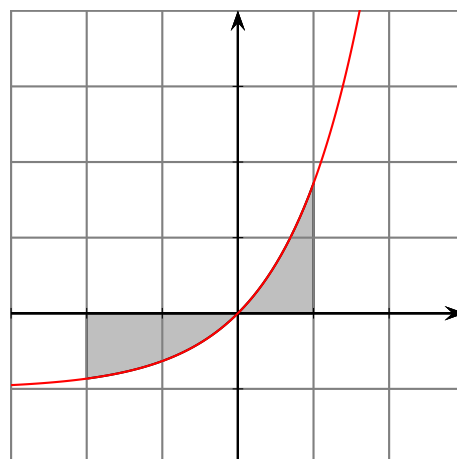
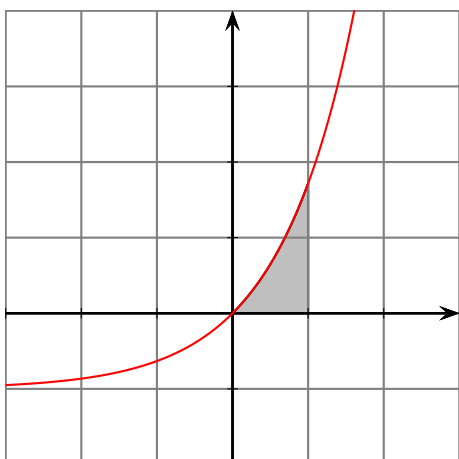


Domaines définis par des courbes

1 Domaine sous une courbe

Soit f la fonction définie sur \mathbf{R} par $f(x) = e^x - 1$.

On veut griser le domaine défini par la courbe représentant f , l'axe des abscisses et les droites d'équations $x = 0$ et $x = 1$ (figure de gauche). C'est le même procédé si la courbe passe en dessous de l'axe des abscisses comme entre $x = -2$ et $x = 1$ (figure de droite).

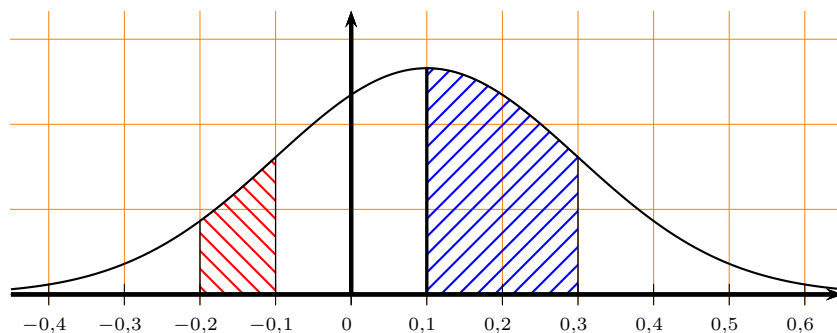


C'est `\pscustom` qui fait le boulot ; pour la figure de gauche, on écrit :

```

\pscustom[fillstyle=solid,fillcolor=lightgray]%% on remplit de gris
{%%
  \psplot{0}{1}{\f}%% tracé entre 0 et 1 de la fonction f
  \psplot{1}{0}{0}%% tracé entre 1 et 0 de la fonction nulle
  \closepath%% fermeture du domaine
}
    
```

Comme autre exemple, on peut dessiner un domaine sous une courbe représentant la fonction de densité d'une loi normale avec `\psGauss` ; on définit dans `mue` la moyenne et dans `sigma` l'écart-type. Cette instruction a besoin de l'extension `pst-func`.



On peut voir ainsi d'autres options de remplissage : `linestyle` et `filestyle`, `hlines` et `vlines`, `hatchcolor`.

2 Domaine entre deux courbes

2.1 On connaît les points d'intersection

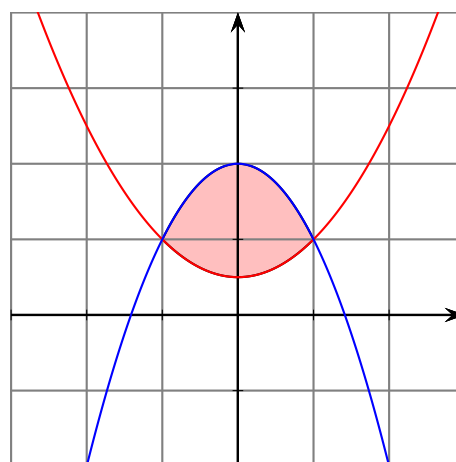
Soient f et g les fonctions définies sur \mathbf{R} par $f(x) = -x^2 + 2$ et $g(x) = \frac{x^2 + 1}{2}$.

