Petit guide, pas à pas, pour `tkz-grapheur`

Un exemple fil rouge commenté

C. Pierquet

Juin 2026

Table des matières

1 Présentation du package

`tkz-grapheur` est un package LaTeX permettant de tracer des graphiques de fonctions de façon *intuitive et quasi algorithmique*, en s'appuyant sur TikZ et le moteur de calcul `xint`.

L'idée n'est en aucun cas de remplacer `pgfplots`, `tkz-fct` ou `tzplot` (packages qui sont de bien meilleure qualité !), mais de proposer des outils et une approche complémentaires !

1.1 Philosophie générale

L'idée centrale : on **définit** d'abord les objets mathématiques (fonctions, courbes, points), puis on les **exploite** graphiquement (images, antécédents, intersections, intégrales…) et enfin on **récupère** les coordonnées calculées pour les réutiliser dans le texte.
Toute commande TikZ classique reste utilisable *en parallèle* des commandes du package.

1.2 Chargement

Code

```
\usepackage{tkz-grapheur}
% options possibles [nonsiunitx] / [nontikzbabel] / [nonpgfplots]
```

1.3 Architecture du process, styles

Le schéma ci-dessous résume l'ordre logique des opérations dans un graphique :

Étape	Rôle
1	Ouvrir l'environnement, définir la fenêtre et la grille
2	Tracer les axes et grilles
3	Définir et tracer les courbes/points
4	Exploitations graphiques (intégrales, intersections…)
5	Récupérer les coordonnées pour les réutiliser dans le texte
6	(Optionnel) Surimpression des axes par-dessus l'intégrale

À retenir

Les étapes **2 et 6** concernent la même commande `\TracerAxesGrilles` utilisée avec les options `[Derriere]` (avant les courbes, sans graduation) et `[Devant]` (après les courbes, pour la surimpression).

Remarque

D'autres commandes graphiques sont disponibles dans `tkz-grapheur` (par exemple liées aux statistiques, aux lois de probabilités, aux courbes paramétriques/polaires, etc) mais elles seront pas explicitées dans ce document, qui restera destiné à un cadre *étude de fonctions de type lycée*.

Les styles de objets utilisés par `tkz-grapheur` sont tous définis par des styles `\tikzset{...}`, qui peuvent donc être (re)définis ou modifiés !

Code

```
%style par défaut pour les courbes
\tikzset{pflcourbe/.style={line width=1.05pt}}
```

2 Mise en place de l'environnement

2.1 L'environnement `GraphiqueTikz`

Tout graphique est encapsulé dans l'environnement principal :

Code

```
\begin{GraphiqueTikz}[options TikZ + clés fenêtre]<clés spécifiques>
  % commandes du package ou commandes compatibles
\end{GraphiqueTikz}
```

2.2 Données de fenêtre et de grille

Les paramètres de la fenêtre sont passés entre crochets `[...]` en même temps que les options Ti*k*Z classiques (`x=...,y=...`) :

Clé	Rôle	Défaut
`Xmin`, `Xmax`	bornes de l'axe des abscisses	-3, 3
`Ymin`, `Ymax`	bornes de l'axe des ordonnées	-3, 3
`Origx`, `Origy`	position de l'origine	0, 0
`Xgrille`, `Ygrille`	pas de la grille principale	1, 1
`Xgrilles`, `Ygrilles`	pas de la grille secondaire	0,5, 0,5
`x=...`, `y=...`	unités Ti*k*Z	–

Il existe aussi un mode *dimensions imposées* via les clés :

- `<Taille={larg/haut}>`, `<Largeur=...,XYratio=...>`,
- etc

qui laissent le package calculer les bonnes unités automatiquement.

2.3 Exemple fil rouge – ouverture de l'environnement

On va travailler tout au long de ce guide avec : $f(x) = x\,e^{-x/2}$ et $g(x) = \frac{x}{4}$ sur $[0;8]$.

Code

```
\begin{GraphiqueTikz}%
  [x=1.5cm, Xmin=0, Xmax=8.5, Xgrille=1, Xgrilles=0.5,
   y=2cm, Ymin=0, Ymax=2.5, Ygrille=0.5, Ygrilles=0.25]
  % -- à compléter dans les sections suivantes --
\end{GraphiqueTikz}
```

À noter que l'environnement ne *trace* rien, il créé la fenêtre dans laquelle les tracés seront réalisés. Cette zone a été matérialisée ici via la clé booléenne `<AffCadre>`.

Sortie

3 Axes et grilles

3.1 La commande \TracerAxesGrilles

Code

```
\TracerAxesGrilles[clés]{gradX}{gradY}
```

Les arguments gradX et gradY précisent quelles valeurs apparaissent sur chaque axe (syntaxe TikZ : listes explicites ou séquences a,b,...,z). Le mot-clé **auto** génère les graduations automatiquement depuis les paramètres de la grille principale.

