

Les modèles **Animation.mod** et **Capture.mod** (1.3)

Pour TeXgraph 1.96

5 juillet 2021

Table des matières

1	Animation.mod	1
1.1	Présentation	1
1.2	Le menu	1
1.3	Créer une animation	2
1.4	Exemple 1	2
1.5	Exemple 2	3
2	Capture.mod	4
2.1	Présentation	4
2.2	Le menu	5
2.3	Exemple 3	5

1 Animation.mod

1.1 Présentation

Le modèle *Animation.mod* permet une création simplifiée d'animations dans TeXgraph, celles-ci pouvant être enregistrées automatiquement sous forme de fichiers pgf (frame1.pgf, frame2.pgf...) puis compilées dans un même fichier pdf avec une image par page, chaque page étant à la taille de l'image.

Ce fichier d'images peut être :

- converti en gif animé [c'est une option du menu du modèle], cela suppose que vous disposiez de la suite *ImageMagick*¹, car c'est son utilitaire **convert** qui est utilisé,
- converti en fichier flash [c'est une option du menu du modèle], cela suppose que vous disposiez des outils *Swftools*². Deux fichiers sont créés : *<fichier>.swf* et *<fichier>.html*, le deuxième contenant le code permettant d'inclure le premier dans une page html,
- utilisé pour fabriquer un pdf animé à l'aide du package *animate* d'Alexander GRAHN.

Depuis la version 1.2, l'animation peut être exportée pour le web sous forme d'une succession d'images svg animées par du code javascript et visualisable dans un navigateur. Le modèle définit une nouvelle variable dans votre graphique : *TitreAnimation*, qui est une chaîne de caractères destinée à contenir le titre de l'animation lors de l'export pour le web.

1.2 Le menu

Celui-ci est composé des boutons suivants :

- **Réinitialiser** : permet d'arrêter l'animation si elle est en cours, et de la replacer à son état initial.
- **Frame** : permet de définir le nombre de frames de l'animation, chaque frame donnera un fichier pgf à l'enregistrement, et une page dans le fichier pdf final.
- **Tempo** : réglage de la temporisation, c'est à dire du laps de temps entre 2 images, celui-ci est compté en milli-secondes. La valeur par défaut est de 100.
- **Mode** : définition du mode d'animation.
 - Mode=0 : c'est le mode normal, l'animation se déroule automatiquement de la première à la dernière frame.
 - Mode=1 : c'est le mode frame par frame, il faut cliquer le bouton *Play/Stop* pour passer à la frame suivante.
 - Mode=2 : c'est le mode en boucle.L'enregistrement se fait dans le mode normal.

1. <http://imagemagick.sourceforge.net/>

2. <http://www.swftools.org/>

- **Play/Stop** : permet de lancer ou stopper l'animation. Pour remettre celle-ci au point de départ alors qu'elle n'est pas terminée, il faut utiliser le bouton *Réinitialiser*.
- **Exporter pour le web** : permet de lancer l'animation en mode normal, lorsque vous avez donné un titre à votre animation, chaque frame est enregistrée dans le dossier TeXgraphMac/Animation/animJs/animation/img sous forme d'un fichier svg : image001.svg, image002.pgf ...
Le fichier TeXgraphMac/Animation/animJs/animation/index.html devrait alors être lancé dans votre navigateur et permettre la visualisation de l'animation, à condition que celui-ci autorise javascript.
- **Exporter en pdf** : permet de lancer l'animation en mode normal (sans affichage écran), chaque frame est enregistrée dans le dossier temporaire sous forme d'un fichier pgf : frame1.pgf, frame2.pgf ... Lorsque l'animation est terminée vous devez donner un nom de fichier (avec chemin, mais sans extension), les fichiers « frames » seront alors compilés en un fichier pdf qui portera ce nom, avec une image par page, et chaque page étant à la taille de l'image.
Mise en garde : si vous avez un grand nombre de frames et une temporisation rapide, il est possible que votre système ait du mal à suivre et que toutes les frames ne soient pas encore enregistrées sur le disque au moment où vous allez lancer la visualisation pour le web ou la compilation en pdf, il ne sert donc à rien de se précipiter lors de cette phase !
- **Pdf vers gif animé** : permet de convertir le fichier pdf créé à l'enregistrement, en gif animé. Cette option utilise le programme *convert*. Le fichier gif porte le même nom que le fichier pdf.
- **Pdf vers flash** : permet de convertir le fichier pdf créé à l'enregistrement, en fichier flash (*.swf). Cette option utilise les outils *Swftools*. Un fichier html est également créé, il contient le code permettant d'inclure le fichier flash dans une page html. Ces deux fichiers portent le même nom que le fichier pdf. Il faut noter que la technologie flash est aujourd'hui en passe de devenir obsolète pour l'internet, et ne devrait plus être utilisée. Il existe cependant les lecteurs de fichiers flash qui fonctionnent en local.
- **Aide** : permet d'ouvrir le présent document dans le lecteur pdf.

