{e_0}{K_m + s_0} << 1$

a./ Résoudre le système d'équations différentielles suivant :

$\frac{ds}{dt}(t) = -\mathrm{k1}\ \mathrm{e0}\ s(t) + (\mathrm{k1}\ s(t) + \mathrm{kr1})\ c(t) \qquad \forall\, t \in \mathbb{R}$

$\frac{dc}{dt}(t) = \mathrm{k1}\ \mathrm{e0}\ s(t) - (\mathrm{k1}\ s(t) + \mathrm{kr1} + \mathrm{k2})\ c(t)$

avec $\mathtt{k1} = 10^4$ en $\mathrm{mM}^{-1}\ s^{-1}$; $\mathtt{k2} = 10^3\ \mathrm{s}^{-1}$, $\mathtt{kr1} = 10^3\ \mathrm{s}^{-1}$

et les concentrations initiales : $\mathtt{e0} = 10^{-3}$ mM, $\mathtt{s0} = 1$ mM (excès de substrat).

b./ Tracer les solutions :

Exercices EDO

→ I. Elaboration d'un Manipulate pour EDO

▪ → A. VectorPlot de 9 applications linéaires fonctions de λ et/ou θ

Champ de vecteurs 9 applications linéaires fondamentales d'angle θ, et/ou d'homothétie H[λ] de rapport λ

▪ → 1. Manipulate pour afficher une matrice parmi 9

En utilisant les définitions de matrices ci-dessous, écrire un Manipulate pour afficher l'une de ces matrices au choix ainsi que le premier coefficient de la matrice :

```
(* -------Liste de choix de 9 matrices 2D pour Manipulate --------- *)
(* Identite H[λ] *)matIdλ[λ_, θ_] := (λ 0
                                       0 λ)
(* SymetrieCentrale H[λ]*) matSymCentλ[λ_, θ_] := (-λ 0
                                                    0 -λ)
(* Symetrie/X H[λ] *) matSymXλ[λ_, θ_] := (λ 0
                                           0 -λ)
(* Symetrie/Y H[λ] *) matSymYλ[λ_, θ_] := (-λ 0
                                           0 λ)
(* Symetrie/Y=X H[λ] *) matSymYeqXλ[λ_, θ_] := (0 λ
                                                λ 0)
(* SymetrieOrthogonale/axe[θ] H[λ] *) matSymλθ[λ_, θ_] := (λ Cos[θ]  λ Sin[θ]
                                                           λ Sin[θ]  -λ Cos[θ])
(* Rotation[θ] H[λ] *) matRotλθ[λ_, θ_] := (λ Cos[θ]  -λ Sin[θ]
                                            λ Sin[θ]  λ Cos[θ])
(* Projection[θ] H[λ] *) matProjλθ[λ_, θ_] := (λ Cos[θ]^2          λ Sin[θ] Cos[θ]
                                               λ Sin[θ] Cos[θ]     λ Sin[θ]^2)
(* Cisaillement[λ] *) matCisλ[λ_, θ_] := (1 λ
                                          0 1)
(* ------------------------------------------------------------- *)
Manipulate[

]
```

▪ → 2. Manipulate pour tracer le champ de vecteurs d'une matrice parmi 9

En reprenant le code ci-dessus, compléter ce Manipulate pour tracer le champ de vecteurs d'une matrice parmi 9 :

```
Manipulate[
 VectorPlot[



 ],
 Style["Choix de Matrices particulières ", Bold],
 {{mat, matIdλ}, {matIdλ, matSymCentλ, matSymXλ,
   matSymYλ, matSymYeqXλ, matSymλθ, matRotλθ, matProjλθ, matCisλ}},
 Control[{{λ, 1, "λ"}, -5, 5, .01, Appearance → "Labeled"}],
 Control[{{θ, 0, "θ"}, -π, π, .01, Appearance → "Labeled"}],
 (* Point P des Initialisations de la ligne du champ de vecteurs *)
 {{p, {.5, .5}}, {-1, -1}, {1, 1}, Locator}
]
```

- **→ 3. Texte et VectorPlot dans deux colonnes adjacentes dans un Manipulate de 9 matrices possibles**

Avec le code vu en préambule, on obtient le Manipulate plus complet dont vous étudierez le code et les effets.

