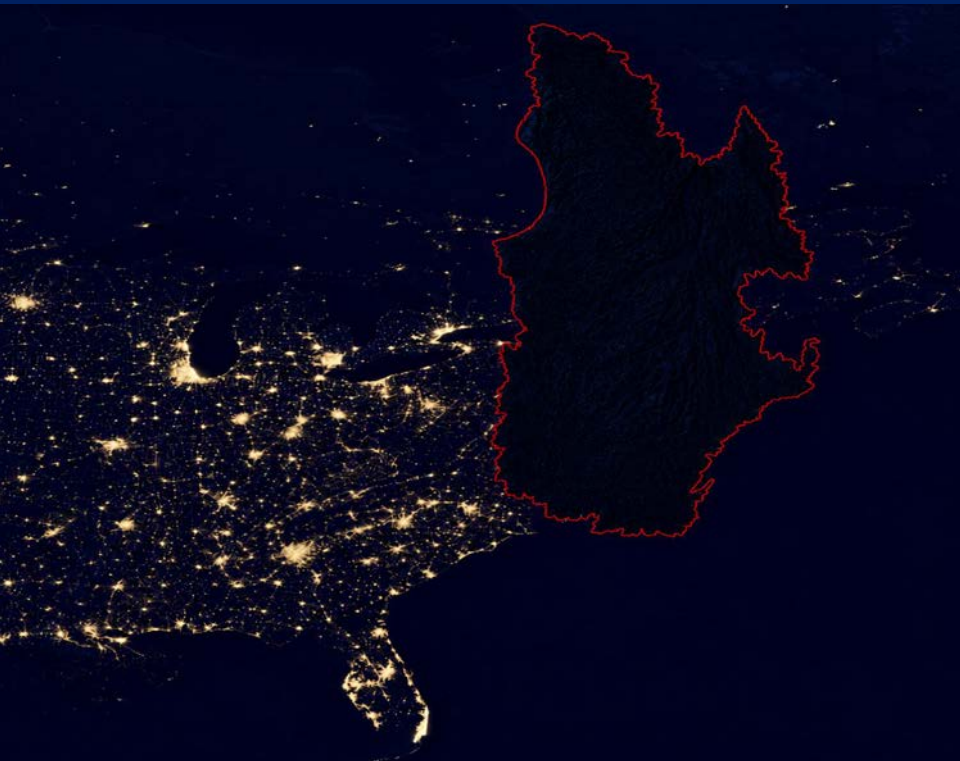


Effet des tempêtes solaires sur les réseaux électriques

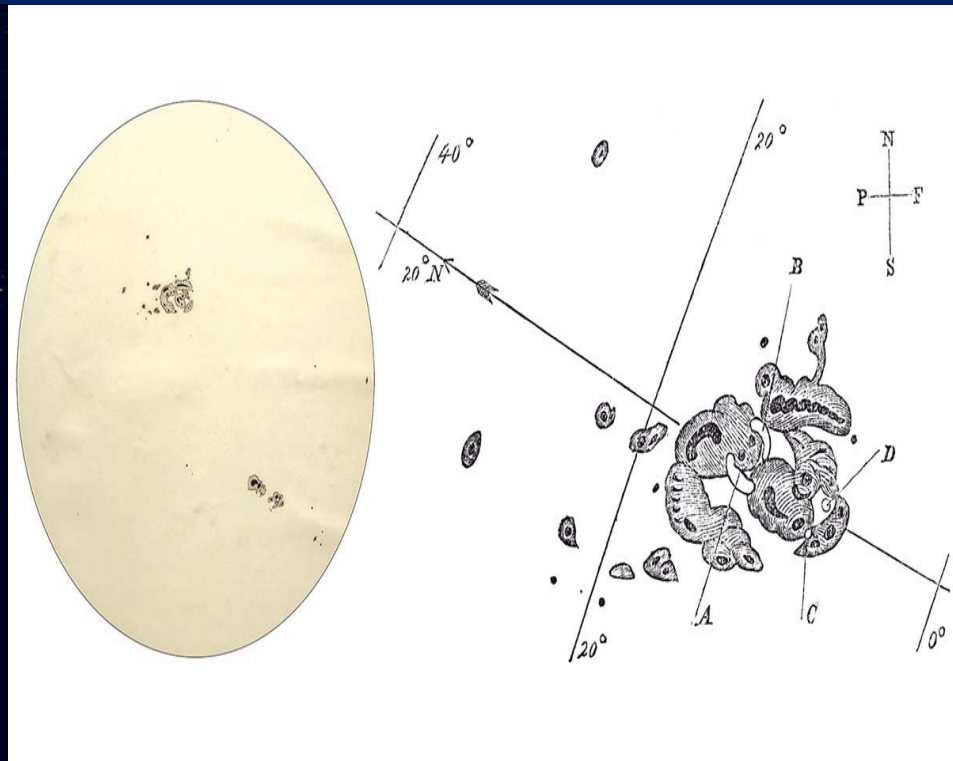
Yassine BASLI -

2025-2026

Le phénomène



Blackout du Québec, mars 1989 (illustration) :
pertes économiques estimées à
2 milliards de dollars

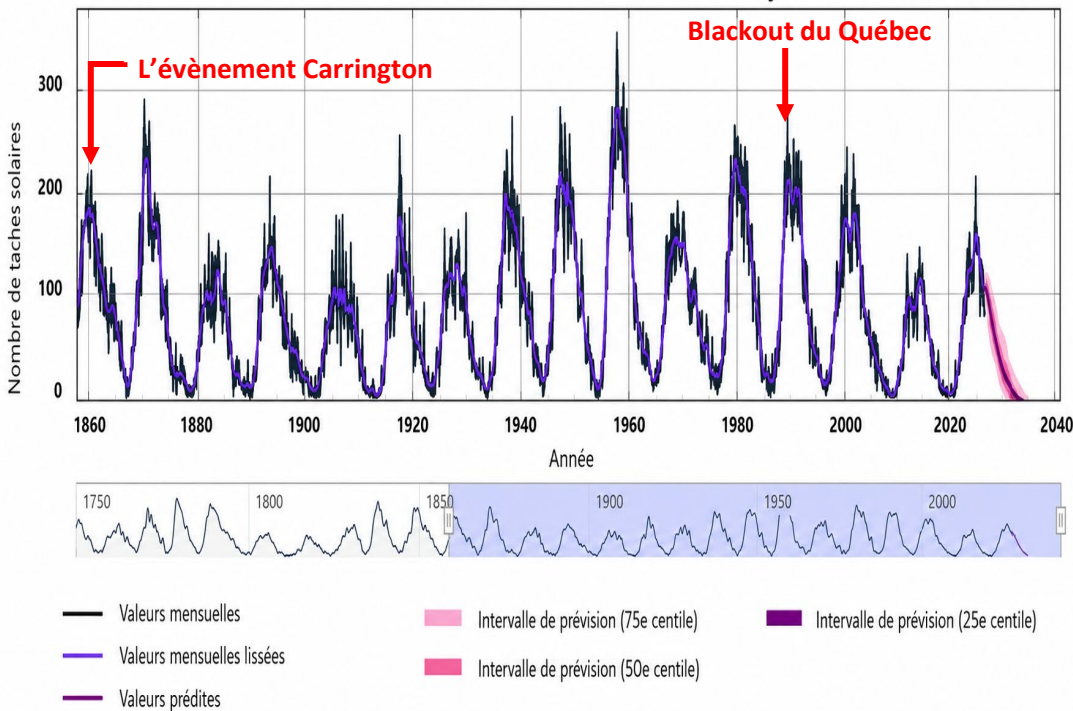


L'événement Carrington (1859) :
La réapparition d'un tel événement de nos
jours conduirait à des dégâts irréversibles
et des pertes économiques estimées à
20 billions de dollars

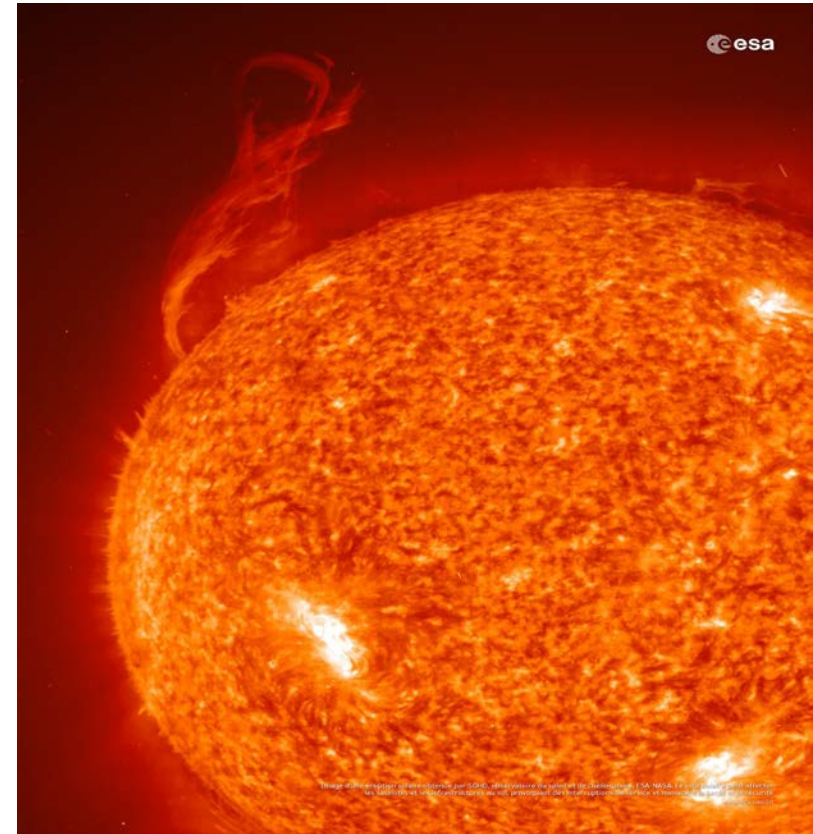
Source : <https://science.nasa.gov/earth/earth-observatory/earth-at-night/maps/> (adapté) , <https://knowablemagazine.org/docserver/fulltext/10.1146/knowable-091721-1/i-carrington-event.jpg>

Constat

Évolution du nombre de taches solaires au cours des cycles solaires



(a) Evolution du nombre de taches solaires depuis 1855



(b) Image d'une éruption solaire

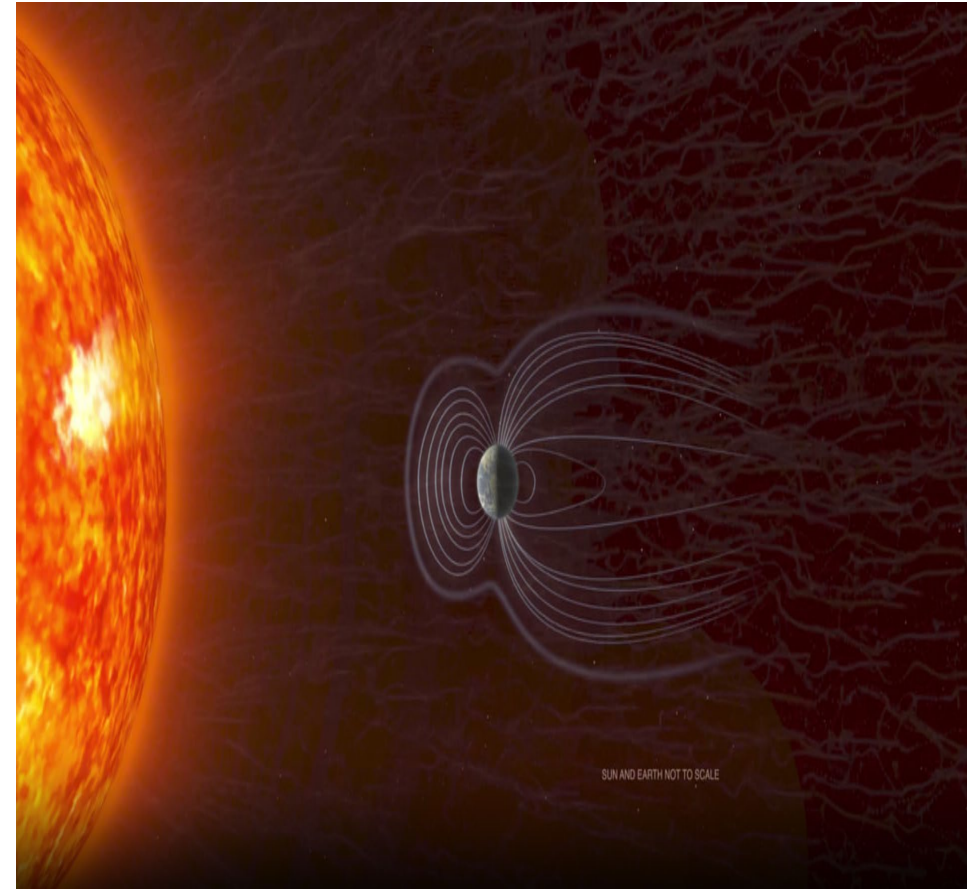
Quelle est la corrélation

Les phénomènes électriques



Image d'un poste électrique prise à la centrale électrique de Radès (Tunisie)

Les tempêtes solaires



Source : NASA's Goddard Space Flight Center Conceptual Image Lab.