On veut griser le domaine compris entre les courbes représentant ces deux fonctions.

Ces deux courbes se coupent en deux points d'abscisses -1 et 1 , et d'ordonnées 1 .

Là encore c'est `\pscustom` qui fait tout le travail, mais on n'utilise plus la fonction nulle mais la fonction g :

```
\pscustom[fillstyle=solid,fillcolor=pink]
{
\psplot{-1}{1}{\f}
\psplot{1}{-1}{\g}
\closepath
}
```



2.2 On ne connaît pas les points d'intersection

Dans le paragraphe précédent, les deux courbes se coupaient en deux points d'abscisses -1 et 1 ; c'était donc facile de dire dans `\pscustom` de griser entre $x = -1$ et $x = 1$.

Mais comment faire si on ne connaît pas les abscisses des points d'intersection ?

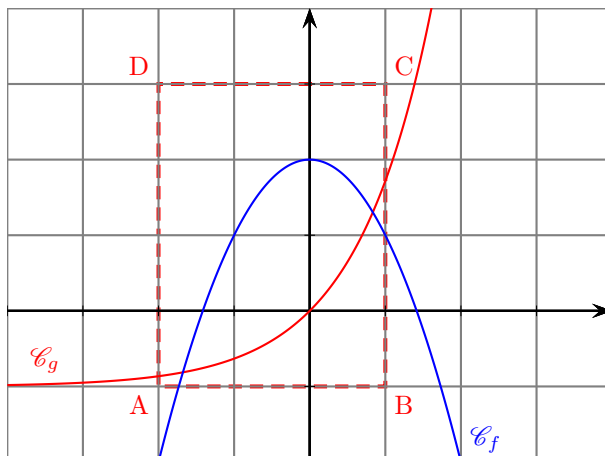
Certes on peut utiliser des valeurs approchées pour les abscisses des points d'intersection ; personne n'y verra de différence. Mais intellectuellement...

On va utiliser `\psclip` présenté dans la [chronique précédente](#) qu'il faut avoir regardée pour tout comprendre. Je vais détailler la procédure dans un exemple.

Soient f et g deux fonctions définies sur \mathbf{R} par $f(x) = e^x - 1$ et $g(x) = -x^2 + 2$.

On trace leurs représentations graphiques \mathcal{C}_f et \mathcal{C}_g .

Au passage, on cherche un rectangle à coordonnées entières qui contient le domaine à colorier ; on peut prendre le rectangle ABCD.



En fait dans le rectangle ABCD, il faut retenir les ordonnées des sommets, $y_{min} = -1$ et $y_{max} = 3$, et leurs abscisses $x_{min} = -2$ et $x_{max} = 1$.

Des deux courbes, on va désigner \mathcal{C}_f « la courbe du dessus », et \mathcal{C}_g « la courbe du dessous ».

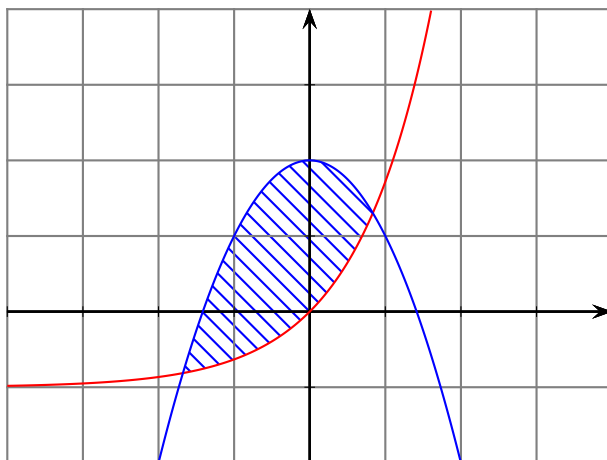
Avec `\pscustom`, on va découper la zone entre la courbe du dessus et y_{min} , ainsi que la zone entre la courbe du dessous et y_{max} , le tout entre x_{min} et x_{max} pour la courbe, et entre x_{max} et x_{min} pour les deux bouts de droites d'équations $y = y_{min}$ et $y = y_{max}$:

```
\pscustom[linestyle=none]          \pscustom[linestyle=none]
{                                     {
\psplot{x_min}{x_max}{fc du dessus} \psplot{x_min}{x_max}{fc du dessous}
\psplot{x_max}{x_min}{y_min}       \psplot{x_max}{x_min}{y_max}
}                                     }
```