Out[27]=

Choix de Matrices particulières fonction de λ et θ

mat matSym$\lambda\theta$

λ 1

θ $\frac{\pi}{3}$

$f : \mathbb{R}^2 \longrightarrow \mathbb{R}^2$

$$f : \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & -\frac{1}{2} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

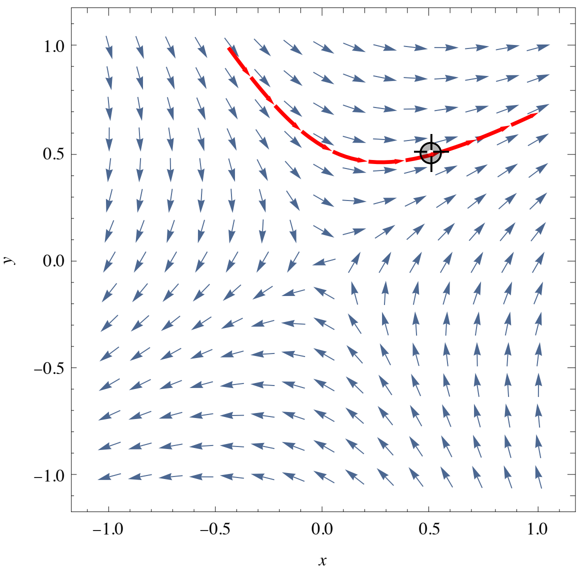
déterminant = −1

$\lambda_1 = -1.00$

$\lambda_2 = 1.00$

$$v_1 = \begin{pmatrix} -\frac{1.00}{\sqrt{3.00}} \\ 1.00 \end{pmatrix}$$

$$v_2 = \begin{pmatrix} \sqrt{3.00} \\ 1.00 \end{pmatrix}$$



a./ Pour les matrices : Identité H[λ], les symétries H[λ], en quoi change le champ de vecteurs en fonction de λ ?

b./ Avec une matrice de rotation, pour quelles valeurs d'angle θ, obtient-on :
un flot circulaire ?

en colimaçon allant vers {0, 0},

allant vers l'infini ?

c./ Y-a-t-il des axes de symétrie dans le cas des
symétries ?

rotations ?

projections ?

des cisaillements ?

Expliquez votre réponse à partir de :
l'observation du champ de vecteurs

et de tout autre élément

■ → 4. Idem + matrice générale + Axes de symétrie et vecteurs propres

On rajoute la possibilité de traiter une matrice réelle générale en activant le "generalFlag", et en contrôlant les valeurs des coefficients a, b, c, d de cette matrice.

Reprendre la question précédente **c./** avec le Manipulate ci-dessous

c./ Y-a-t-il des axes de symétrie dans le cas des
symétries ?

rotations ?

projections ?

des cisaillements ?

des matrices générales ?

Expliquez votre réponse à partir de :
l'observation du champ de vecteurs

et de tout autre élément

Choix de Matrices particulières fonction de λ et θ

mat matSym$\lambda\theta$

λ 1 θ $\frac{\pi}{3}$

generalFlag

Matrices générales fonction de a, b, c, d

a 0.5 b −0.75

c 0.1 d 0.5

Out[28]=

$f : \mathbb{R}^2 \longrightarrow \mathbb{R}^2$

$f : \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & -\frac{1}{2} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