1.3 Créer une animation

On commence par importer le fichier modèle (F3) *Animation.mod* (le fichier de macros *Animation.mac* est automatiquement chargé par le modèle).

- On compose le graphique qui correspond à la première frame. Il y aura en général des éléments fixes et des éléments mobiles, ces derniers vont dépendre de **paramètres** qui vont changer au cours de l'animation, on déclare ces paramètres en **variables globales** avec leur valeur d'origine (valeur par défaut).

Puis il faut compléter certaines macros qui sont déjà présentes dans la liste des macros :

- La macro **Initialiser()** : dans celle-ci on remet les paramètres à leur valeur par défaut et on peut terminer par la commande *ReCalc()* qui aura pour effet de revenir à la première frame.

- La macro **OnBeginAnim()** : si vous avez des éléments graphiques à créer juste le temps de l'animation, c'est ici qu'il faut le faire. Cette macro est exécutée juste avant le lancement de l'animation.

- La macro **OnEndAnim()** : si vous avez des éléments graphiques à détruire juste après l'animation, c'est ici qu'il faut le faire. Cette macro est exécutée juste après l'animation.

- La macro **MakeFrame()** : cette macro à un paramètre (%1 qui représente le numéro de la frame) est le cœur de l'animation, dans celle-ci vous devez donner à vos différents paramètres la valeur correspondant à la frame n° %1. On termine en général cette macro par un *ReCalc()* (ou un *ReCalc(image modifiée)*).

- Avec le bouton *Frames*, définissez le nombre de frames de votre animation.

1.4 Exemple 1

Nous allons représenter un point *M* en mouvement circulaire autour de l'origine avec un rayon de 4 :

1. Charger le modèle *Animation.mod* (Shift+F2).
2. Créer un élément graphique Utilisateur (Ctrl+U), appelé **fond**, avec la commande :

```
[
  view(-4.5,4.5,-4.5,4.5), Marges(0,0,0,0), //réglages fenêtre
  Axes(0,1+i) //tracé des axes
]
```

3. Nous prendrons comme paramètre l'angle $(\vec{i}, \overrightarrow{OM})$, appelons le **alpha**. On crée alors une variable globale (F7), on la nomme **alpha**, avec la valeur 0.

4. Créer un élément graphique Utilisateur (Ctrl+U), appelé **trajectoire**, avec la commande :

```
[
  Width:=8, Color:=red,
  Arc(4,0,4*exp(i*alpha),4), //arc de cercle
  DotStyle:=bigdot, Color:=black,
  LabelDot( 4*exp(i*alpha), "M", "N", 1) //visualisation du point
]
```

5. Dans la macro **Initialiser()**, ajouter les instructions (entre les crochets) :

alpha:=0, ReCalc(trajectoire)

6. Dans la macro **MakeFrame()**, ajouter les instructions (entre les crochets) :

alpha:= %1*pi/18, ReCalc(trajectoire)

cela signifie que l'on tourne de $\pi/18$ (10 degrés) à chaque frame, cela nous donne donc 36 frames.

7. Avec le bouton *Frames* définir le nombre de frames à 36. Reste à essayer!

Nous allons enregistrer l'animation

8. Cliquer le bouton *Exporter en pdf* et valider. À la fin de l'export une fenêtre s'ouvre demandant un nom de fichier, donner un nom (avec chemin) mais sans extension, et valider. Nous supposons que ce nom est *Exemple1*.

Ce fichier peut-être utilisé pour faire une animation dans un fichier pdf, comme ci-dessous (les animations dans un pdf fonctionnent seulement avec Acrobat Reader, Pdf-XChange ou Foxit Reader) :

1.5 Exemple 2

Dans cet exemple, nous allons illustrer ce qu'est une courbe cycloïde, en montrant une roue de rayon 1 roulant sur l'axe des abscisses. Initialement la roue aura son centre sur l'axe des ordonnées et la valve sera confondue avec l'origine. Au fur et à mesure que la roue avancera il nous faudra dessiner la trajectoire décrite par la valve.