Principales clés disponibles :

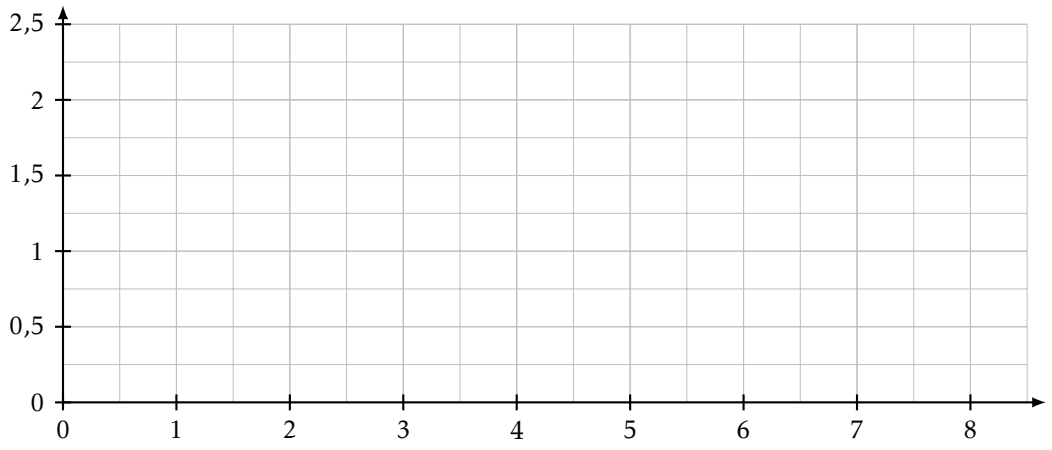
Clé	Effet
Elargir=...	allonge les flèches des axes
Police=\small	taille de la police des graduations
Grads=false	supprime les graduations (utile avant une intégrale)
Grille=false	supprime la grille (utile pour la surimpression)
Format=...	formate les graduations (frac, dfrac, trig, dtrig…)
Derriere	axes sans graduation, placés avant les courbes
Devant	graduation seule, sur-imprimée après les courbes

3.2 Fil rouge – axes et grille

Code

```
\begin{GraphiqueTikz}%
  [x=1.5cm, Xmin=0, Xmax=8.5, Xgrille=1, Xgrilles=0.5,
  y=2cm, Ymin=0, Ymax=2.5, Ygrille=0.5, Ygrilles=0.25]
  \TracerAxesGrilles[Elargir=2.5mm, Police=\small]%
    {0,1,...,8}%
    {0,0.5,...,2.5}%
\end{GraphiqueTikz}
```

Sortie



Remarque

Les nœuds (graphe-nw), (graphe-ne), (graphe-c), etc. (et leurs cousins liés aux axes : (axeox-e), (axeoy-n), (axes-orig)) sont créés automatiquement et utilisables pour placer du texte ou des étiquettes dans l'environnement.

4 Création d'objets – l'approche *path*

4.1 Définir et tracer une courbe

Le package sépare volontairement la **définition** du **tracé**, pour permettre d'insérer des éléments (intégrale, grille de fond) entre les deux.

Code

```
% Définir la fonction (pas de tracé)
\DefinirCourbe[Nom=cf, Debut=0, Fin=8]<f>{x*exp(-x/2)}
% Tracer la courbe (utilise la fonction f définie ci-dessus)
\TracerCourbe[Couleur=red, Debut=0, Fin=8]{f(x)}
% Ou tout en une fois avec la clé Trace :
\DefinirCourbe[Nom=cf, Debut=0, Fin=8, Trace, Couleur=red]<f>{x*exp(-x/2)}
```

Clés importantes de `\DefinirCourbe` :

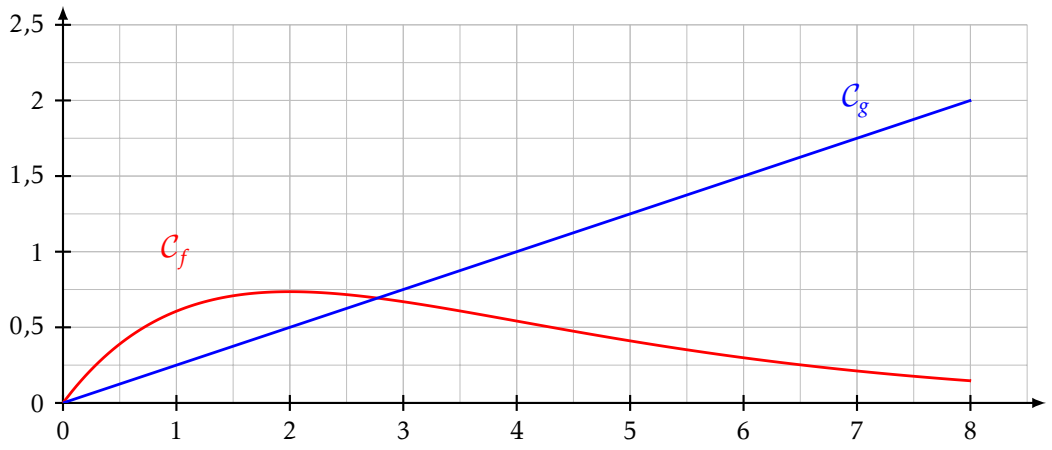
Clé	Rôle
`Nom=cf`	nom du *path* TikZ (indispensable pour l'exploitation)
`<f>`	nom de la fonction `xint` (pour les calculs)
`Trace`	tracer en même temps