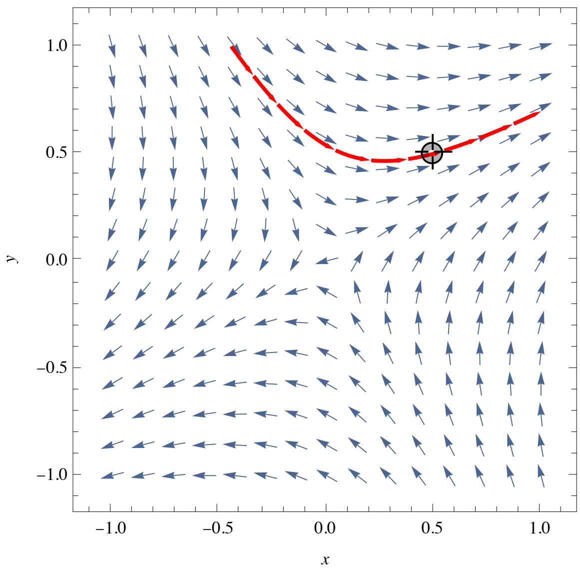
déterminant = −1

$\lambda_1 = -1.00$

$\lambda_2 = 1.00$

$$v_1 = \begin{pmatrix} -\frac{1.00}{\sqrt{3.00}} \\ 1.00 \end{pmatrix}$$

$$v_2 = \begin{pmatrix} \sqrt{3.00} \\ 1.00 \end{pmatrix}$$



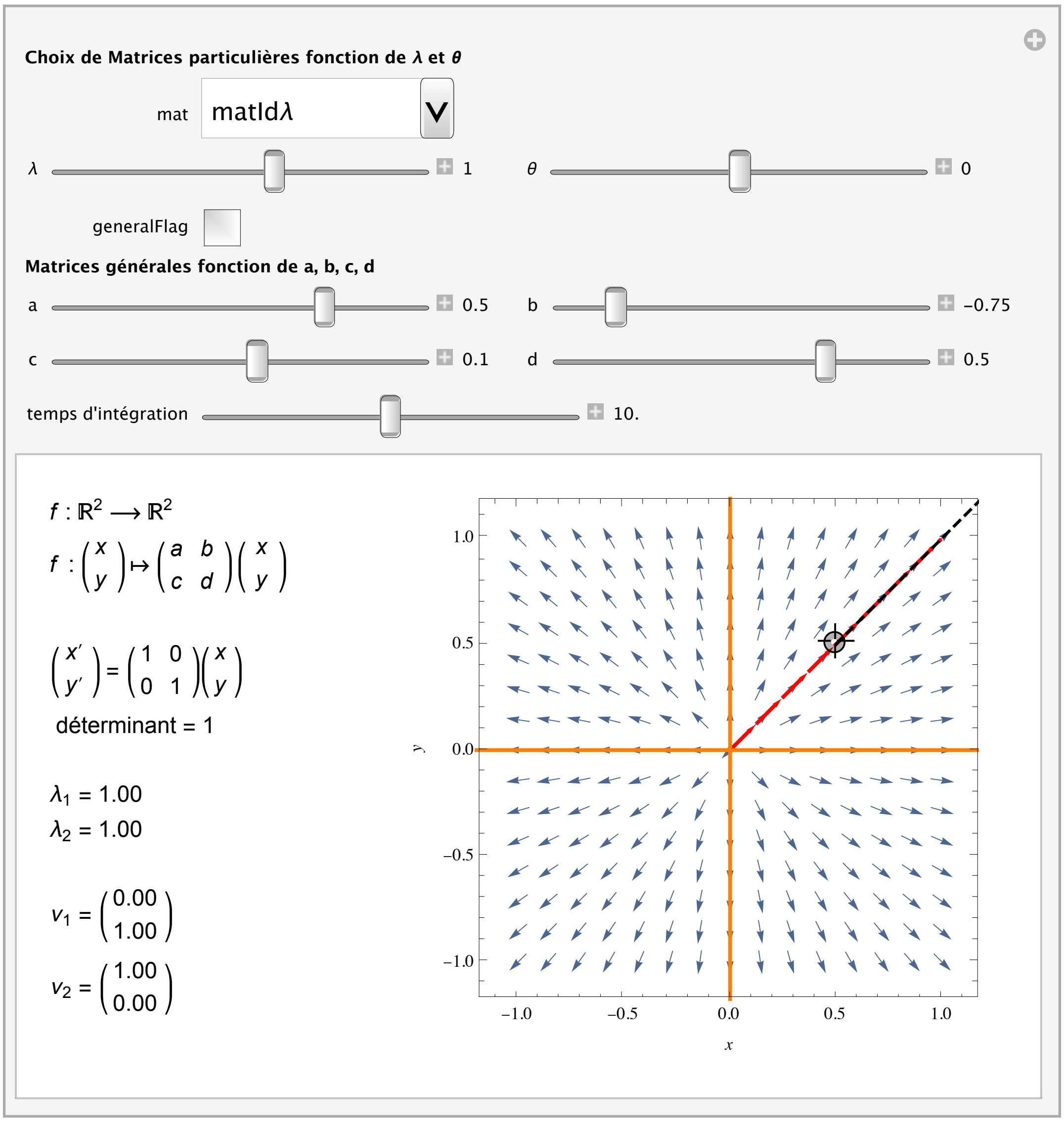
- **→ B. VectorPlot de 10 applications linéaires + solutions EDO**