1. Charger le modèle *Animation.mod* (F3).

2. Créer un élément graphique Utilisateur (Ctrl+U), appelé **fond**, avec la commande :

```
[
  view(-1.5,13.75,-0.75,3.25), Marges(0,0,0,0),
  dollar:=0, Axes(0,1+i)
]
```

3. Créer une variable globale appelée *courbe* et initialisée à 0. Elle contiendra la liste des positions successives de la valve, le premier étant l'origine.
4. Créer une variable globale appelée *Theta* et initialisée à 0, cette variable sera notre paramètre pour les éléments mobiles, elle représente l'angle $(\overrightarrow{CM}, -\vec{j})$ où *C* désigne le centre de la roue et *M* la valve.
5. Créer un élément graphique Utilisateur appelé *mobile* et avec la commande :

```
[
  Width:=8, Color:=blue,
  $M:=Theta+i*exp(i*(-pi/2-Theta)),
  Cercle(Theta+i,1), Line([Theta+i, M]), //cercle et rayon
  DotStyle:=cross, Point(Theta+i), // centre
  DotStyle:=bigdot, LabelDot( M, "M", "N", 1), // la valve
  Width:=12, Color:=red, Line(courbe) //la courbe
]
```

Nous avons constitué la frame initiale, passons aux macros

6. Dans la macro **Initialiser()**, ajouter les instructions :

```
Theta:=0, courbe:=0, ReCalc()
```

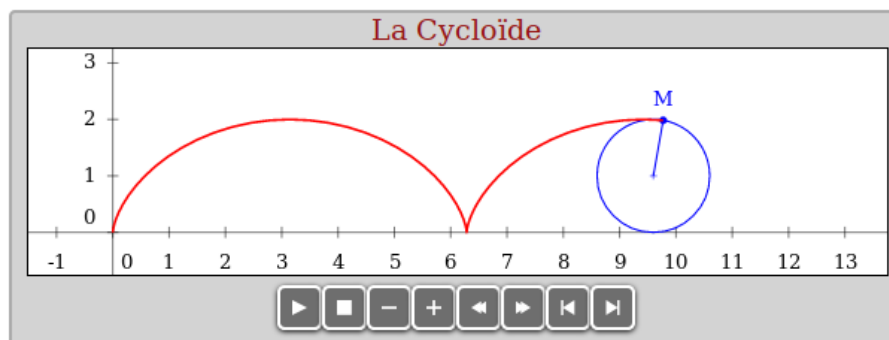
7. Dans la macro **Makeframe()**, ajouter les instructions :

```
if %1>1 then
  Inc(Theta, pi/18), Insert(courbe, Theta+i*exp(i*(-pi/2-Theta)) ),
  ReCalc(mobile)
fi
```

on tourne de 10 degrés, on ajoute la nouvelle position à la liste *courbe*, on déplace la roue et on redessine la trajectoire. La rotation ne commence qu'à la frame 2 de manière à ce que la figure initiale soit enregistrée comme frame 1. Il faudra en tenir compte pour le nombre de frames.

8. Avec le bouton *Frames* définir le nombre de frames à 73, ce qui correspond à 2 tours de roue (donc 2 arches de cycloïde) plus la position initiale.

Si on exporte cette animation pour le web, on doit voir ceci apparaître dans le navigateur :



2 Capture.mod

2.1 Présentation

Ce modèle est prévu pour venir se greffer sur un graphique déjà fait, et permettre à l'utilisateur de capturer des frames quand il le désire. Quand les captures sont terminées, l'utilisateur peut ensuite en faire une animation pour le web (si les frames ont été exportées en svg) ou bien un fichier pdf (si les frames ont été exportées en pgf). L'export

pdf permet de fabriquer un gif animé ou un fichier Flash, comme avec le premier modèle, mais il peut aussi être inclus dans un document \TeX avec le package *animate*.

On commence par charger son graphique, puis on charge le modèle *Capture.mod* (F3). À partir de là, si vous enregistrez votre graphique, alors à chaque ouverture de celui-ci, les macros de *Capture.mac* seront automatiquement chargées.

Depuis la version 1.2, le modèle définit deux nouvelles variables dans votre graphique :

- **TitreAnimation**, qui est une chaîne de caractères destinée à contenir le titre de l'animation lors de l'export pour le web.
- **AnimationSVG**, qui contient 0 ou 1, la valeur par défaut est 1, ce qui signifie que les frames seront exportées en svg (sinon elles seront exportées en pgf).

2.2 Le menu

Celui-ci apparaît dans la liste déroulante, voici quelques unes des options (il faut cliquer le bouton OK pour activer l'option) :