4.2 Fil rouge – définition des courbes

Code

```
\begin{GraphiqueTikz}%
  [x=1.5cm, Xmin=0, Xmax=8.5, Xgrille=1, Xgrilles=0.5,
  y=2cm, Ymin=0, Ymax=2.5, Ygrille=0.5, Ygrilles=0.25]
  \TracerAxesGrilles[Elargir=2.5mm, Police=\small]{0,1,...,8}{0,0.5,...,2.5}
  % Définir f et g (sans tracer encore)
  \DefinirCourbe[Nom=cf, Debut=0, Fin=8]<f>{x*exp(-x/2)}
  \DefinirCourbe[Nom=cg, Debut=0, Fin=8]<g>{x/4}
  % Tracer les courbes
  \TracerCourbe[Couleur=red,   Debut=0, Fin=8]{f(x)}
  \TracerCourbe[Couleur=blue,  Debut=0, Fin=8]{g(x)}
  % Étiquettes
  \PlacerTexte[Police=\large,Couleur=red ]{(1,1)}{$\mathcal{C}_f$}
  \PlacerTexte[Police=\large,Couleur=blue]{(7,2)}{$\mathcal{C}_g$}
\end{GraphiqueTikz}
```

Sortie



5 Exploitation – intégrales et intersections

5.1 Représenter une intégrale

Deux variantes coexistent. La **version 2** (`\RepresenterIntegrale`) est la plus pratique dans la majorité des cas : elle s'appuie sur les *name paths* créés par `\DefinirCourbe[Nom=...]` et gère automatiquement quel que soit l'ordre des courbes.

Code

```
% sous une courbe nommée
\RepresenterIntegrale[Couleurs=blue/cyan!40]{cf}{a}{b}
% entre deux courbes nommées (ordre automatique)
\RepresenterIntegrale[Couleurs=red/pink!40]{cf}[cg]{a}{b}
```

À retenir

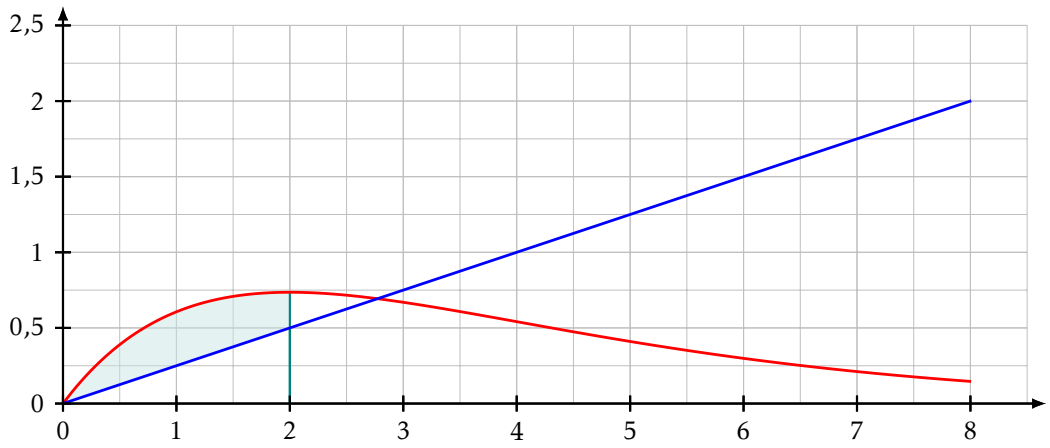
Pour que l'intégrale apparaisse *sous* les courbes, il faut la tracer **avant** les courbes. On utilise alors `\TracerAxesGrilles[Derriere]` (sans graduation) en premier, puis l'intégrale, puis les courbes, puis un second `\TracerAxesGrilles[Devant]` pour superposer graduation et axes.