- **→ 1. Idem + calcul et affichage des solutions EDO**

On rajoute :

la partie calcul des EDO à la fin de I. Definitions et calcul des variables, et
le tracé des solutions à la fin de II. Affichage,
le contrôle du temps total de calcul de l'EDO à la fin du III. Contrôles des variables du Manipulate.

La solution de l'EDO est tracée en pointillés noirs à partir du point P donné par le "Locator".



→ **a./** Quel est l'élément nouveau ?

→ **b./** Qu'observez-vous ?

▪ → C. VectorPlot de 10 (applications + EDO) linéaires + cas non linéaire

▪ 1. Redéfinition des matrices sans homothétie (À VALIDER)

L'homothétie de rapport λ complique beaucoup les calculs, alors qu'elle n'a pas d'effet dans VectorPlot qui remet à l'échelle. On introduit donc d'abord la simplification des matrices ci-dessous.
Dans le code du Manipulate, il suffit de changer une fois mat[λ,θ] en mat[θ] dans le code ci-dessous, et de modifier le contrôle des matrices avec les noms sans λ, mais avec seulement θ.

```
(* -------Liste de choix de 9 matrices 2D pour Manipulate --------- *)
(* Simplification en supprimant H[λ] l'homothétie de rapport λ *)
(* Identite *)                    matId[ϕ_] := (1 0
                                                0 1)
(* SymetrieCentrale *)            matSymCent[ϕ_] := (-1  0
                                                      0 -1)
(* Symetrie/X *)                  matSymX[ϕ_] := (1  0
                                                  0 -1)
(* Symetrie/Y *)                  matSymY[ϕ_] := (-1 0
                                                   0 1)
(* Symetrie/Y=X *)                matSymYeqX[θ_] := (0 1
                                                     1 0)
(* SymetrieOrthogonale/axe[θ] *) matSymθ[θ_] := (Cos[θ]  Sin[θ]
                                                 Sin[θ] -Cos[θ])
(* Rotation[ϕ] *)                 matRotθ[θ_] := (Cos[θ] -Sin[θ]
                                                  Sin[θ]  Cos[θ])
(* Projection[ϕ] *) matProjθ[θ_] := (      Cos[θ]^2     Sin[θ] Cos[θ]
                                     Sin[θ] Cos[θ]        Sin[θ]^2  )
(* Cisaillement[ϕ] *)             matCisθ[θ_] := (1 θ
                                                  0 1)
(* --------------------------------------------------------------- *)
```

- **→ 2. Ajout d'un cas EDO non linéaire (Roméo et Juliette)**

Puis on introduit la la possibilité de termes non linéaires dans la matrice, et dans l'EDO. Pour cela, il faut procéder aux modifications suivantes.

Dans I. Définitions et calcul des variables, il faut :

* remplacer la matrice $\begin{pmatrix} \mathbf{a} & \mathbf{b} \\ \mathbf{c} & \mathbf{d} \end{pmatrix}$ par

$$\begin{pmatrix} \mathbf{a} & \mathbf{b\ If[includeNonlinearTerms,\ JM - Abs[y]\ ,\ 1]} \\ \mathbf{c\ If[includeNonlinearTerms,\ RM - Abs[x]\ ,\ 1]} & \mathbf{d} \end{pmatrix}$$