1 Etude théorique

- Introduction de l'effet électromagnétique des éruptions solaires
- Modélisation électrique
- Détermination de l'expression du courant induit géomagnétiquement

2 Application à des configurations réelles

- Tempête du 13 mars 1989 et réseau de transport électrique d'Hydro-Québec
- Résolution numérique pour le réseau d'Hydro-Québec et comparaison avec les mesures réels

3 Sensibilité régionale

- Bilan sur la visite d'une centrale électrique
- Etude qualitative sur la dépendance géographique

Etude Théorique

Le vent solaire

Vent solaire



Flux continu de particules
chargées

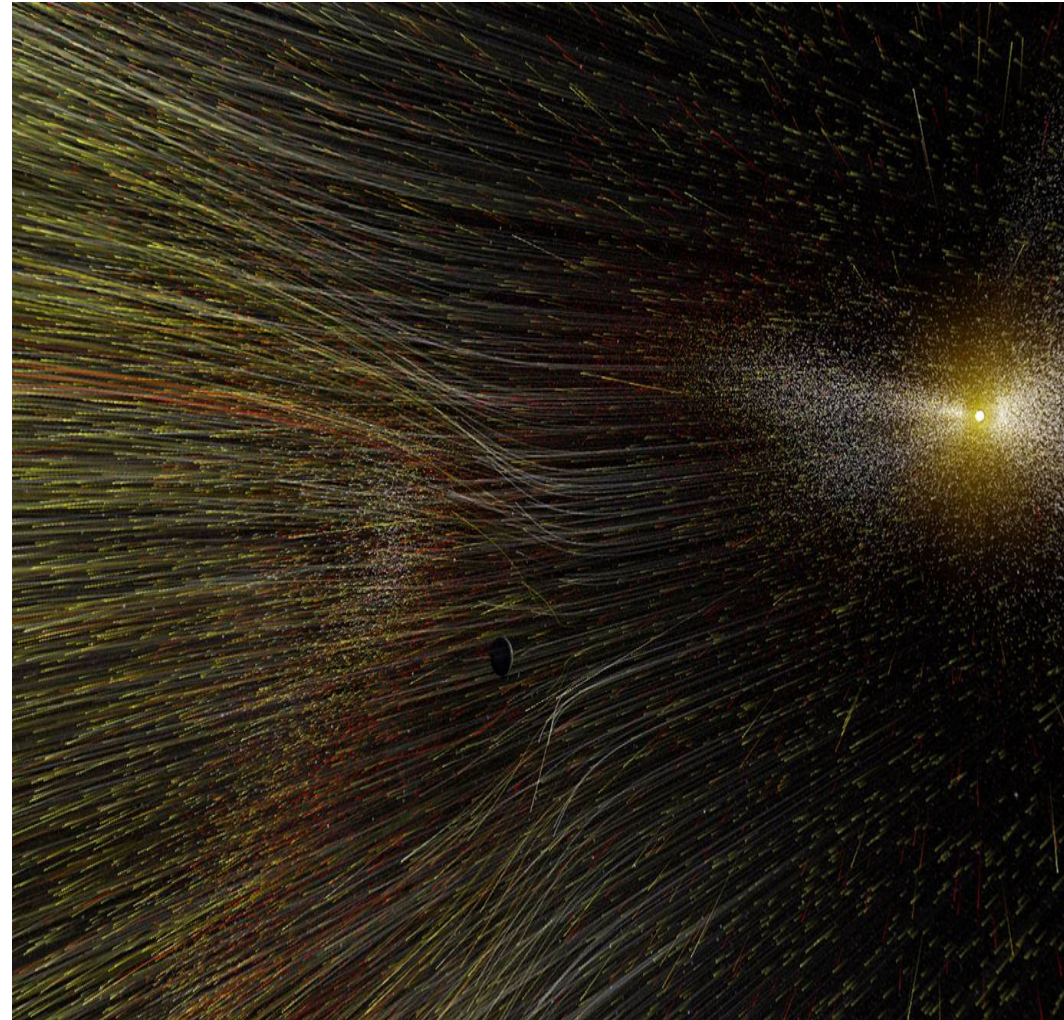


Création d'un champ
électromagnétique



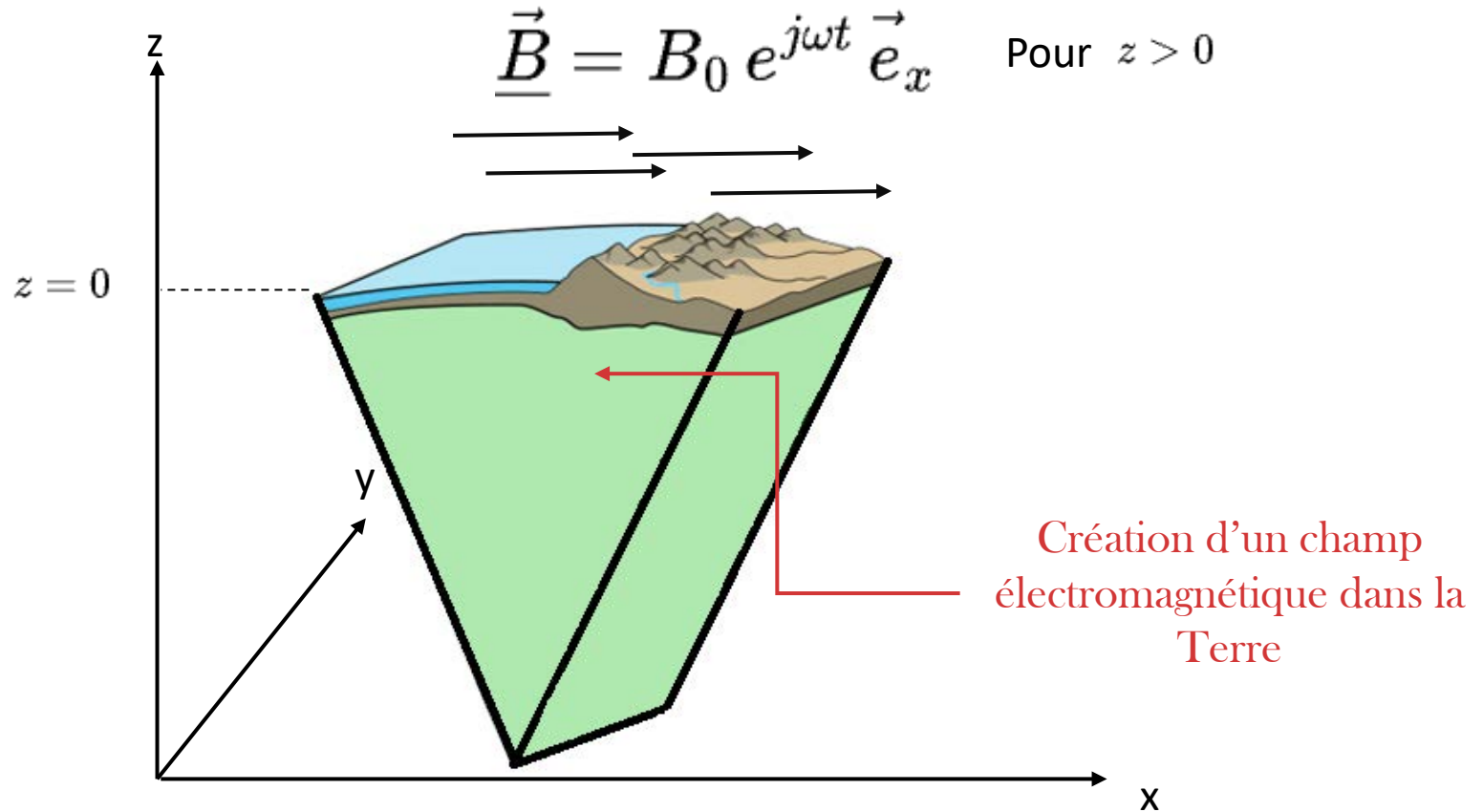
Perturbation géomagnétique

Image illustrative du vent solaire



Source : NASA's Goddard Space Flight Center Conceptual Image Lab.
https://svs.gsfc.nasa.gov/14892/#media_group_378813

Perturbation géomagnétique harmonique



- ❖ Propriété typique des perturbations dues aux éjections de masse coronales :

$$f \in [10^{-4}, 1] \text{ Hz}$$

Perturbation géomagnétique harmonique

□ Expression du champ crée dans la Terre :

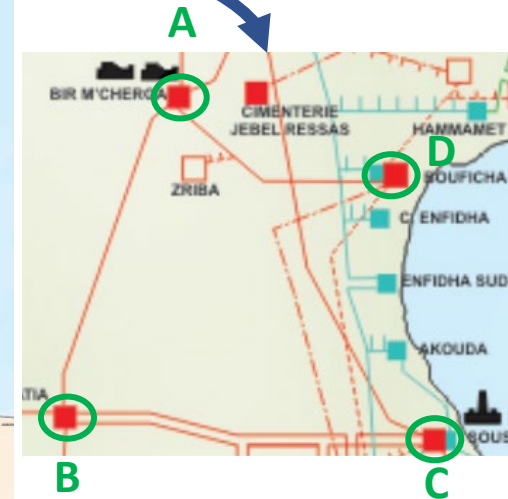
Expression du champ magnétique

$$\vec{B}(z, t) = B_0 e^{j(\omega t + \frac{z}{\delta})} e^{\frac{z}{\delta}} \vec{e}_x \quad \forall z < 0$$

Avec : $\delta = \sqrt{\frac{2}{\mu_0 \gamma \omega}}$: L'épaisseur de peau du matériau

Pour la Terre : $\delta \approx 50 \text{ km}$

Effet d'induction



(a) Lignes à haute tension

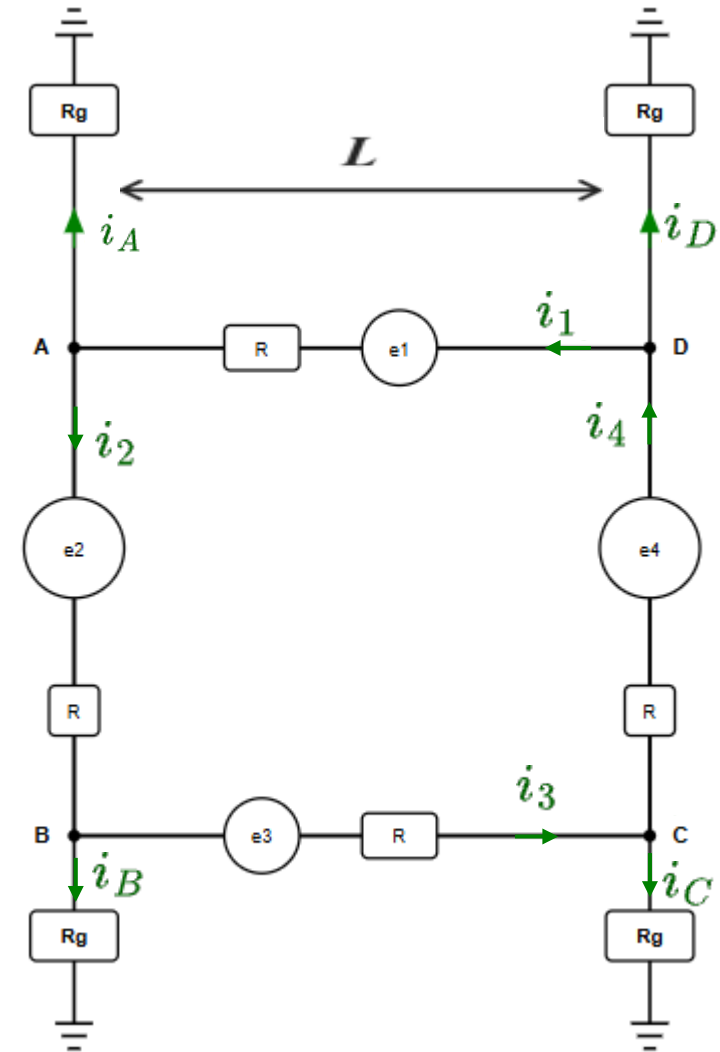
(b) Carte du réseau de transport électrique tunisien

Sources : *Tunisie Tribune*, article du 15/08/2023 , Carte de réseau fournit par Mr Hamdi BOUNAOURA ingénieur principal à la Centrale électrique de Radés

Modélisation électrique

□ Modélisation électrique :

- A, B, C, D : postes électriques
- $R_g = 0.5 \Omega$: résistance de la terre
- $R = 5 \Omega$: résistance des lignes électrique
- L : longueur des lignes électriques
- e_1, e_2, e_3, e_4 : f.é.m. d'induction
- i_A, i_B, i_C, i_D : courants entrants dans les postes électriques



Modélisation électrique

□ Expression des f.é.m. induites :