Fil rouge – intégrale entre $\mathcal{C}_f$ et $\mathcal{C}_g$

Code

```
\begin{GraphiqueTikz}%
  [x=1.5cm, Xmin=0, Xmax=8.5, Xgrille=1, Xgrilles=0.5,
  y=2cm, Ymin=0, Ymax=2.5, Ygrille=0.5, Ygrilles=0.25]
  % Étape 1 : axes de fond (sans graduation)
  \TracerAxesGrilles[Derriere, Elargir=2.5mm]{0,1,...,8}{0,0.5,...,2.5}
  % Étape 2 : définitions (sans tracer)
  \DefinirCourbe[Nom=cf, Debut=0, Fin=8]<f>{x*exp(-x/2)}
  \DefinirCourbe[Nom=cg, Debut=0, Fin=8]<g>{x/4}
  % Étape 3 : intégrale entre les deux courbes (de x=0 à x=2)
  \RepresenterIntegrale[Couleurs=teal/teal!20]{cf}[cg]{0}{2}
  % Étape 4 : tracé des courbes
  \TracerCourbe[Couleur=red,  Debut=0, Fin=8]{f(x)}
  \TracerCourbe[Couleur=blue, Debut=0, Fin=8]{g(x)}
  % Étape 5 : surimpression des axes et graduations
  \TracerAxesGrilles[Devant, Elargir=2.5mm, Police=\small]{0,1,...,8}{0,0.5,...,2.5}
\end{GraphiqueTikz}
```

Sortie



5.2 Trouver les intersections de deux courbes

Code

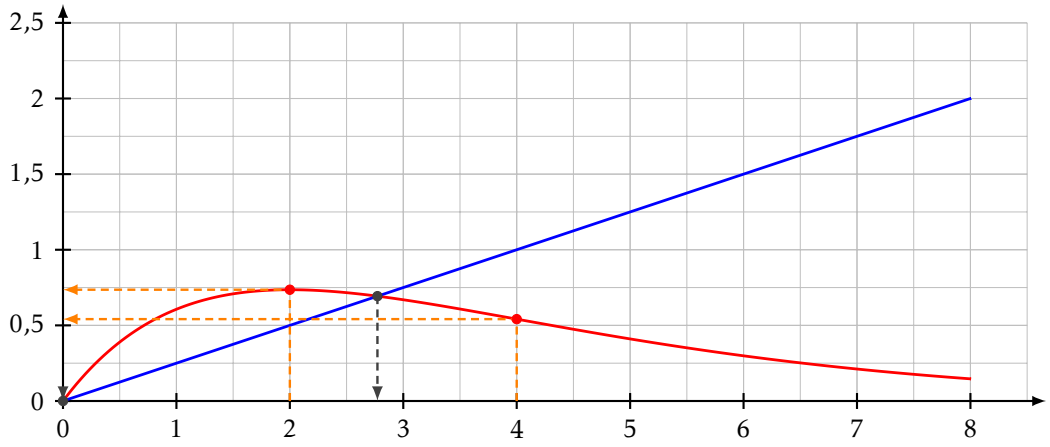
```
\TrouverIntersections[Nom=I, Couleur=darkgray, Aff]{cf}{cg}<\nbinter>

% Les nœuds sont créés sous les noms (I-1), (I-2), ...
% \nbinter contient le nombre d'intersections détectées.
% ListeRes=macroname stocke les abscisses en CSV.
```

Fil rouge – intersections et images

Code

```
\begin{GraphiqueTikz}%
  [x=1.5cm, Xmin=0, Xmax=8.5, Xgrille=1, Xgrilles=0.5,
  y=2cm, Ymin=0, Ymax=2.5, Ygrille=0.5, Ygrilles=0.25]
  \TracerAxesGrilles[Elargir=2.5mm, Police=\small]{0,1,...,8}{0,0.5,...,2.5}
  \DefinirCourbe[Nom=cf, Debut=0, Fin=8]<f>{x*exp(-x/2)}
  \DefinirCourbe[Nom=cg, Debut=0, Fin=8]<g>{x/4}
  \TracerCourbe[Couleur=red,  Debut=0, Fin=8]{f(x)}
  \TracerCourbe[Couleur=blue, Debut=0, Fin=8]{g(x)}
  % Intersections
  \TrouverIntersections[Nom=I, Couleur=darkgray, Aff, Traits]{cf}{cg}<\nbinter>
  % Images de quelques abscisses sur Cf
  \PlacerImages[Traits, Couleurs=red/orange]{f}{2,4}
\end{GraphiqueTikz}
```

Sortie



6 Récupération de coordonnées a posteriori

6.1 Les commandes de récupération

Une fois les nœuds créés (intersections, images, antécédents, extremums…), on peut en extraire les coordonnées et les réutiliser dans le texte du document :

Code

```
\RecupererAbscisse{(I-1)}[\monX]      % stocke l'abscisse dans \monX
\RecupererOrdonnee{(I-1)}[\monY]      % stocke l'ordonnée dans \monY
\RecupererCoordonnees{(I-2)}[\xB][\yB]  % les deux en une fois

% Affichage dans le texte :
$x_A \approx \num{\monX}$             % via siunitx
$x_A \approx \ArrondirNum[2]{\monX}$ % via la commande interne (2 décimales)
```

À retenir

Les valeurs récupérées sont des approximations flottantes (précision $\approx 10^{-4}$). La commande `\ArrondirNum[n]{...}` (fournie par le package) arrondit proprement à n décimales pour l'affichage.