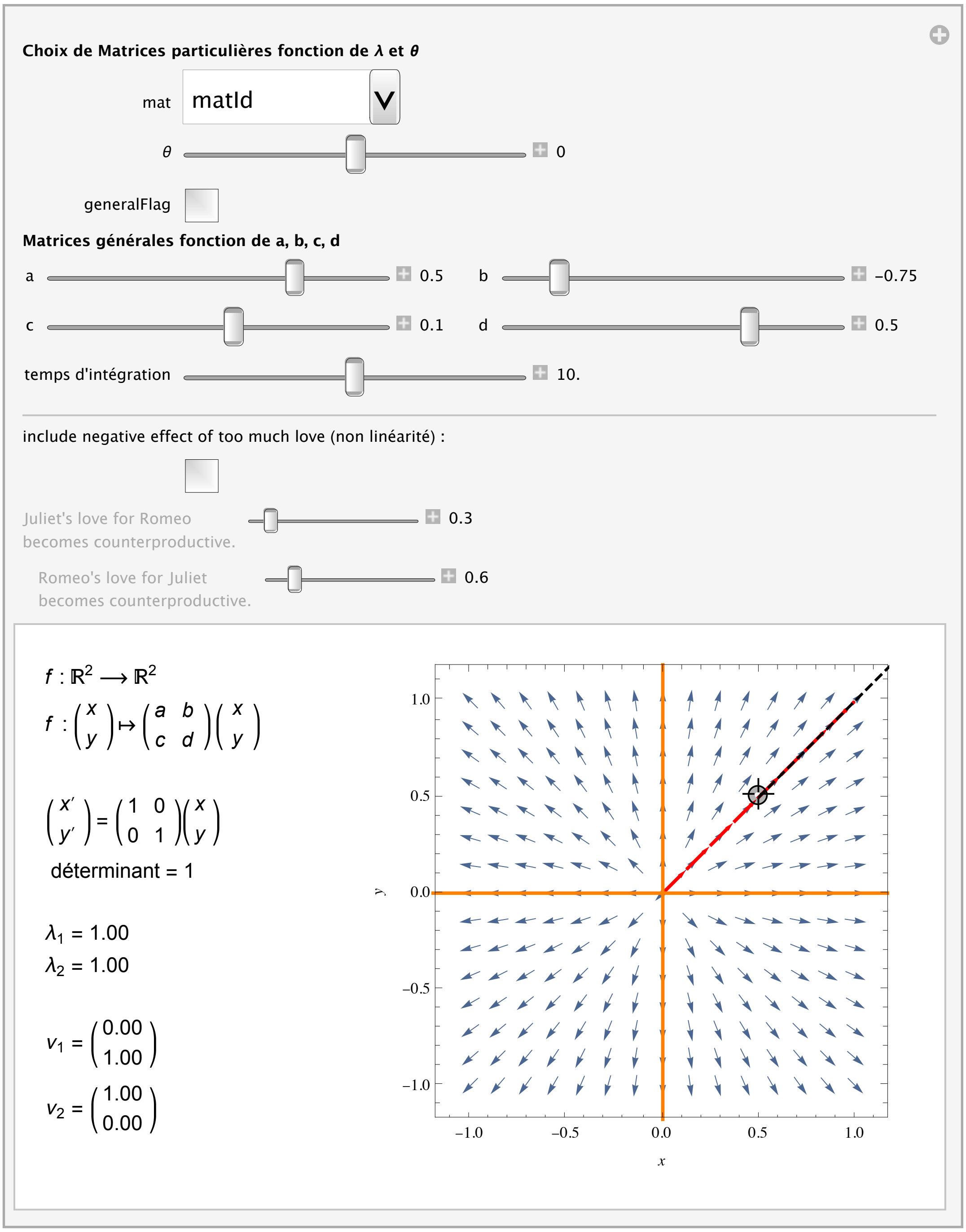
** remplacer **mat[λ,θ]** par **mat[θ]**

*** faire les substitutions **/.{Abs[y]→ Abs[J[t]]** et **/.{Abs[x]→ Abs[R[t]]** dans les EDO.

À la fin du III. Contrôles des variables du Manipulate, il suffit de :

supprimer le contrôle de λ, et

d'ajouter le contrôle des coefficients non linéaires portant sur JM et Rm de l'EDO .



→ **a./** Qu'observez-vous ?