➤ Pour $k \in \{1, 2, 3, 4\}$ on note :

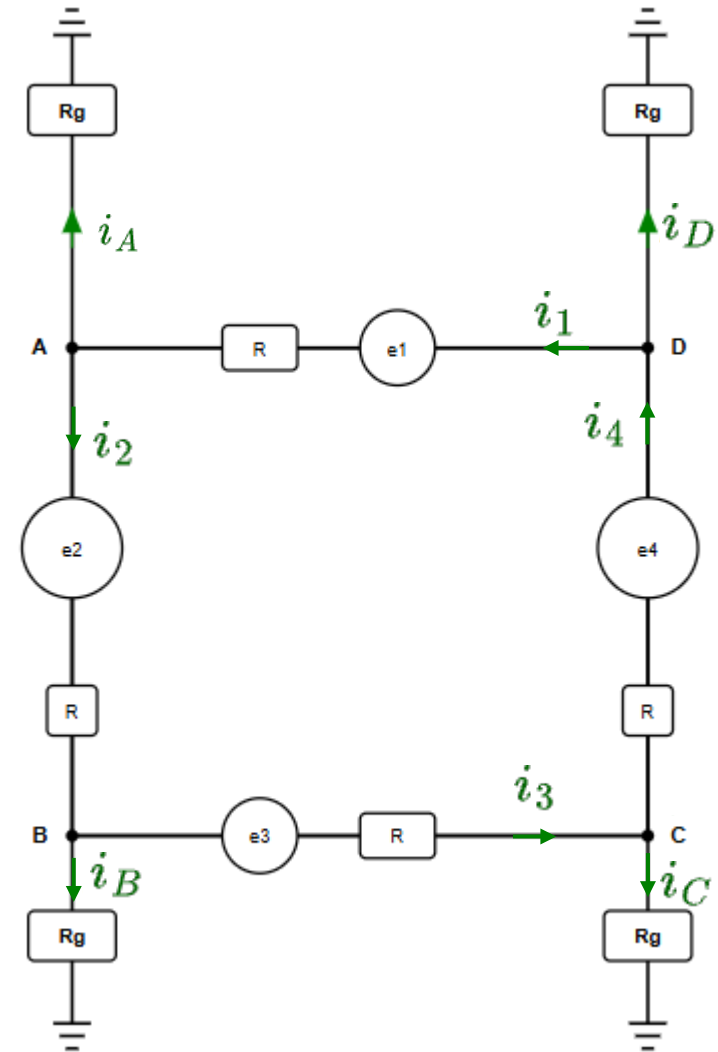
γ_k : Conductivité électrique moyenne du sol traversé par la ligne k

• δ_k : épaisseur de peau moyenne du sol traversé par la ligne k

$$e_k(t) = \frac{\omega \delta_k L}{\sqrt{2}} B_0 \cos(\omega t)$$

❖ Application des lois de Kirchhoff

$$\left\{ \begin{array}{l} i_A + i_2 - i_1 = 0 \\ i_B + i_3 - i_2 = 0 \\ i_C + i_4 - i_3 = 0 \\ i_D + i_1 - i_4 = 0 \\ i_C - i_B - \frac{R}{R_g} i_3 - \frac{e_3}{R_g} = 0 \\ i_D - i_C - \frac{R}{R_g} i_4 - \frac{e_4}{R_g} = 0 \\ i_A - i_D - \frac{R}{R_g} i_1 - \frac{e_1}{R_g} = 0 \\ i_B - i_A - \frac{R}{R_g} i_2 - \frac{e_2}{R_g} = 0 \end{array} \right.$$



➤ Résolution du système :

$$i_A = \frac{LB_0\sqrt{\frac{\omega}{\mu_0}}}{R^2 - 6RR_g + 8R_g^2} \left[\frac{-R + 3R_g}{\sqrt{\gamma_1}} + \frac{R - 3R_g}{\sqrt{\gamma_2}} - \frac{R_g}{\sqrt{\gamma_3}} + \frac{R_g}{\sqrt{\gamma_4}} \right] \cos(\omega t)$$

$$i_B = \frac{LB_0\sqrt{\frac{\omega}{\mu_0}}}{R^2 - 6RR_g + 8R_g^2} \left[\frac{R_g}{\sqrt{\gamma_1}} + \frac{-R + 3R_g}{\sqrt{\gamma_2}} + \frac{R - 3R_g}{\sqrt{\gamma_3}} - \frac{R_g}{\sqrt{\gamma_4}} \right] \cos(\omega t)$$

$$i_C = \frac{LB_0\sqrt{\frac{\omega}{\mu_0}}}{R^2 - 6RR_g + 8R_g^2} \left[-\frac{R_g}{\sqrt{\gamma_1}} + \frac{R_g}{\sqrt{\gamma_2}} + \frac{-R + 3R_g}{\sqrt{\gamma_3}} + \frac{R - 3R_g}{\sqrt{\gamma_4}} \right] \cos(\omega t)$$

$$i_D = \frac{LB_0\sqrt{\frac{\omega}{\mu_0}}}{R^2 - 6RR_g + 8R_g^2} \left[\frac{R - 3R_g}{\sqrt{\gamma_1}} - \frac{R_g}{\sqrt{\gamma_2}} + \frac{R_g}{\sqrt{\gamma_3}} + \frac{-R + 3R_g}{\sqrt{\gamma_4}} \right] \cos(\omega t)$$

On remarque que : $I_{GIC} \propto B_0 L$

Supposons que : $\gamma_1 = \gamma_2 = \gamma_3 = \gamma_4$ Alors, $i_A = i_B = i_C = i_D = 0$

Interprétation

Constat

La dissymétrie géologique des terrains traversés par les lignes à haute tension ainsi que la longueur des trajets favorisent l'apparition de courants induits géomagnétiquement.

Généralisation à une perturbation quelconque

- ❖ Les perturbations géomagnétiques réelles ne suivent pas un modèle harmonique

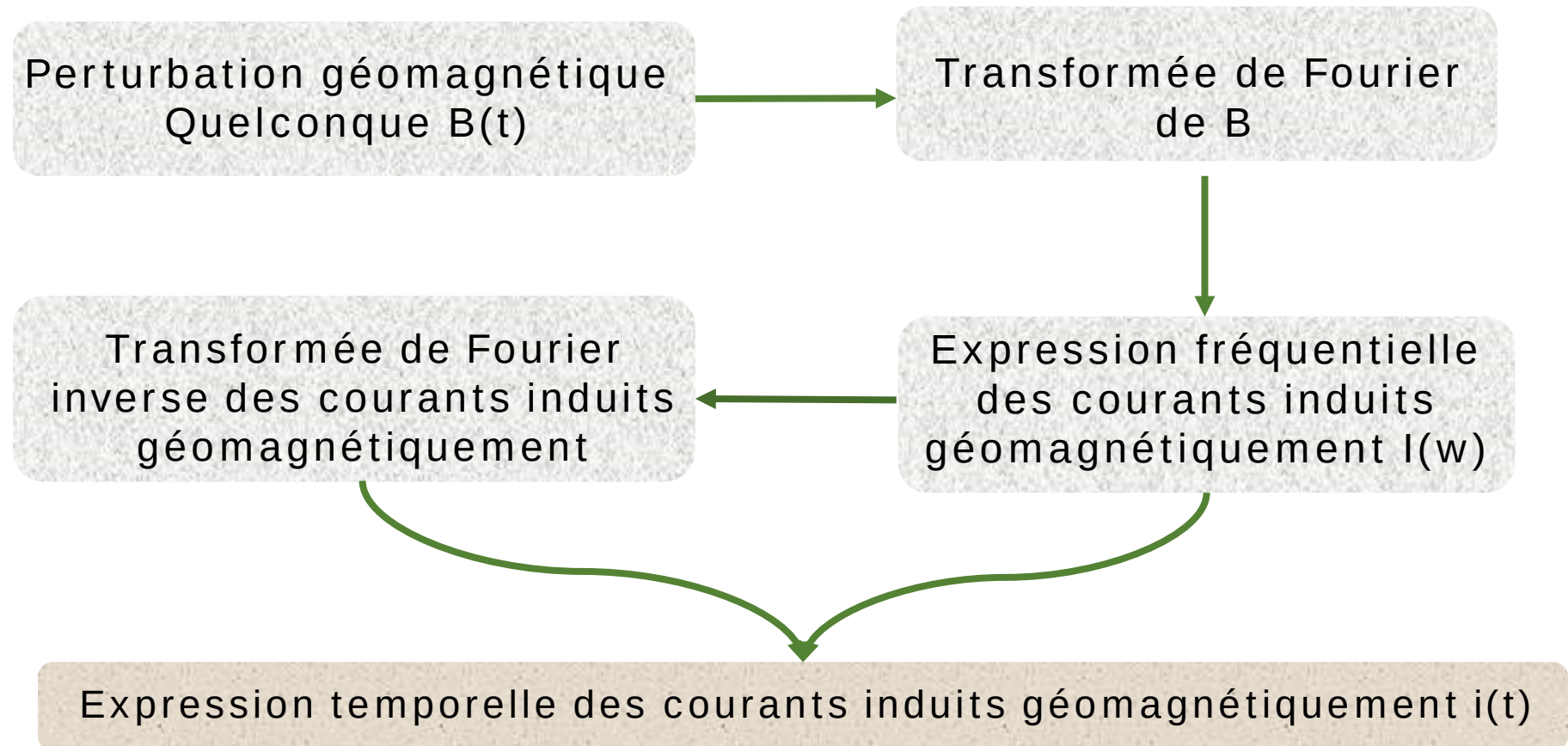
Transformée de Fourier et transformée de Fourier inverse

Pour f une fonction assez régulière, en particulier, C^1 par morceaux intégrable, dont la dérivée est localement intégrable, f s'exprime :

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(\omega) e^{j\omega t} d\omega \quad \text{avec} \quad \hat{f}(\omega) = \int_{-\infty}^{+\infty} f(t) e^{-j\omega t} dt$$

Généralisation à une perturbation quelconque

□ Démarche :



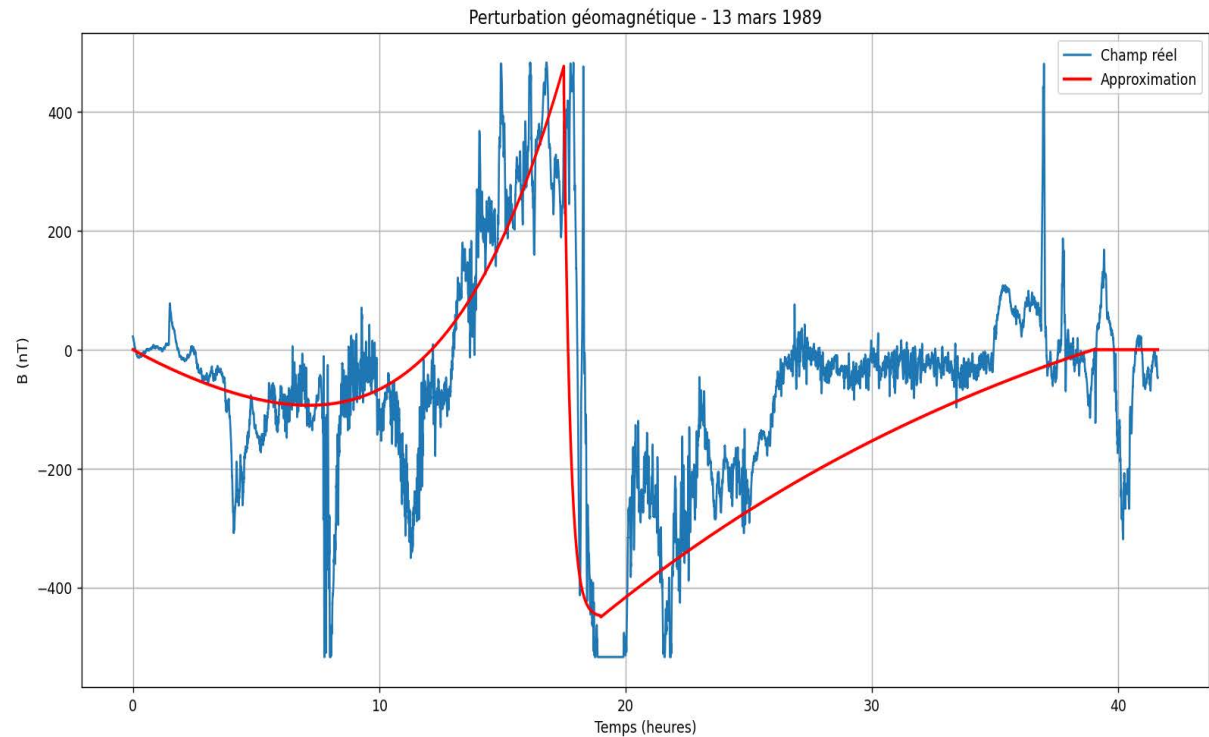
Application à des configurations réelles

Tempête géomagnétique du Québec (1989)