6.2 Fil rouge – graphique complet avec récupération

Voici le graphique complet de notre exemple, avec récupération et affichage automatique des coordonnées des points d'intersection et du maximum de f.

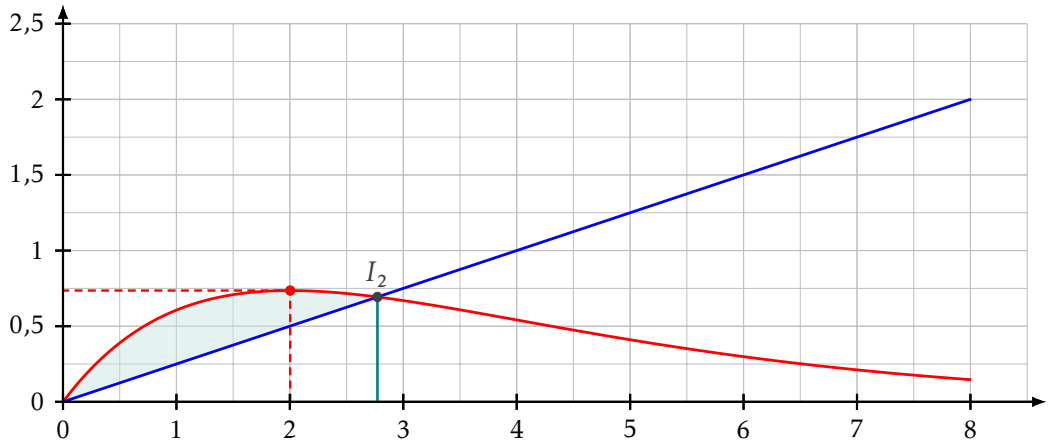
Code

```
\begin{GraphiqueTikz}%
  [x=1.5cm, Xmin=0, Xmax=8.5, Xgrille=1, Xgrilles=0.5,
  y=2cm, Ymin=0, Ymax=2.5, Ygrille=0.5, Ygrilles=0.25]
  % Grille de fond (avant l'intégrale)
  \TracerAxesGrilles[Derriere, Elargir=2.5mm]{0,1,...,8}{0,0.5,...,2.5}
  % Définitions
  \DefinirCourbe[Nom=cf, Debut=0, Fin=8]<f>{x*exp(-x/2)}
  \DefinirCourbe[Nom=cg, Debut=0, Fin=8]<g>{x/4}
  % Intersection (pour borner l'intégrale)
  \TrouverIntersections[Nom=I, Aff=false]{cf}{cg}
  \RecupererCoordonnees{(I-2)}[\xinter][\yinter]
  % Intégrale de 0 à l'intersection I-2 ( 5.76)
  \RepresenterIntegrale[Couleurs=teal/teal!20]{cf}[cg]{0}{\xinter}
  % Tracé des courbes
  \TracerCourbe[Couleur=red,  Debut=0, Fin=8]{f(x)}
  \TracerCourbe[Couleur=blue, Debut=0, Fin=8]{g(x)}
  % Maximum de f
  \TrouverMaximum[Debut=0, Fin=8]{f}[cf-max]
  \MarquerPts*[Couleur=red, Traits]{(cf-max)}
  % Affichage des intersections
  \MarquerPts[Couleur=darkgray]{(I-2)/$I_2$/above}
  % Surimpression axes
  \TracerAxesGrilles[Devant, Grille=false, Elargir=2.5mm, Police=\small]%
    {0,1,...,8}{0,0.5,...,2.5}
  % Récupération des coordonnées
  \RecupererCoordonnees{(I-2)}[\xinter][\yinter]
  \RecupererCoordonnees{(cf-max)}[\xmax][\ymax]
\end{GraphiqueTikz}
```

Code

```
Les courbes $\mathcal{C}_f$ et $\mathcal{C}_g$ se coupent (hors l'origine)
en $I_2 \approx (\ArrondirNum[2]{\xinter}\,;\,\ArrondirNum[2]{\yinter})$.

Le maximum de $f$ est atteint en $x \approx \ArrondirNum[2]{\xmax}$
et vaut $f(x) \approx \ArrondirNum[3]{\ymax}$.
```

Sortie



Sortie

Les courbes $\mathcal{C}_f$ et $\mathcal{C}_g$ se coupent (hors l'origine) en $I_2 \approx (2{,}77\,;\,0{,}69)$.
Le maximum de f est atteint en $x \approx 2$ et vaut $f(x) \approx 0{,}736$.

