In [213] :
```
%matplotlib inline
import matplotlib as mpl
import matplotlib.colors as colors
import matplotlib.pyplot as plt

import numpy as np
import scipy.special as spec
import scipy.integrate as inte

import sympy as sp

sp.init_printing()
```

1 Position du problème

2 Construction du champ magnétique B

In [258] :
```
r = sp.symbols('r')
t = sp.symbols('t')
omega, phi = sp.symbols('omega phi')

c = 3e8       # Célérité de la lumière dans le vide
k = omega/c   # Nombre d'onde

# Composante du champ magnétique associée à la pulsation omega
B_component = (sp.besselj(0,k*r)-sp.bessely(0,k*r)/(1+sp.exp(-omega*1e-5))) \
    *sp.cos(omega*t-phi)

# Création du champ (expression et fonction) en sommant les composantes
def create_Bfield(puls,phas):
    spectr = zip(puls,phas)
    B_field = sum([B_component.subs({omega:om, phi:ph}) for (om,ph) in spectr])
    B_function = sp.lambdify((r, t), B_field,
        modules=['numpy',{"besselj":spec.jn, "bessely":spec.yn}])
    return B_field, B_function
```

In [312] :
```
def graphe_B(times):
    '''
    Construit les graphes du champ magnétique B aux temps donnés dans la liste
    "times"
    '''
    radii = np.linspace(rmin, rmax, 10*rmax)

    fig, ax = plt.subplots(1,1, figsize=(8,5), dpi=400)

    if hasattr(times, '__iter__'):
        for ti in times:
            leg = r'$t= {:g}$'.format(ti)
```

```python
            leg = leg + r"$\ \mathrm{s}$"
            champ = B_function(radii, ti)
            ax.plot(radii, champ, label=leg)
    else:
        champ = B_function(radii, ti)
        leg = r'$t = {:g}$'.format(ti)
        leg = leg + r"$\ \mathrm{s}$"
        ax.plot(radii, champ, label=leg)

    ax.grid()
    ax.legend()
    ax.set_xlabel("Distance $r$ (m)")
    ax.set_ylabel("Valeur du champ (T)")
    ax.set_title(r'Champ magnétique ' + r'$\mathbf{B}$' \
                 + ' créé par un courant variable')

    fig.tight_layout()

    return fig, ax
```

In [285] :
```python
def build_field(t):
    """
    Portrait du champ magnétique à l'instant t
    """
    wind = rmax

    Y, X = np.ogrid[-wind:wind:wind*10j, -wind:wind:wind*10j]

    def field_func(x,y):
        r = np.sqrt(x*x+y*y)
        Btheta = B_function(r, t)
        direct = np.array([-y/r, x/r])
        return Btheta*direct

    BX, BY = field_func(X, Y)


    fig, ax = plt.subplots(1, 1, figsize=(8,8))
    ax.grid(False)
    ax.set_aspect('equal')

    ax.set_xlim((-wind,wind))
    ax.set_ylim((-wind,wind))

    title_text = r'Champ magnétique $\mathbf{B}$ à '
    title_text += r"$t={:g}$".format(t)
    title_text += r" $\mathrm{s}$"
    ax.set_title(title_text)
```

```python
    intensity = np.sqrt(BX**2+BY**2)
    intensity = np.nan_to_num(intensity)

    heat = ax.imshow(intensity,
                     norm=colors.LogNorm(),
                     extent=[-wind, wind, -wind, wind],
                     alpha=0.6)
    cbar = fig.colorbar(heat, label='Intensité du champ (T)')

    strm = ax.streamplot(X,Y, BX, BY,
        arrowstyle='->',
        color='k',
        cmap='inferno',
        linewidth=0.8,
        arrowsize=2,
        density=1.4,
        )

    fig.tight_layout()

    return fig
```

3 Tracés

3.1 Données initiales

Entrez dans la variable `freqs` les fréquences du courant voulu, et dans `phas` les phases associées, et exécutez la cellule (`Ctrl + Entrée` sur le clavier) pour définir la fonction de champ :

In [262] :
```python
freqs = [n*1e7 for n in range(5,7)] + \
    [n*1e4 for n in range(2,5)]
puls = 2*np.pi*np.asarray(freqs) # Pulsations associées aux fréquences

phases = [0,0,0.7,0.8,0.3]

B_field, B_function = create_Bfield(puls, phases)
B_field
```

Out[262] :