264.75	598.75	573.50
249.00	595.50	571.50
235.50	593.25	569.50
221.75	593.00	567.25
205.50	595.00	566.25
189.00	598.75	565.00
176.00	603.25	563.50
165.25	607.00	561.00
156.50	608.50	557.25
148.75	610.00	553.50
143.00	611.50	550.25
137.25	613.00	547.50
131.25	615.00	544.00
123.50	616.75	541.25
114.25	617.75	537.25
106.75	619.75	533.00
100.75	623.00	528.75
98.50	626.25	524.50

Ln 3, Col 24 830 785 caractères

Texte brut

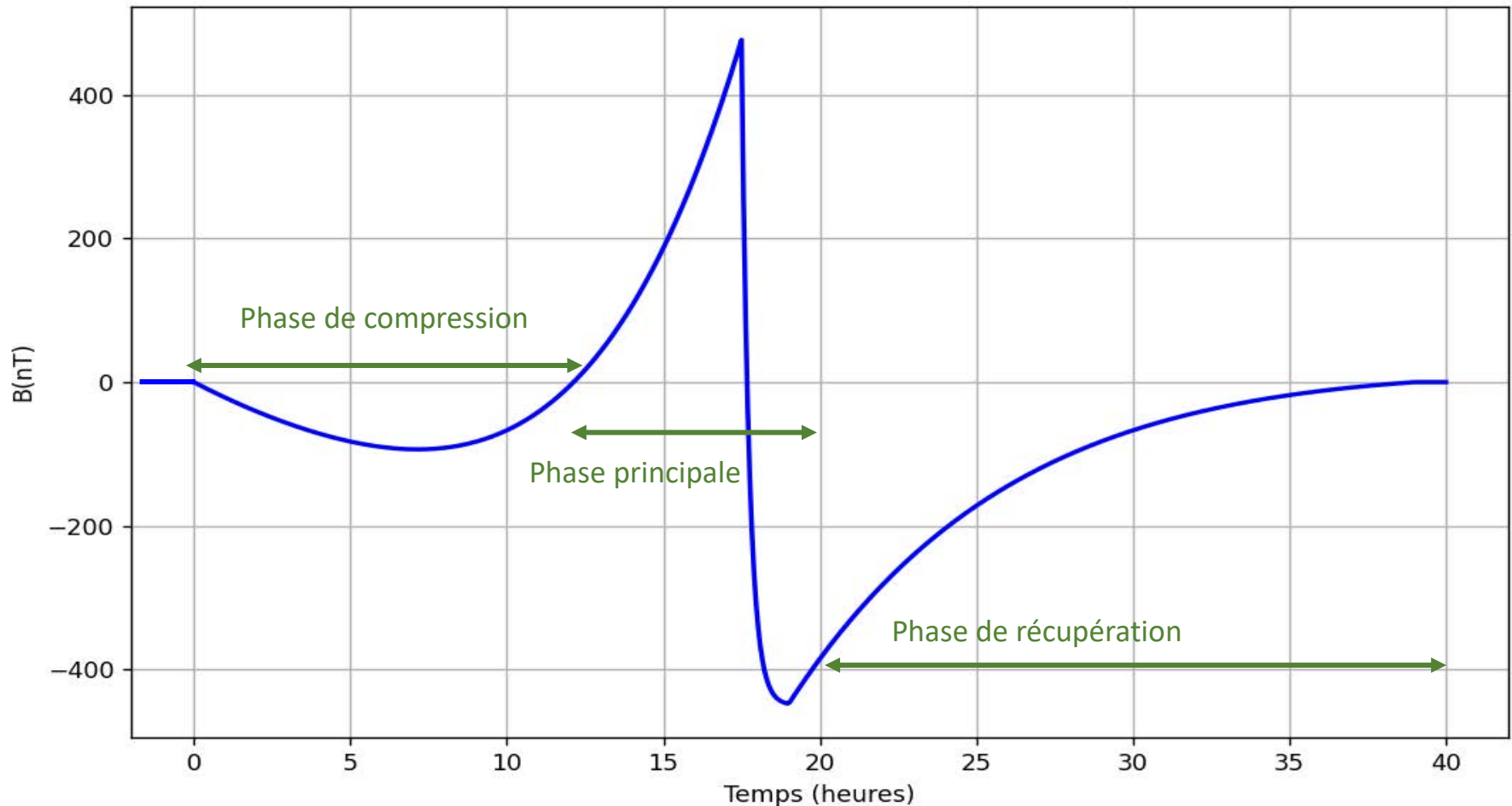


(a) Données sur le champ magnétique au sol le 13-14 mars 1989 avec un pas de 5s récupérées par l'Université de Lancaster

(b) Simulation numérique de l'une des composantes de la perturbation géomagnétique

Tempête géomagnétique du Québec (1989)

□ Allure approchée de la perturbation géomagnétique



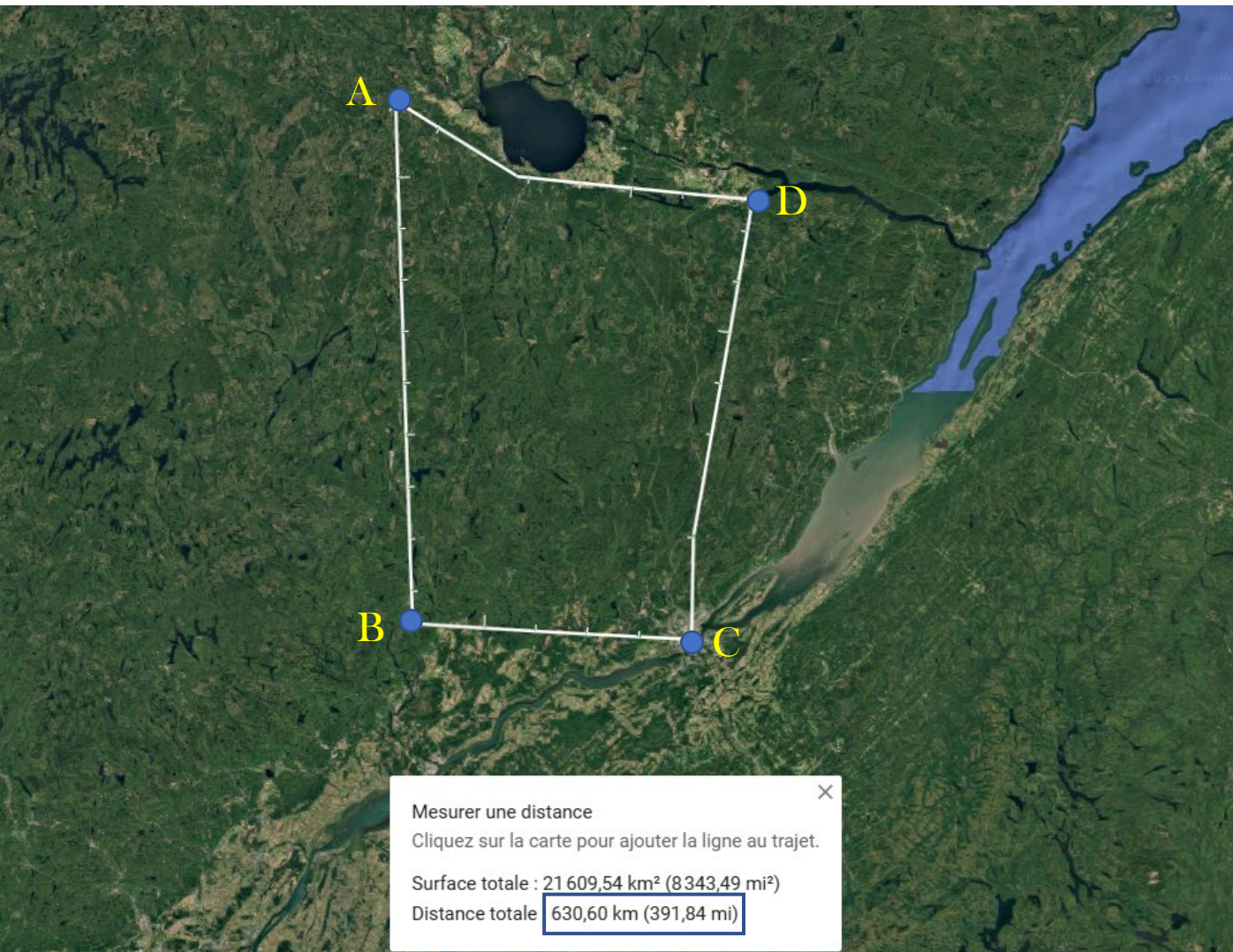
Réseau d'Hydro Québec



Carte du réseau d'Hydro Québec

Source : https://fr.wikipedia.org/wiki/Fichier:Quebec_Map_with_Hydro-Qu%C3%A9bec_infrastructures-fr.png

- Ajout d'une station fictive dans l'une des zones à risque (la station B) pour s'approcher du modèle théorique



$$L_{\text{total}} = 630 \text{ km}$$

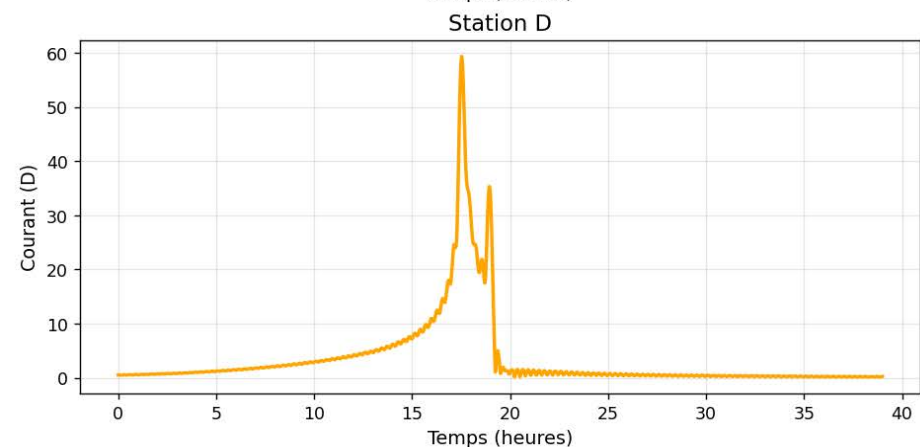
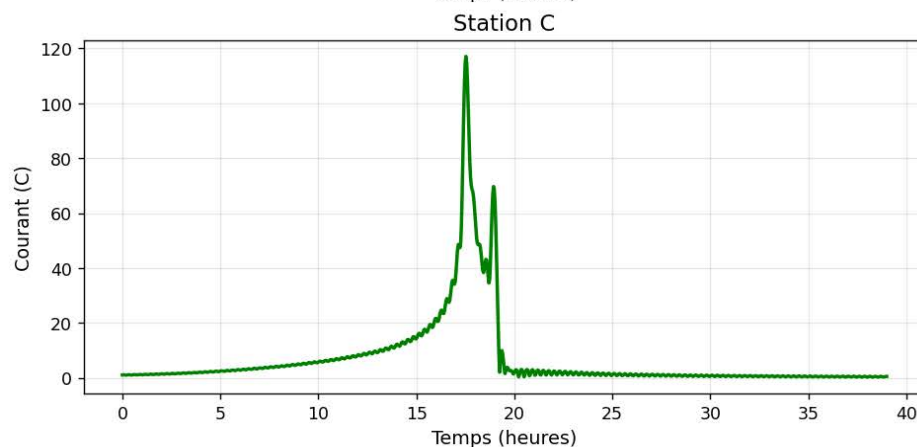
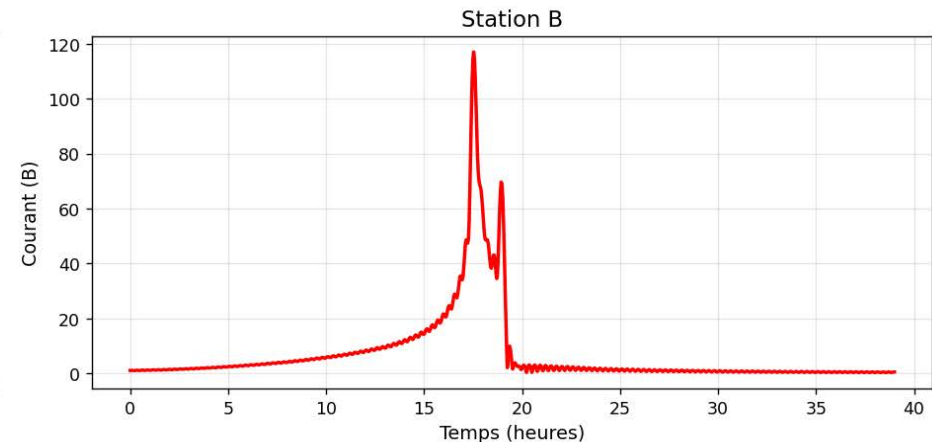
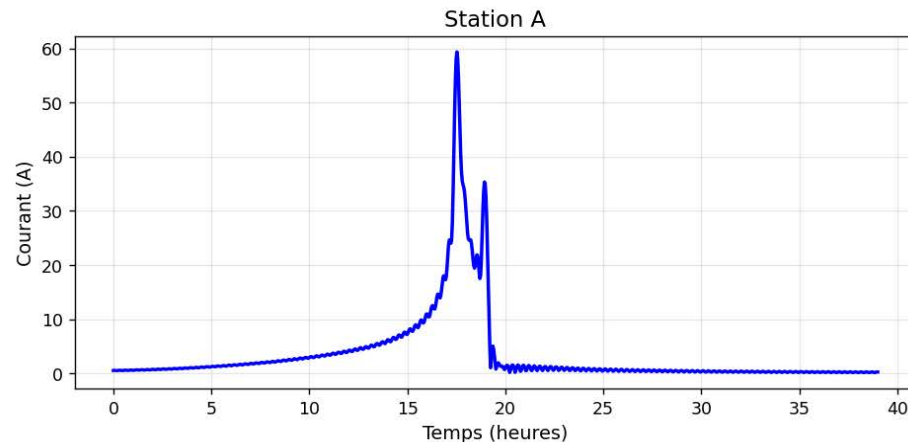
$$L = 157 \text{ km}$$