Remarque

Résumé du mécanisme de récupération :
Chaque commande d'exploitation (`\TrouverIntersections`, `\TrouverMaximum`, `\PlacerAntecedents`, `\TrouverAntecedents`…) crée des *nœuds TikZ nommés*. On récupère ensuite leurs coordonnées via `\RecupererAbscisse`, `\RecupererOrdonnee` ou `\RecupererCoordonnees`, et on les formate avec `\ArrondirNum` ou `\num` de `siunitx`.

7 Exemples complémentaires

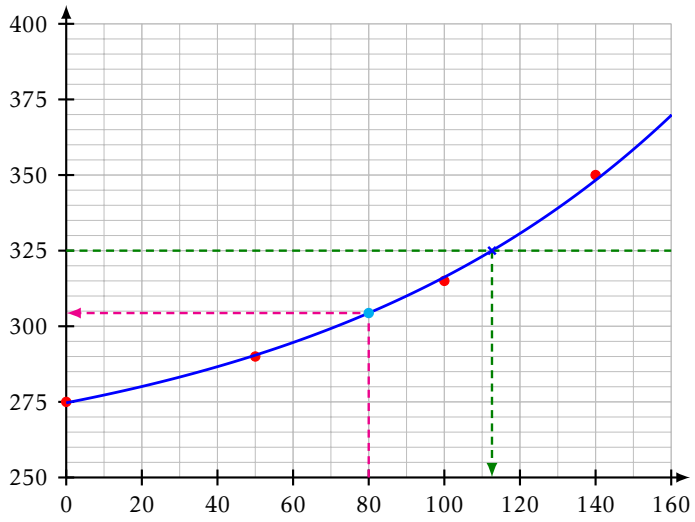
7.1 Statistiques

Code

```
\def\LISTEXX{0,50,100,140}\def\LISTEYY{275,290,315,350}%
ListeX := \LISTEXX\\
ListeY := \LISTEYY
\begin{GraphiqueTikz}
  [x=0.05cm,y=0.04cm,Xmin=0,Xmax=160,Xgrille=20,Xgrilles=10,
  Origy=250,Ymin=250,Ymax=400,Ygrille=25,Ygrilles=5]
  %préparation de la fenêtre
  \TracerAxesGrilles[Elargir=2.5mm,Police=\footnotesize]{auto}{auto}
  %nuage de points
  \TracerNuage[Style=o,CouleurNuage=red]{\LISTEXX}{\LISTEYY}
  %ajustement expoffset
  \TracerAjustement[Couleur=blue,Nom=ajust]<ajust>{expoff=250}{\LISTEXX}{\LISTEYY}
  %exploitations
  \PlacerImages[Couleurs=cyan/magenta,Traits]{ajust}{80}
  \PlacerAntecedents[Style=x,Couleurs=blue/green!50!black,Traits]{ajust}{325}
\end{GraphiqueTikz}

\xintexpoffreg[offset=250,round=3/1]{\LISTEXX}{\LISTEYY}%
On obtient $y=250+\num{\expregoffb}\text{e}^{\num{\expregoffa}x}$
```

Sortie

ListeX := 0,50,100,140
ListeY := 275,290,315,350



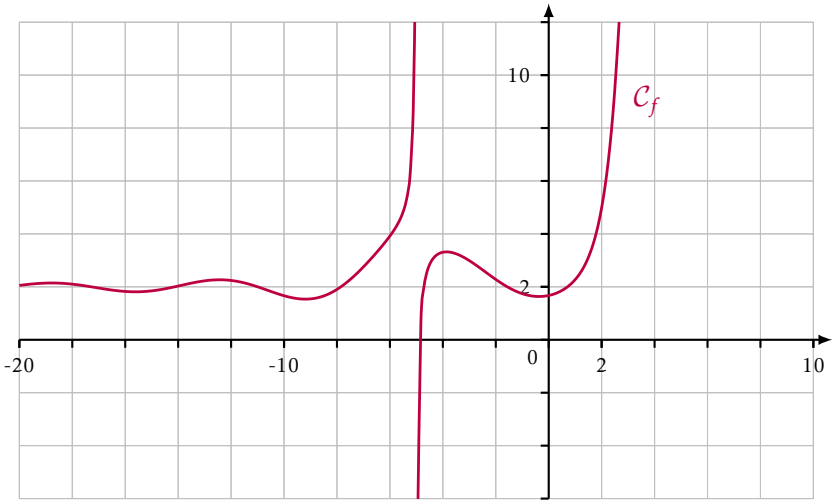
On obtient $y = 250 + 24{,}7\text{e}^{0{,}01x}$