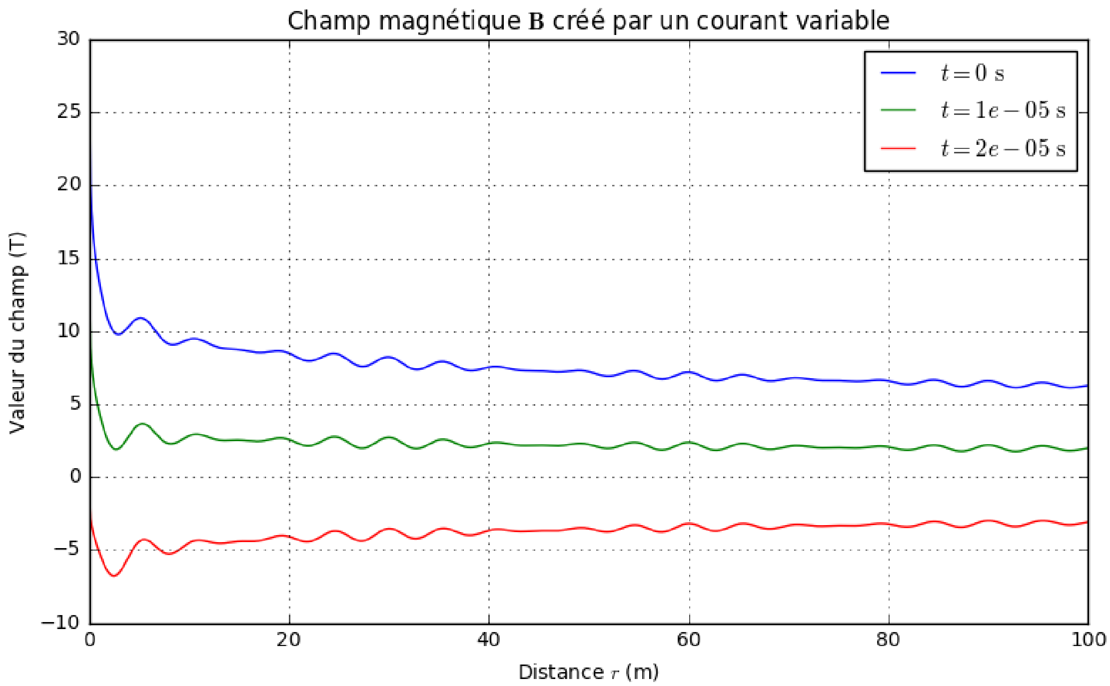
$(J_0(0.000418879020478639r) - 0.778446653450105Y_0(0.000418879020478639r))\cos(125663.706143592t - 0.7) + (J$

La cellule suivante définit les distances minimale et maximale pour lesquels tracer le profil du champ magnétique :