- ❑ Une étude de la géologie dominante de chaque ligne nous conduit à la détermination des conductivités électriques moyennes de chaque ligne

Lignes	Nature géologique dominante	Conductivité électrique
A-B	Socle cristallin	$\gamma_1 \approx 5 \times 10^{-5} \text{ S} \cdot \text{m}^{-1}$
B-C	Terrain hybride	$\gamma_2 \approx 10^{-4} \text{ S} \cdot \text{m}^{-1}$
C-D	Dépôts glaciaires et milieu humide	$\gamma_3 \approx 10^{-3} \text{ S} \cdot \text{m}^{-1}$
D-A	Socle rocheux cristallin forestier	$\gamma_4 \approx 5 \times 10^{-4} \text{ S} \cdot \text{m}^{-1}$

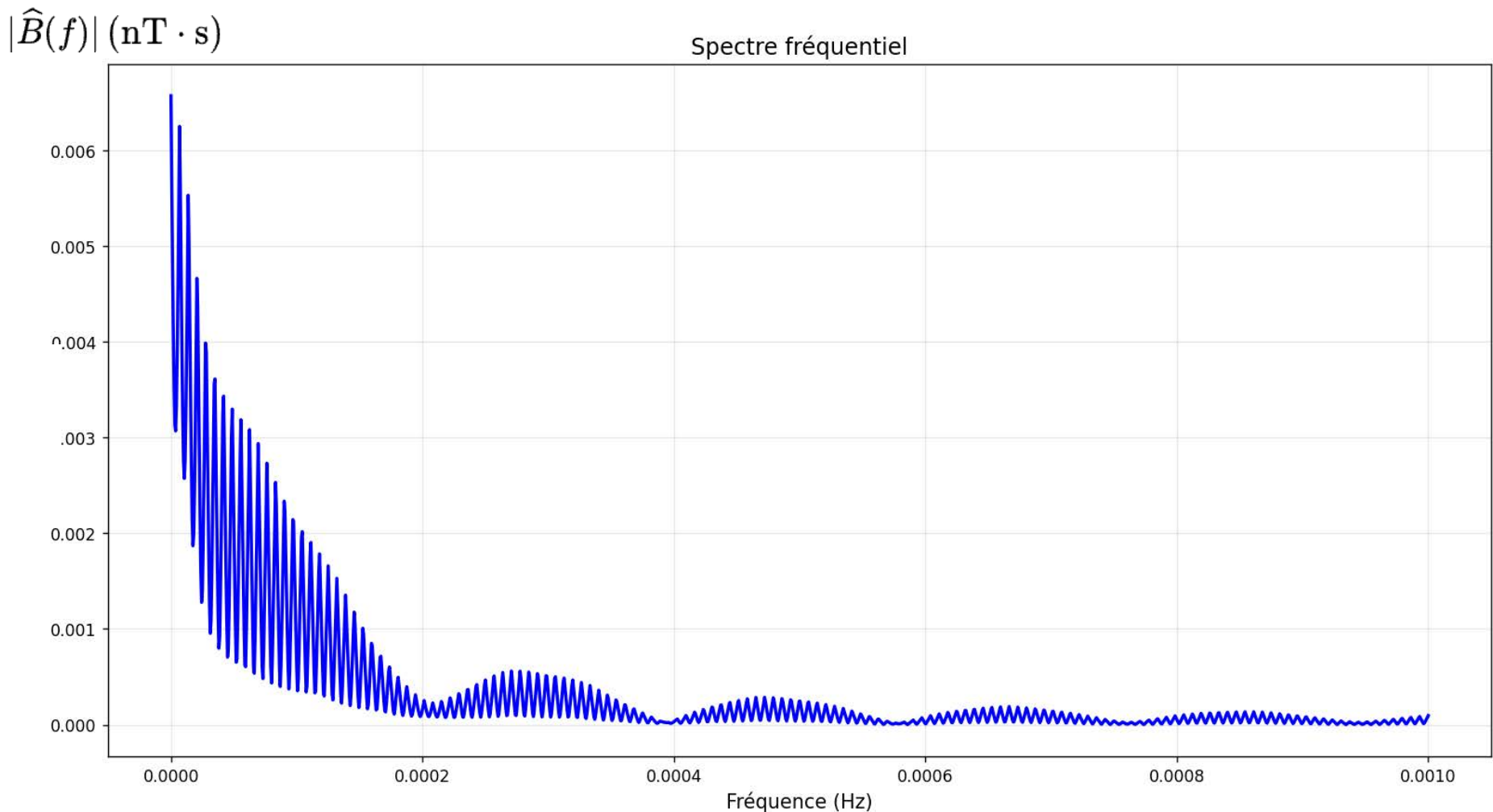
Détermination des courants induits géomagnétiquement

❑ Résultats numériques



➤ Des mesures réalisées par Hydro-Québec rapportent des intensités ayant atteint les 184 A

Détermination des courants induits géomagnétiquement



- les rapports de la Société Nord-Américaine de Fiabilité des Réseaux Electriques signalent la création d'un courant quasi-continu.

Sensibilité régionale

Visite de la centrale thermique de Radés



❖ Retour général à la suite de la visite :

1. La difficulté de détecter les courants quasi continus pénétrant les réseaux par les protections classiques
2. Les dangers liés à la circulation de courant quasi-continu dans les transformateurs
3. La faible exposition des pays loin des pôles magnétiques face aux effets des tempêtes solaires



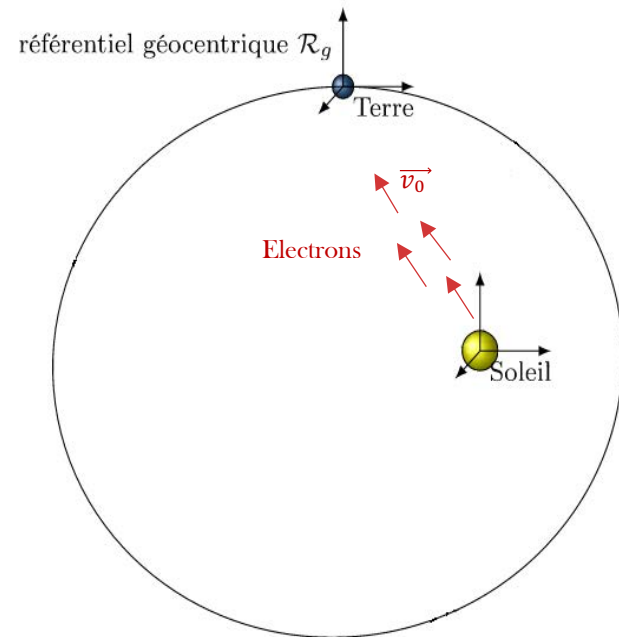
Poste de transformation capturé à la centrale thermique de Radés

Qu'est-ce qui rend certains pays plus sensibles
aux tempêtes solaires que d'autres ?

Retour sur l'origine des perturbations géomagnétiques

Hypothèses :

- On se place dans le référentiel géocentrique
- On étudie le mouvement d'un électron issu d'une éjection de masse coronale se dirigeant vers la Terre à une vitesse initiale $\vec{v}_0$
- On néglige les interactions entre l'électron étudié et les autres particules du vent solaire.
- On assimile le champ magnétique terrestre à un champ dipolaire stationnaire.



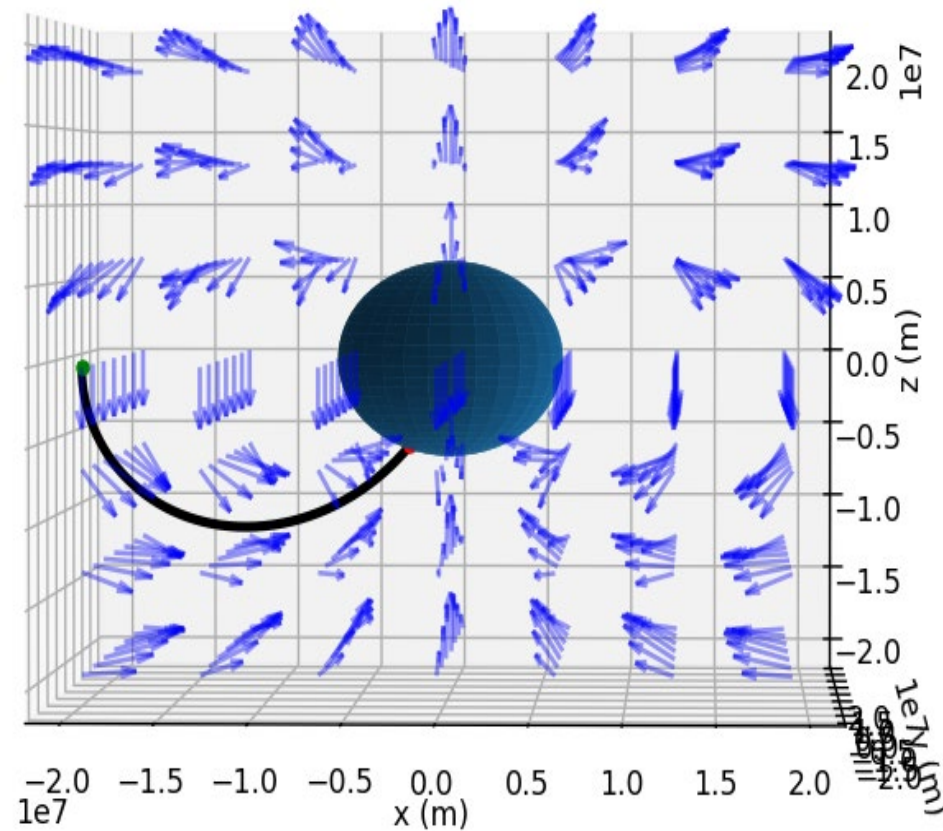
Source (adapté) : <https://femto-physique.fr/mecanique/referentiels-non-inertiels.php>

☐ Simulations de trajectoires (Sur une durée de 50s) :