7.2 Fonctions avec valeur(s) interdite(s)

Code

```
\begin{GraphiqueTikz}%
  [x=0.35cm,y=0.35cm,Xmin=-20,Xmax=10,Ymin=-6,Ymax=12,%
   Xgrille=2,Ygrille=2,Xgrilles=2,Ygrilles=2]%
   <TailleGrad=0pt/3pt>
  \TracerAxesGrilles%
    [Police=\scriptsize,Elargir=2.5mm,Origine,Grads=false]%
    {auto}{auto}
  \RajouterValeursAxeX[Police=\scriptsize]%
    {-20,-10,2,10}{-20,-10,2,10}
  \RajouterValeursAxeY[Police=\scriptsize]%
    {2,10}{2,10}
  \TracerCourbe%
    [Couleur=purple,ValeursInterdites=-5,DeltaVI=0.05]%
    {(exp(2*x-1)-2*cos(x))/(x+5)+2}
  \PlacerTexte[Couleur=purple]{(3.75,9)}{$\mathcal{C}_f$}
\end{GraphiqueTikz}
```

Sortie

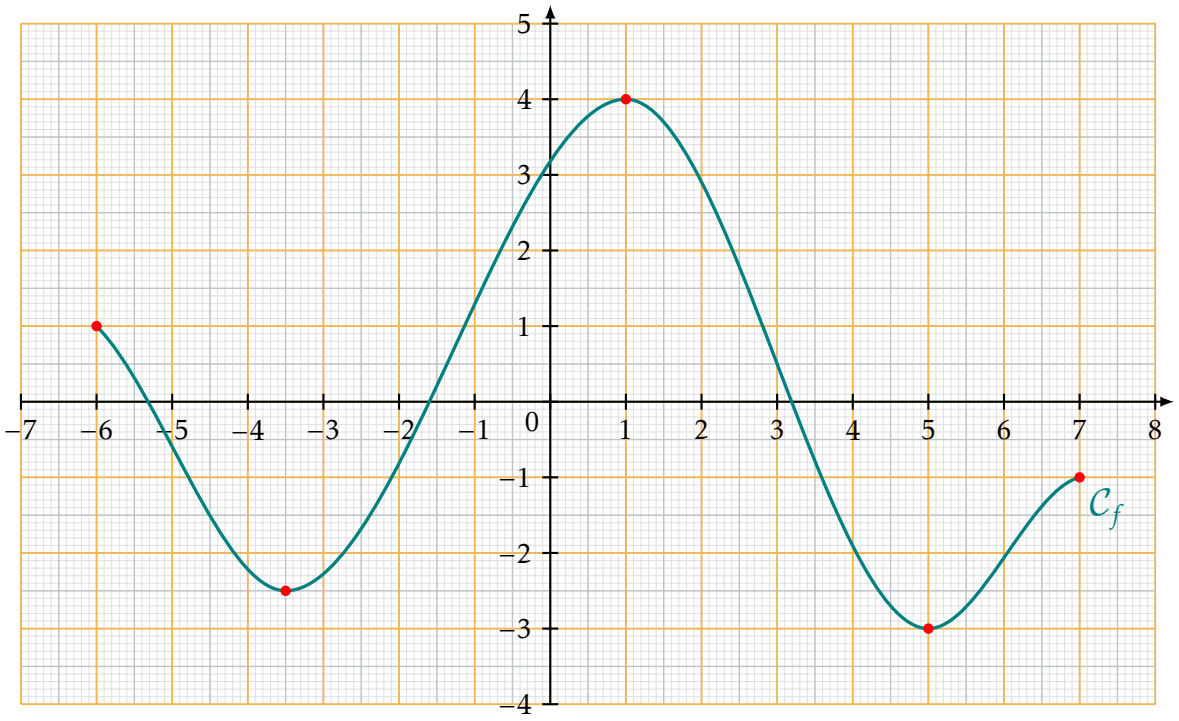


7.3 Interpolation cubique, avec gestion de la dérivée

Code

```
\begin{GraphiqueTikz}%
  [Xmin=-7,Xmax=8,Ymin=-4,Ymax=5,Xgrillei=0.5,Xgrilles=0.1,
   Ygrillei=0.5,Ygrilles=0.1]<Theme=standard>
  \tikzset{pflcourbe/.style={line width=1.25pt}}
  \TracerAxesGrilles*[Elargir=2.5mm,Origine,Grille Intermediaire]{auto}{auto}
  \def\donnneespline{-6/1/-1 § -3.5/-2.5/0 § 1/4/0 § 5/-3/0 § 7/-1/0.25}
  %définition et tracé du spline cubique
  \DefinirCourbeSpline[Alt,Nom=monspline,Trace,Couleur=teal]{\donnneespline}
  \MarquerPts*[Couleur=red]{(-6,1),(-3.5,-2.5),(1,4),(5,-3),(7,-1)}
  \PlacerTexte[Couleur=teal,Police=\Large,Position=below right]{(7,-1)}{$\mathcal{C}_f$}
\end{GraphiqueTikz}
```