In [292] :
```python
rmin = 0.01
rmax = 100
```

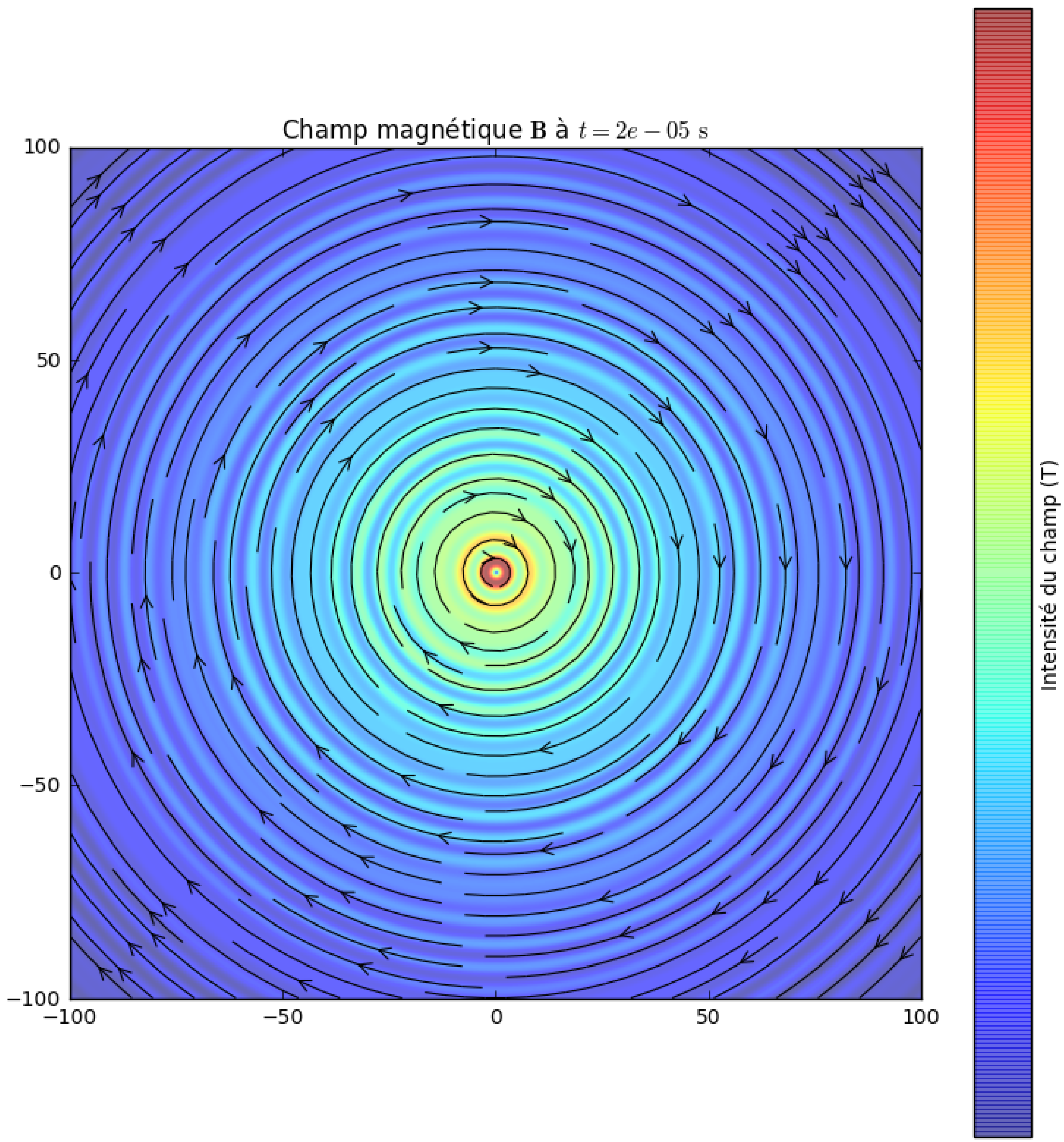
In [313] : times = [1e-6*k for k in [0, 10, 20]]

```
fig_Btheta, ax_Btheta = graphe_B(times)
fig_Btheta.savefig('profil_champ.png')
```



In [289] : t = times[2]   # Temps auquel calculer le portrait du champ (pas de liste)

```
output_field = build_field(t)
output_field.savefig("champmag_courant_variable.png")
```



3.2 Animations

work in progress

L'idée est de réécrire les fonctions de la partie précédente pour qu'elles produisent un fichier .mp4 qui donne un profil du champ magnétique évoluant au cours du temps (utiliser la classe `animate` de `matplotlib`).

4 Théorie (non-trivial) (BAC + 2)

Le champ magnétique $\mathbf{B}$ dérive d'un champ $\mathbf{A}$ appelé *potentiel vecteur* : $\mathbf{B} = \nabla \wedge \mathbf{A}$. Par symétrie cylindrique, on a $\mathbf{B}(\mathbf{r}, t) = B(r, t)\mathbf{e}_\theta$. Par suite $\mathbf{A}(\mathbf{r}, t) = A(r, t)\mathbf{e}_z$.

Le potentiel vecteur $\mathbf{A} = A(r,t)\mathbf{e}_z$ est solution de l'équation d'onde

$$\Delta \mathbf{A} - \frac{1}{c^2}\frac{\partial^2 \mathbf{A}}{\partial t^2} = -\mu_0 \mathbf{J}(r,t), \tag{1}$$

avec $\mathbf{J}(r,t) = \dfrac{i(t)\delta(r)}{2\pi r}\mathbf{e}_\theta$ la densité volumique de courant.

Pour un courant sinusoïdal $i(t) = I\exp(i\omega t)$, le potentiel s'écrit $A(r,t) = f(r)\exp(i\omega t)$ et l'équation aux dérivées partielles se réduit à

$$\frac{1}{r}\frac{\mathrm{d}}{\mathrm{d}r}\left(r\frac{\mathrm{d}f}{\mathrm{d}r}\right) + k^2 f(r) = -\frac{\mu_0 I\delta(r)}{2\pi r}, \tag{2}$$

avec $k = \dfrac{\omega}{c}$.

La solution générale prend la forme

$$f(r) = CJ_0(kr) + DY_0(kr)$$

où C et D dépendent de la pulsation ω du courant, et J_0, Y_0 sont les 0-ièmes fonctions de Bessel de la première et seconde espèce, solutions de

$$xy''(x) + y'(x) + xy(x) = 0.$$