Ce qui donne :

```
\pscustom[linestyle=none]          \pscustom[linestyle=none]
{                                     {
\psplot{-2}{1}{\f}                 \psplot{-2}{1}{\g}
\psplot{1}{-2}{-1}                 \psplot{1}{-2}{3}
}                                     }
```

Ce sont ces deux `\pscustom` que l'on va mettre comme zone-limite dans `\psclip`.



Voici le code :

```
\psset{unit=1cm}
\def\xmin {-4} \def\xmax {4} \def\ymin {-2} \def\ymax {4}
\begin{pspicture*}(\xmin,\ymin)(\xmax,\ymax)
%\uput[dl](0,0){0}
\def\f{x neg x mul 2 add}%          fonction du dessus
\def\g{2.7183 x exp 1 sub}%        fonction du dessous
\psclip{
\pscustom[linestyle=none]
{\psplot{-2}{1}{\f}
\psplot{1}{-2}{-1}}
\pscustom[linestyle=none]
{\psplot{-2}{1}{\g}
\psplot{1}{-2}{3}}
```

```

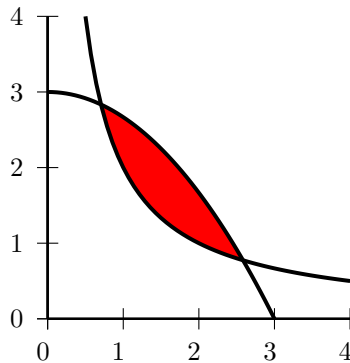
}%% fin du découpage de la zone-limite
\psframe[hatchcolor=blue,fillstyle=vlines](-2,-1)(1,3)
\endpsclip
\psgrid[subgriddiv=1,gridlabels=0,gridcolor=gray]
\psaxes[arrowsize=3pt 3,ticks=-2pt 2pt,labels=none]%%
{->}(0,0)(\xmin,\ymin)(\xmax,\ymax)
\psset{yMaxValue=\ymax,yMinValue=\ymin}
\psplot[plotpoints=1000,linewidth=red]{\xmin}{\xmax}{\g}
\psplot[plotpoints=1000,linewidth=blue]{-2}{2}{\f}
\end{pspicture*}

```

Attention ! La présence de `\psclip` détruit l'étoile de `pspicture*` : les courbes ne sont plus limitées dans le rectangle défini par `\begin{pspicture*}`. Il faut donc, soit gérer les abscisses pour que les images ne sortent pas du cadre, soit utiliser les variables `yMinValue` et `yMaxValue` ainsi :

```
\psset{yMaxValue=\ymax,yMinValue=\ymin}
```

Le mode d'emploi de PSTricks dans sa version la plus récente (2007), `pst-user`, donne l'exemple ci-dessous dont je me suis inspiré.



Voici le code :

```

\begin{pspicture}(4,4)
\psclip
{%% début zone-limite
\pscustom[linewidth=none]
{
\psplot{.5}{4}{2 x div}
\lineto(4,4)
}%% fin du 1er \pscustom
\pscustom[linewidth=none]
{
\psplot{0}{3}{3 x x mul 3 div sub}
\lineto(0,0)
}%% fin du 2d \pscustom
}%% fin de la zone-limite
\psframe*[linewidth=red](0,0)(4,4)
\endpsclip
\psplot{.5}{4}{2 x div}
\psplot{0}{3}{3 x x mul 3 div sub}
\psaxes(4,4)
\end{pspicture}

```

N'étant pas très à l'aise avec les `\lineto`, je n'ai pas utilisé cette instruction dans mon exemple.