Sortie

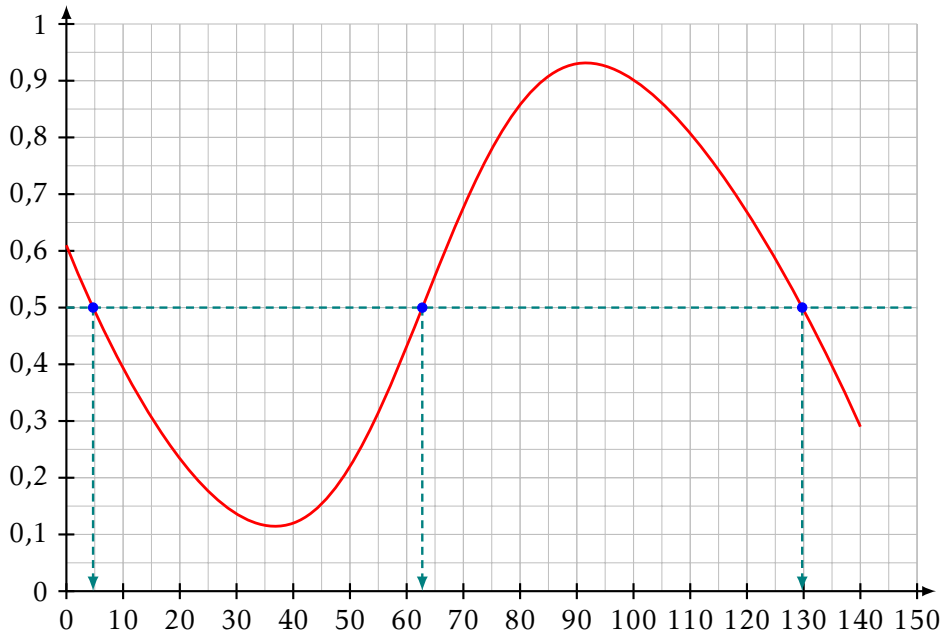


7.4 Courbes sous contraintes

Code

```
%la courbe est prévue pour qu'il y ait 3 antécédents
\ChoisirNbAlea[2]{0.75}{0.95}[\YrandMax]%
\ChoisirNbAlea[2]{0.05}{0.25}[\YrandMin]%
\ChoisirNbAlea*[2]{0.55}{\YrandMax}[\YrandA]%
\ChoisirNbAlea*[2]{\YrandMin}{0.45}[\YrandB]%

\begin{GraphiqueTikz}
  [x=0.075cm,y=7.5cm,Xmin=0,Xmax=150,Xgrille=10,Xgrilles=5,
   Ymin=0,Ymax=1,Ygrille=0.1,Ygrilles=0.05]
  \TracerAxesGrilles[Dernier,Elargir=2.5mm]{auto}{auto}
  \DefinirCourbeInterpo[Couleur=red,Trace,Nom=fonctiontest,Tension=0.75]
    {(0,\YrandA)(40,\YrandMin)(90,\YrandMax)(140,\YrandB)}
  \TrouverAntecedents[Aff=false,Nom=ANTECED]{fonctiontest}{0.5}
  \PlacerAntecedents[Couleurs=blue/teal,Traits]{fonctiontest}{0.5}
  \RecupererAbscisse{(ANTECED-1)}[\Aalpha]
  \RecupererAbscisse{(ANTECED-2)}[\Bbeta]
  \RecupererAbscisse{(ANTECED-3)}[\Cgamma]
\end{GraphiqueTikz}\\
Les solutions de $f(x)=\num{0.5}$ sont, par lecture graphique :
$\begin{cases}
  \alpha \approx \ArrondirNum[0]{\Aalpha} \\
  \beta \approx \ArrondirNum[0]{\Bbeta} \\
  \gamma \approx \ArrondirNum[0]{\Cgamma}
\end{cases}$.
```

Sortie



Les solutions de $f(x) = 0{,}5$ sont, par lecture graphique : $\begin{cases} \alpha \approx 5 \\ \beta \approx 63 \\ \gamma \approx 130 \end{cases}$.

8 Pour aller plus loin

8.1 Ce que peut faire `tkz-grapheur`

— Tracer des courbes classiques, d'interpolation
— Représenter et analyser des données statistiques
— Travailler sur des lois de probabilités
— Manipuler des courbes paramétriques, polaires, implicites
— Effectuer des calculs analytiques

Remarque

La plupart de ces fonctionnalités sont décrites :
— dans la documentation officielle (https://ctan.org/pkg/tkz-grapheur) ;
— sur le site compagnon (https://tkzgrapheur.cpierquet.fr).

8.2 Compléments généraux

— `xint` : https://ctan.org/pkg/xint
— `pgfplots` : https://ctan.org/pkg/pgfplots
— `tkz-fct` : https://ctan.org/pkg/tkz-fct
— `tzplot` : https://ctan.org/pkg/tzplot