- **Capturer l'image** : initialement, la variable *NumFrame* a la valeur 1, cette option permet d'enregistrer le graphique en cours dans un fichier *frameXXX.svg* (ou pgf) où XXX désigne la valeur de *NumFrame*, puis cette valeur est incrémentée de 1. Il est possible d'automatiser la capture en appelant directement la macro *CaptureImage()* (sans paramètre).
- **Exporter vers web** : lorsque la variable *AnimationSVG* vaut 1, les frames sont exportées en svg dans le dossier *TeXgraphMac/Animation/animJs/animation/img*, après avoir choisi un titre, le fichier *TeXgraphMac/Animation/animJs/animation/index.html* devrait alors être lancé dans votre navigateur et permettre la visualisation de l'animation, à condition que celui-ci autorise javascript.
- **Exporter en pdf** : lorsque la variable *AnimationSVG* vaut 0, les frames sont exportées en pgf dans le dossier temporaire, après avoir choisi un nom de fichier (avec chemin, mais sans extension), les frames seront alors compilés en un fichier pdf qui portera ce nom, avec une image par page, et chaque page étant à la taille de l'image. Il est possible d'automatiser l'enregistrement en appelant directement la macro à un paramètre *MakePdf(<nombre de frames>)*, le nombre de frames déjà enregistrées est **NumFrame-1**. Si la variable *AnimationSVG* vaut 1, la compilation échouera puisque les frames auront été exportées en svg dans le dossier *TeXgraphMac/Animation/animJs/animation/img*.
- **Modifier** : permet de remplacer une frame par le graphique en cours, le numéro de frame doit être compris entre 1 et le nombre de frames déjà capturées. Par contre le fichier pdf n'est pas modifié, pour cela il faut de nouveau compiler.

2.3 Exemple 3

Nous avons repris le fichier *Hanoi.teg* (que l'on trouve sur le site de TeXgraph dans la galerie) :

- charger ce fichier dans TeXgraph,
- charger le modèle *Capture.mod*, par défaut la variable *AnimationSVG* à la valeur 1, ce qui signifie que les exports se feront en svg,
- cliquer le bouton *Nb disques* et mettre la valeur 4.
- cliquer le bouton *Nouveau* pour mettre la pile dans son état initial.
- faire une première capture avec l'option *Capturer l'image* dans la liste déroulante (ne pas oublier de cliquer OK pour valider),
- nous allons maintenant modifier la macro *deplacer()* pour l'obliger à faire deux captures : une après l'affichage de la flèche qui indique le déplacement à effectuer, et une après le déplacement du disque. La macro initiale est la suivante :

```
[ if %1=1 then source :=get(T1,1)-l/2+ep*i/2, Tourdep :=Tour1
  elif %1=2 then source :=get(T2,1)-l/2+ep*i/2, Tourdep :=Tour2
  else source :=get(T3,1)-l/2+ep*i/2, Tourdep := Tour3
fi,
if %2=1 then but :=get(T1,1)-l/2+ep*i/2, Tourbut :=Tour1,
  elif %2=2 then but :=get(T2,1)-l/2+ep*i/2, Tourbut :=Tour2
  else but :=get(T3,1)-l/2+ep*i/2, Tourbut :=Tour3
fi,
$disque :=Copy(Tourdep,1,1), $disque2 :=Copy(Tourbut,1,1),
if disque2=Nil then disque2 :=n+1 fi,
if disque<>Nil
then
```

```

    if disque2 < disque then Message("Déplacement impossible")
    else
      Eval(["Del(Tour", %1, ", 1, 1)"]),
      ReCalc(deplacement),
      Eval(["Insert(Tour", %2, ", ", disque, ", 1)"]),
      Inc(compteur, 1), ReCalc(Compteur, disques)
    fi
  fi
]

```

Nous plaçons une instruction *CaptureImage()* après *ReCalc(deplacement)*, et après *ReCalc(Compteur, disques)*, ce qui donne :

```

[ if %1=1 then source :=get(T1,1)-l/2+ep*i/2, Tourdep :=Tour1
  elif %1=2 then source :=get(T2,1)-l/2+ep*i/2, Tourdep :=Tour2
  else source :=get(T3,1)-l/2+ep*i/2, Tourdep := Tour3
  fi,
  if %2=1 then but :=get(T1,1)-l/2+ep*i/2, Tourbut :=Tour1,
  elif %2=2 then but :=get(T2,1)-l/2+ep*i/2, Tourbut :=Tour2
  else but :=get(T3,1)-l/2+ep*i/2, Tourbut :=Tour3
  fi,
  $disque :=Copy(Tourdep,1,1), $disque2 :=Copy(Tourbut,1,1),
  if disque2=Nil then disque2 :=n+1 fi,
  if disque <> Nil
  then
    if disque2 < disque then Message("Déplacement impossible")
    else
      Eval(["Del(Tour", %1, ", 1, 1)"]),
      ReCalc(deplacement), CaptureImage(), // <- modification ici
      Eval(["Insert(Tour", %2, ", ", disque, ", 1)"]),
      Inc(compteur, 1), ReCalc(Compteur, disques), CaptureImage(), // <- modification ici
    fi
  fi
]

```

- Cliquer ensuite le bouton *Solution*, les frames s'enregistrent toutes seules.
- Pour voir l'animation dans votre navigateur il suffit de sélectionner l'option *Exporter vers web* du menu déroulant, et de valider avec le bouton OK.

Pour avoir l'animation au format pdf, on recommence la même manipulation après avoir mis la variable *AnimationSVG* à la valeur 0, ce qui donne, après inclusion dans un document \LaTeX et le package *animate* :