

## Exemple de calculs du produit scalaire entre fonctions

```
In [2]: from sympy import * # la librairie pour le calcul symbolique
%config InlineBackend.figure_formats = {'png', 'retina'} # makes nicer graphics
init_printing(use_latex='mathjax') # pour afficher le resultat joliment
x = symbols('x') # par la suite, x sera une variable de maths symbolique
```

### produit scalaire entre $\sin(x)$ et $\sin(2x)$

```
In [3]: f=sin(x)
g=sin(2*x)
```

exprimons, juste pour vérifier, le produit de ces fonctions

```
In [4]: f*g
```

```
Out[4]: sin(x) sin(2x)
```

Affichons l'expression de la primitive de fg. Le `simplify` n'est pas indispensable mais fournit une expression un peu plus simple. On n'utilise pas cette primitive dans les calculs ultérieurs, c'est juste ici pour montrer qu'on peut le calculer si besoin.

```
In [10]: simplify(integrate(f*g, x))
```

```
Out[10]:  $\frac{2}{3}\sin^3(x)$ 
```

On calcule numériquement ce qui nous est vraiment utile : l'intégrale du produit de f et g.

```
In [11]: borne_depart=0
borne_arrivee=2*pi
integrate(f*g, (x,borne_depart,borne_arrivee))
```

```
Out[11]: 0
```

On constate bien que les deux fonctions sont orthogonales

### produit scalaire entre $\sin(x)$ et $\sin(x)$ - (c.a.d. $\|\sin(x)\|^2$ )

```
In [12]: f=sin(x)
integrate(f*f, x) # On aurait pu utiliser simplify en plus, pour avoir une autre formulation de cette expression
```

```
Out[12]:  $\frac{x}{2} - \frac{1}{2}\sin(x)\cos(x)$ 
```

```
In [13]: borne_depart=0
borne_arrivee=2*pi
integrate(f*f, (x,borne_depart,borne_arrivee))
```

```
Out[13]:  $\pi$ 
```

On constate bien que les deux fonctions ne sont évidemment pas orthogonales, puisqu'il s'agit du produit scalaire d'une fonction avec elle-même, donc la norme au carré