- Trajectoire
- Départ
- Arrivée
- Lignes de champ

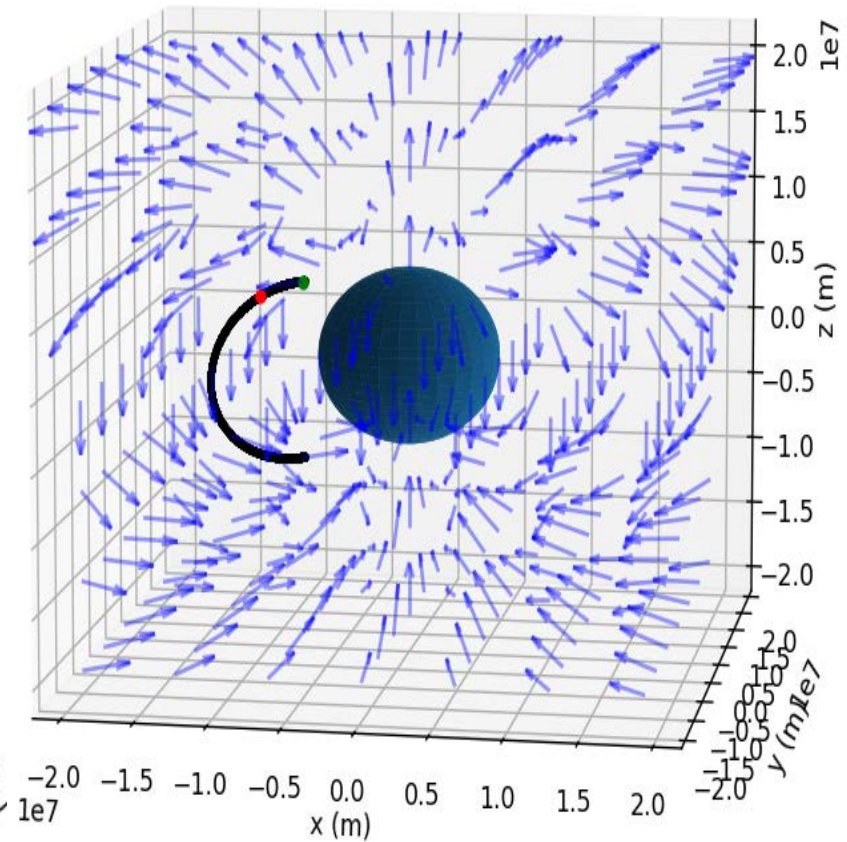
1^{er} cas

$\vec{v}_0$ positivement colinéaire à $\vec{B}$

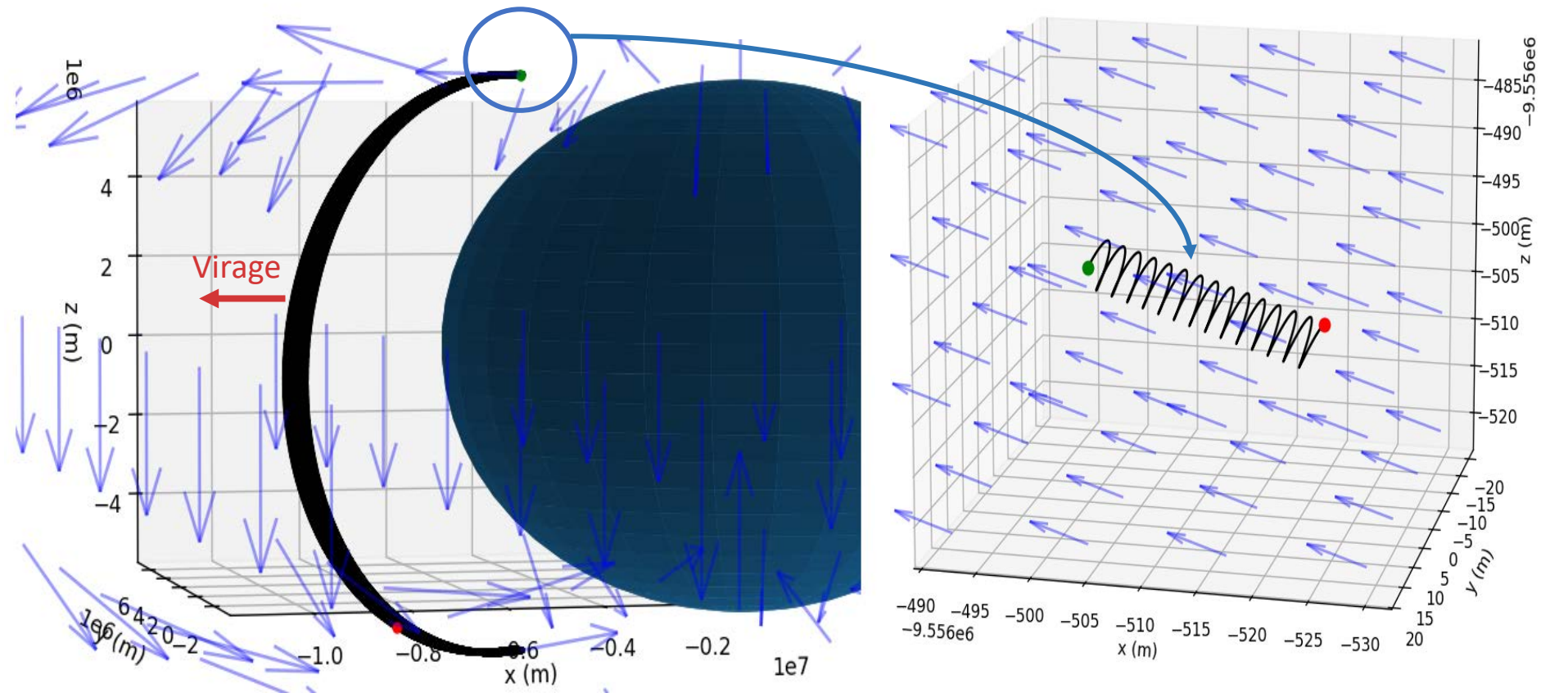


2^{eme} cas

$\vec{v}_0$ négativement colinéaire à $\vec{B}$



□ Nature du mouvement :



❑ Interprétation :

Mouvement hélicoïdal des électrons
autour des lignes de champ terrestre

Guidage des électrons vers les pôles
magnétiques terrestres

Concentration des électrons aux
pôles magnétiques

Création d'une perturbation magnétique
plus intense au voisinage des pôles

Conclusion



Annexe

□ Expression du champ crée dans la Terre :

$$\overrightarrow{\text{rot}} \underline{\vec{B}} = \mu_0 \vec{j} = \mu_0 \gamma \underline{\vec{E}} : \text{Maxwell Ampère}$$

$$\overrightarrow{\text{rot}} \underline{\vec{E}} = -\frac{\partial \underline{\vec{B}}}{\partial t} = -j\omega \underline{\vec{B}} : \text{Maxwell Faraday}$$

$$\overrightarrow{\text{rot}} \left(\overrightarrow{\text{rot}} \underline{\vec{B}} \right) = \overrightarrow{\text{grad}} \left(\text{div} \underline{\vec{B}} \right) - \Delta \underline{\vec{B}} \quad \text{Avec : } \text{div} \underline{\vec{B}} = 0 : \text{Maxwell Flux}$$

$$\Rightarrow \Delta \underline{\vec{B}} = j\omega \mu_0 \gamma \underline{\vec{B}}$$

□ Par des considérations de symétrie et par continuité du champ en $z = 0$

$$\underline{\vec{B}} = \underline{B}(z, t) \vec{e}_x \quad \forall z \leq 0$$

Annexe

$$\Rightarrow \frac{\partial^2 \underline{\vec{B}}}{\partial z^2} = j\omega\mu_0\gamma \underline{\vec{B}}$$

Les racines de l'équation caractéristique sont :

$$r = \pm \frac{1+j}{\delta} \quad \text{avec} \quad \delta = \sqrt{\frac{2}{\mu_0\omega\gamma}}$$

$$\underline{B}(z) = \underline{A}_1 e^{-\frac{(1+j)z}{\delta}} + \underline{A}_2 e^{\frac{(1+j)z}{\delta}} \quad \text{Si, } \underline{A}_1 \neq 0, \quad \lim_{z \rightarrow -\infty} \|\underline{\vec{B}}\| = +\infty$$

$$\text{Alors, } \underline{A}_2 = B_0 e^{j\omega t} \quad \text{et} \quad \underline{\vec{B}}(z, t) = B_0 e^{j\omega t} e^{(1+j)\frac{z}{\delta}} \vec{e}_x$$

Annexe

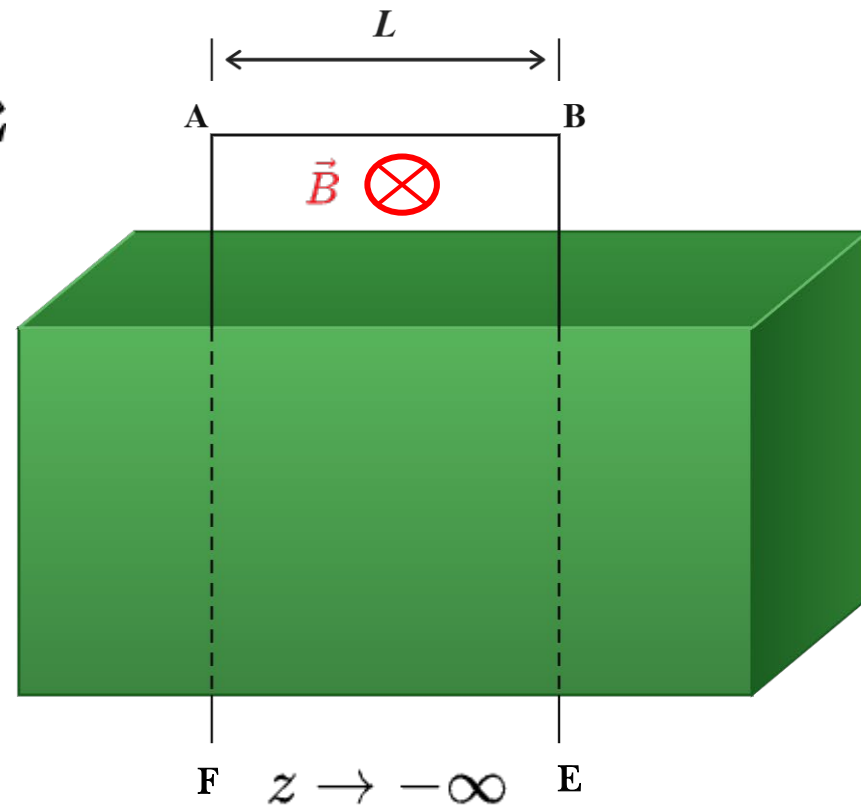
□ Détermination des f.é.m. induite :

$$e = -\frac{d\Phi}{dt} \quad \text{où} \quad \Phi = \iint_S \vec{B} \cdot d\vec{S}$$

On considère une ligne électrique de longueur L reliant deux postes électriques A et B

- On choisit la surface S , délimité par ABEF
- On réalise la fermeture à l'infinie en E et F. Plus formellement :

$$|z_E| \gg \delta \quad \text{et} \quad |z_F| \gg \delta$$



Annexe

$$\Phi = \iint_{ABEF} \vec{B} \cdot d\vec{S} \quad \text{avec} \quad d\vec{S} = dy dz \vec{e}_x$$

$$\Rightarrow \Phi = \int_{-\infty}^0 \int_0^L B_0 e^{j\omega t} e^{\frac{(1+j)z}{\delta}} dy dz \quad \text{avec} \quad \int_{-\infty}^0 e^{\frac{(1+j)z}{\delta}} dz = \frac{\delta}{1+j}$$

$$\Rightarrow \boxed{\Phi = \frac{B_0 L \delta}{1+j} e^{j\omega t}} \quad \Rightarrow \underline{e} = -\frac{d\Phi}{dt} = -\frac{j\omega B_0 L \delta}{1+j} e^{j\omega t}$$

$$\text{donc, } \underline{e} = \frac{\delta \omega L}{\sqrt{2}} B_0 e^{j(\omega t + \frac{\pi}{4})} \quad \text{On pose } t' = t + \frac{\pi}{4\omega}$$

On retrouve l'expression temporelle :

$$\boxed{e(t') = \frac{\omega \delta L}{\sqrt{2}} B_0 \cos(\omega t')}$$

Annexe

❖ Code python utilisé pour la résolution du système 8x8

resolution systeme 8x8.py ×

```
1 import sympy as sp
2
3 i1, i2, i3, i4, iA, iB, iC, iD, e1, e2, e3, e4, R, Rg = sp.symbols('i1,i2,i3,i4,iA,iB,iC,iD,e1,e2,e3,e4,R,Rg')
4
5 equations = [iA + i2 - i1 , iB + i3 - i2 , iC + i4 - i3 , iD + i1 - i4 , iC - iB - R/Rg * i3 - e3/Rg ,
6             iD - iC - R/Rg * i4 - e4/Rg , iA - iD - R/Rg * i1 - e1/Rg , iB - iA - R/Rg * i2 - e2/Rg]
7
8 solution = sp.solve(equations, [i1, i2, i3, i4, iA, iB, iC, iD])
9
10 for x in [i1, i2, i3, i4, iA, iB, iC, iD]:
11     y= sp.simplify(solution[x])
12     print(f"\n{x} = {y}")
```

Shell ×

```
iB = (-R*e2 + R*e3 + Rg*e1 + 3*Rg*e2 - 3*Rg*e3 - Rg*e4)/(R**2 - 6*R*Rg + 8*Rg**2)
iC = (-R*e3 + R*e4 - Rg*e1 + Rg*e2 + 3*Rg*e3 - 3*Rg*e4)/(R**2 - 6*R*Rg + 8*Rg**2)
iD = (R*e1 - R*e4 - 3*Rg*e1 - Rg*e2 + Rg*e3 + 3*Rg*e4)/(R**2 - 6*R*Rg + 8*Rg**2)
```

Annexe

- ❖ Code python utilisé pour le traçage de la perturbation magnétique de 1989 (1/2)

Porgramme.py ×

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 fichier = "gm13031989.5s"
5 Bx = []
6
7 with open(fichier, "r") as f:
8     lignes = f.readlines()
9     for ligne in lignes[1:]:
10         try:
11             valeurs = ligne.split()
12             bx = float(valeurs[0])
13             if abs(bx) > 1000:
14                 continue
15             Bx.append(bx)
16         except:
17             continue
18
19 Bx = np.array(Bx)
20 Bx0 = np.mean(Bx[:1000])
21 dBx = Bx - Bx0
22 t = np.arange(len(Bx)) * 5 / 3600
23
```

Annexe

- ❖ Code python utilisé pour le traçage de la perturbation magnétique de 1989 (2/2)

```
24 def approximation(t):
25
26     t1 = 17.5 * 3600
27     t2 = t1 + 1.5 * 3600
28     t3 = t2 + 20 * 3600
29
30     if t < 0 or t > t3:
31         return 0
32     if t <= t1:
33         return (-200 * (1 - np.exp(-2 * t / t1)) + 650 * (t / t1)**4)
34     elif t <= t2:
35         return 450 - 900 * (1 - np.exp(-6 * (t - t1) / (t2 - t1)))
36     else:
37         return -448 * (1 - (t - t2) / (t3 - t2)) * np.exp(-2 * (t - t2) / (t3 - t2))
38
39 T = np.arange(len(Bx)) * 5
40 B_approx = [approximation(ti) for ti in T]
41 print(Bx0)
42
43 plt.figure(figsize=(12,6))
44 plt.plot(t, dBx, label="Champ réel")
45 plt.plot(t, B_approx, linewidth=2, color='red', label="Approximation")
46 plt.xlabel("Temps (heures)")
47 plt.ylabel("Bx (nT)")
48 plt.title("Perturbation géomagnétique - 13 mars 1989")
49 plt.grid()
50 plt.legend()
51 plt.show()
```

Annexe

□ Allure approchée la perturbation géomagnétique

$$B(t) = \begin{cases} -200(1 - e^{-2t/17.5}) + 650(t/17.5)^4 & : 0 \leq t \leq 17.5 h \\ 450 - 900(1 - e^{-6(t-17.5)/1.5}) & : 17.5 h < t < 19h \\ -448(1 - (t-19)/20)e^{-2(t-19)/20} & : 19 h < t \leq 39 h \\ 0 & : t < 0, t > 39 h \end{cases}$$

En (nT)

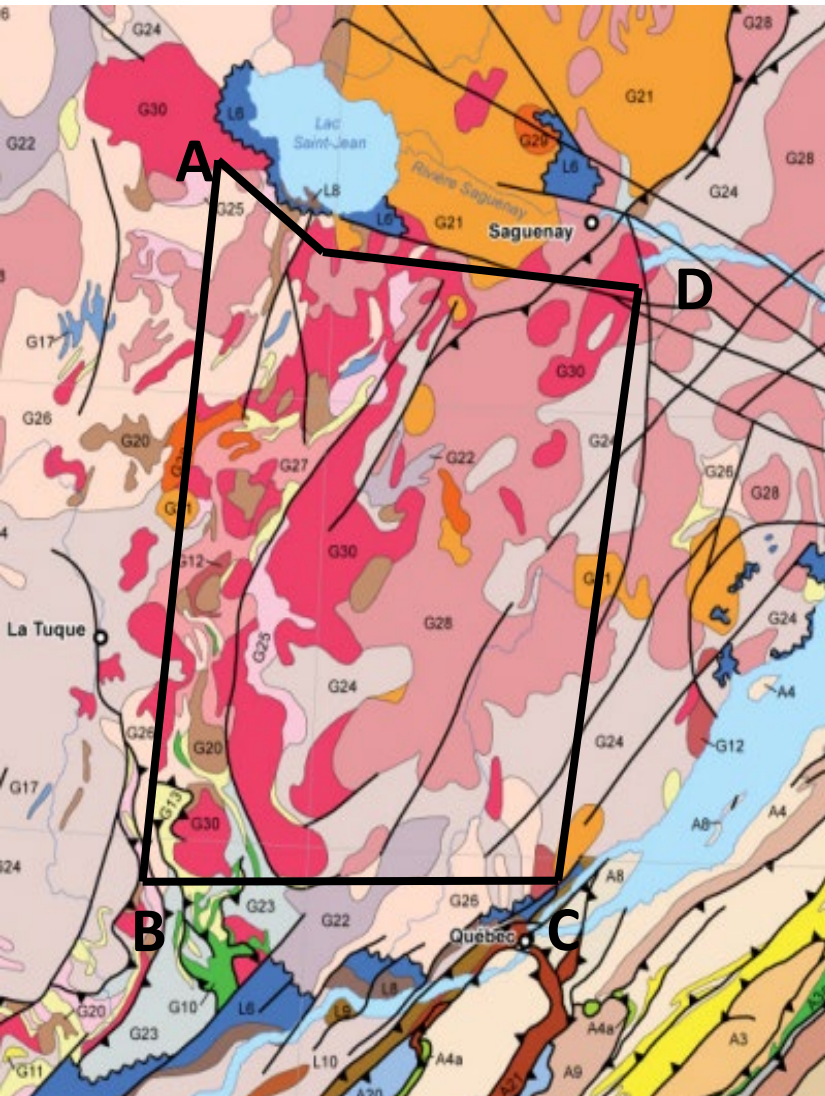
Annexe

□ Allure approchée la perturbation géomagnétique (simulation)

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 def B(t):
5
6     t1 = 17.5 * 3600
7     t2 = t1 + 1.5 * 3600
8     t3 = t2 + 20 * 3600
9
10    if t < 0 or t > t3:
11        return 0
12    if t <= t1:
13        return (-200 * (1 - np.exp(-2 * t / t1)) + 650 * (t / t1)**4)
14    elif t <= t2:
15        return 450 - 900 * (1 - np.exp(-6 * (t - t1) / (t2 - t1)))
16    else:
17        return -448 * (1 - (t - t2) / (t3 - t2)) * np.exp(-2 * (t - t2) / (t3 - t2))
18
19 t = np.linspace(0, 40*3600, 600)
20 dst = [B(ti) for ti in t]
21
22 plt.figure(figsize=(10, 6))
23 plt.plot(t/3600, dst, linewidth=2, color='blue')
24
25 plt.xlabel('Temps (heures)')
26 plt.ylabel('B(nT)')
27 plt.title('Tempete du Québec')
28 plt.grid(True)
29 plt.show()
```

Annexe

□ Nature géologique :



GRANITOÏDES

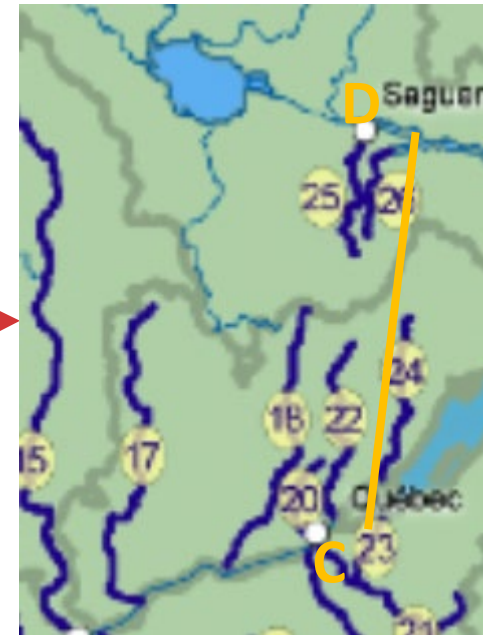
G30	Granite, quartz monzonite, quartz syénite et pegmatite
G29	Syénite, monzonite, monzodiorite et autres intrusions alcalines associées
G28	Granitoïdes ortho-pyroxène : charnockite, mangerite, jotunite et syénite hypersthène
G27	Granitoïdes non subdivisés
G26	Migmatite
G25	Gneiss granitoïde
G24	Gneiss charnockitique et gneiss mixte
G23	Gneiss tonalitique et trondhjémitique; présence commune de tonalite
G22	Gneiss non subdivisé

→ Roches cristallines dominantes sur le trajet A-B (G30,G26,G24) et assez présentes dans D-A (G30,G26) et B-C (mixte) et C-D (G24,G28)

<https://gq.mines.gouv.qc.ca/documentation/informations-complementaires/carte-geologique-du-quebec/>

Annexe

Carte hydrologique du sud du Québec

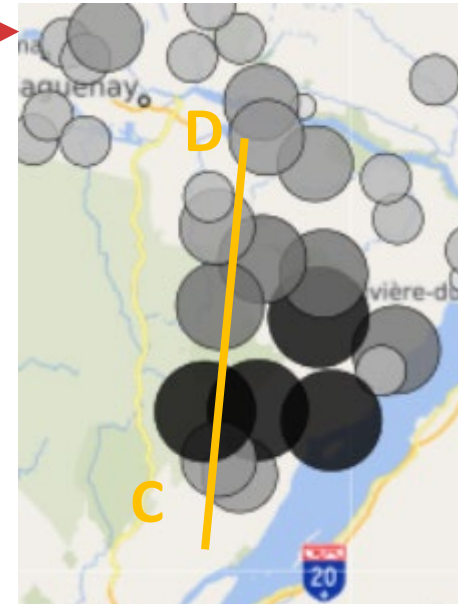
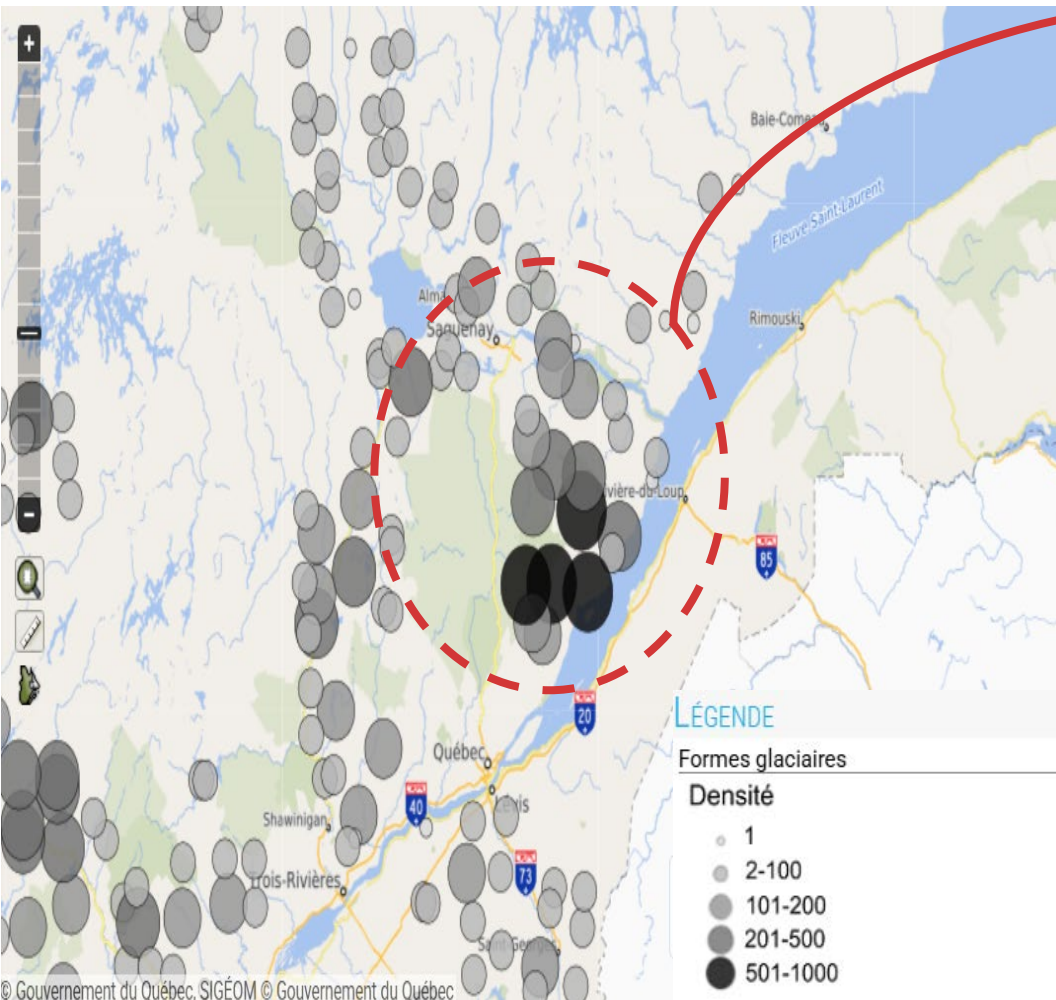


C-D traverse une zone
→ hydrologiquement dense donc
une zone « humide »

<https://journals.openedition.org/developpementdurable/docannexe/image/2762/img-3-small580.jpg>

Annexe

Carte des formes glaciaires



La ligne C-D Traverse une zone dominée par des dépôts glaciaires

Annexe

Lignes	Nature géologique dominante	Matériaux associés	Conductivité électrique moyenne retenue
A-B	Socle cristallin	Granite, gneiss	$\gamma_1 \approx 5 \times 10^{-5} \text{ S/m}$
B-C	Terrain hybride	Granite, gneiss avec présence de schistes	$\gamma_2 \approx 10^{-4} \text{ S/m}$
C-D	Dépôts glaciaires et milieu humide	Till glaciaire humide Argiles saturés en eau	$\gamma_3 \approx 10^{-3} \text{ S/m}$
D-A	Socle rocheux cristallin forestier	Granite et gneiss fracturés	$\gamma_4 \approx 5 \times 10^{-4} \text{ S/m}$

Matériau	Conductivité électrique (S/m)
Granite sec	$10^{-6} \text{ à } 10^{-4}$
Gneiss	$10^{-5} \text{ à } 10^{-3}$
Schistes	$10^{-4} \text{ à } 10^{-1}$
Gneiss fracturé	$10^{-4} \text{ à } 10^{-2}$
Granite fracturé	$10^{-5} \text{ à } 10^{-3}$
Eau douce	$10^{-2} \text{ à } 10^{-1}$
Argile humide	$10^{-2} \text{ à } 1$
Till glaciaire humide	$10^{-4} \text{ à } 10^{-3}$

- L'humidité favorise l'augmentation de la conductivité électrique
- La fracturation favorise l'augmentation de la conductivité électrique

Sources : https://www.researchgate.net/figure/Electrical-resistivity-ranges-for-several-rocks-sediments-and-soils-modified-from_fig2_323586689

Telford et al., *Applied Geophysics*, Cambridge University Press, 1990.

Annexe

❖ Code python utilisé pour la transformée de Fourier (1/5)

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy.interpolate import interp1d
4
5 def B(t):
6     t1 = 17.5 * 3600
7     t2 = t1 + 1.5 * 3600
8     t3 = t2 + 20 * 3600
9
10    if t < 0 or t > t3:
11        return 0
12    if t <= t1:
13        return (-200 * 1e-9 * (1 - np.exp(-2 * t / t1)) + 650 * (t / t1)**4) * 1e-9
14    elif t <= t2:
15        return 450 * 1e-9 - 900 * (1 - np.exp(-6 * (t - t1) / (t2 - t1))) * 1e-9
16    else:
17        return -448 * 1e-9 * (1 - (t - t2) / (t3 - t2)) * np.exp(-2 * (t - t2) / (t3 - t2)) * 1e-9
18
19    T0 = 40 * 3600
20    f1 = 0
21    f2 = 1e-3
22    w1 = np.pi * 2 * f1
23    w2 = np.pi * 2 * f2
24    L, R, Rg, mu_0, gam1, gam2, gam3, gam4 = 1.57e5, 5, 0.5, 4 * np.pi * 1e-7, 5e-5, 1e-4, 1e-3, 5e-4
25    D = (R - 2 * Rg) * (R - 4 * Rg)
26    cste1 = L / (D * np.sqrt(mu_0)) * ((3 * Rg - R) / np.sqrt(gam1) + (R - 3 * Rg) / np.sqrt(gam2) - Rg / np.sqrt(gam3) + Rg / np.sqrt(gam4))
27    cste2 = L / (D * np.sqrt(mu_0)) * (Rg / np.sqrt(gam1) + (3 * Rg - R) / np.sqrt(gam2) + (R - 3 * Rg) / np.sqrt(gam3) - Rg / np.sqrt(gam4))
28    cste3 = L / (D * np.sqrt(mu_0)) * (-Rg / np.sqrt(gam1) + Rg / np.sqrt(gam2) + (3 * Rg - R) / np.sqrt(gam3) + (R - 3 * Rg) / np.sqrt(gam4))
29    cste4 = L / (D * np.sqrt(mu_0)) * ((R - 3 * Rg) / np.sqrt(gam1) - Rg / np.sqrt(gam2) + Rg / np.sqrt(gam3) + (3 * Rg - R) / np.sqrt(gam4))
30
31    N = 500000
32    temps = np.linspace(0, T0, N)
33    dt = T0 / N
34    signal = np.array([B(t) for t in temps])
35    G = np.fft.fft(signal) * dt
36    w = 2 * np.pi * np.fft.fftfreq(N, dt)
37
38    sort_idx = np.argsort(w)
39    interp_reel = interp1d(w[sort_idx], np.real(G[sort_idx]), kind='linear')
40    interp_imag = interp1d(w[sort_idx], np.imag(G[sort_idx]), kind='linear')
```

Annexe

❖ Code python utilisé pour la transformée de Fourier (2/5)

```
42 def poids_frequence(w):
43     return interp_reel(w) + 1j * interp_imag(w)
44
45 def courant(t,cste):
46     w_liste = np.linspace(w1, w2, 10000)
47     dw = w_liste[1] - w_liste[0]
48     G_values = np.array([poids_frequence(w) for w in w_liste])
49     integral = np.sum(G_values * cste * np.sqrt(w_liste) * np.exp(1j * w_liste * t)) * dw
50     return abs(integral)
51
```

Annexe

❖ Code python utilisé pour la transformée de Fourier (3/5)

```
52 temp = np.linspace(0, 39*3600, 1000)
53 I1 = [courant(t,cste1) for t in temp]
54 I2 = [courant(t,cste2) for t in temp]
55 I3 = [courant(t,cste3) for t in temp]
56 I4 = [courant(t,cste4) for t in temp]
57
58 plt.figure(figsize=(14, 8))
59 plt.subplot(2, 2, 1)
60 plt.plot(temp/3600, I1, color="blue", linewidth=2)
61 plt.xlabel('Temps (heures)', fontsize=11)
62 plt.ylabel('Courant (A)', fontsize=11)
63 plt.title('Station A', fontsize=13)
64 plt.grid(True, alpha=0.3)
65
```

Annexe

❖ Code python utilisé pour la transformée de Fourier (4/5)

```
66 plt.subplot(2, 2, 2)
67 plt.plot(temp/3600, I2, color="red", linewidth=2)
68 plt.xlabel('Temps (heures)', fontsize=11)
69 plt.ylabel('Courant (B)', fontsize=11)
70 plt.title('Station B', fontsize=13)
71 plt.grid(True, alpha=0.3)
72
73 plt.subplot(2, 2, 3)
74 plt.plot(temp/3600, I3, color="green", linewidth=2)
75 plt.xlabel('Temps (heures)', fontsize=11)
76 plt.ylabel('Courant (C)', fontsize=11)
77 plt.title('Station C', fontsize=13)
78 plt.grid(True, alpha=0.3)
79
80 plt.subplot(2, 2, 4)
81 plt.plot(temp/3600, I4, color="orange", linewidth=2)
82 plt.xlabel('Temps (heures)', fontsize=11)
83 plt.ylabel('Courant (D)', fontsize=11)
84 plt.title('Station D', fontsize=13)
85 plt.grid(True, alpha=0.3)
86
```

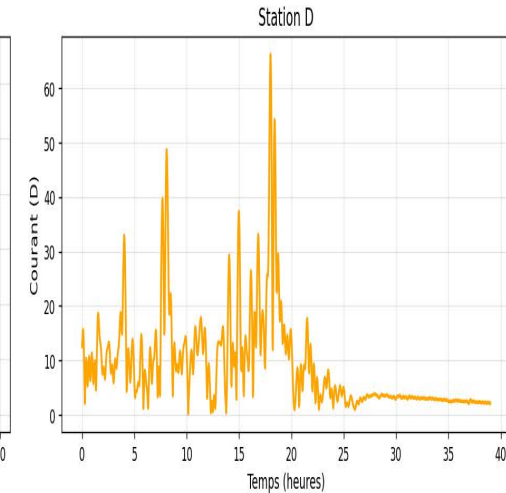
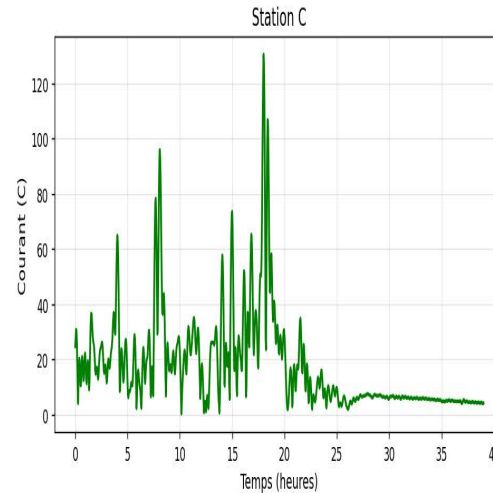
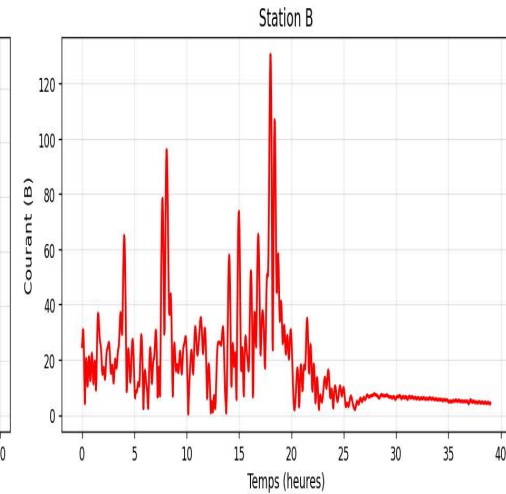
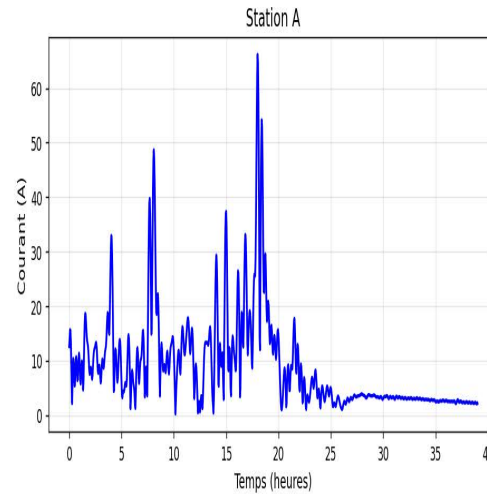
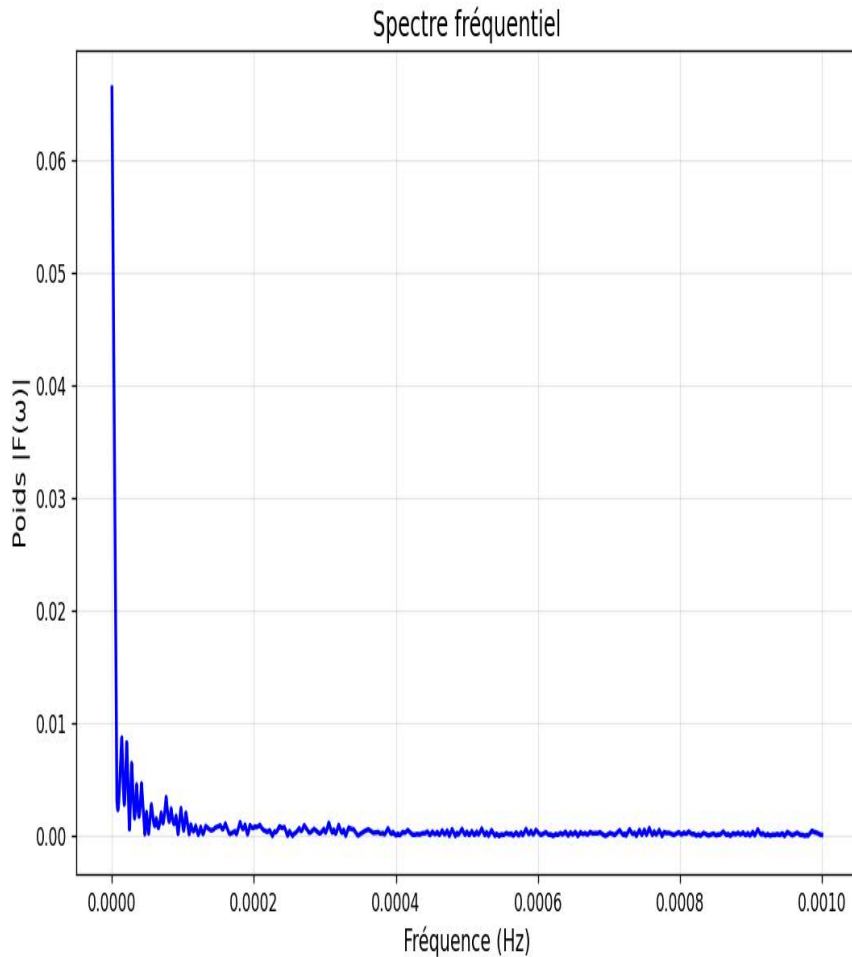
Annexe

❖ Code python utilisé pour la transformée de Fourier (5/5)

```
87 plt.tight_layout()
88 plt.show()
89
90 frequencies = np.linspace(f1, f2, 1000)
91 L_poids = []
92 for f in frequencies:
93     w0 = 2 * np.pi * f
94     p = abs(poids_frequence(w0))
95     L_poids.append(p)
96
97 plt.figure(figsize=(10, 6))
98 plt.plot(frequencies, L_poids, color="blue", linewidth=2)
99 plt.xlabel('Fréquence (Hz)', fontsize=12)
100 plt.ylabel('Poids |F(ω)|', fontsize=12)
101 plt.title('Spectre fréquentiel', fontsize=14)
102 plt.grid(True, alpha=0.3)
103 #plt.yscale('log')
104 plt.show()
105
```

Annexe

□ Traçage à l'aide de la perturbation réelle :



Annexe

❖ Visite de la centrale thermique de Radés :

Les personnes contactées principalement :

Mr Hamdi BOUNAOURA – direction centrale de la production d'électricité

Mr Wissam HENI - direction centrale réseau de transport électrique

Mr Wadi MOUSSA - direction centrale de la production d'électricité



Turbine à vapeur



Conduites d'amenée d'eau de mer

Annexe

□ Saturation d'un transformateur :

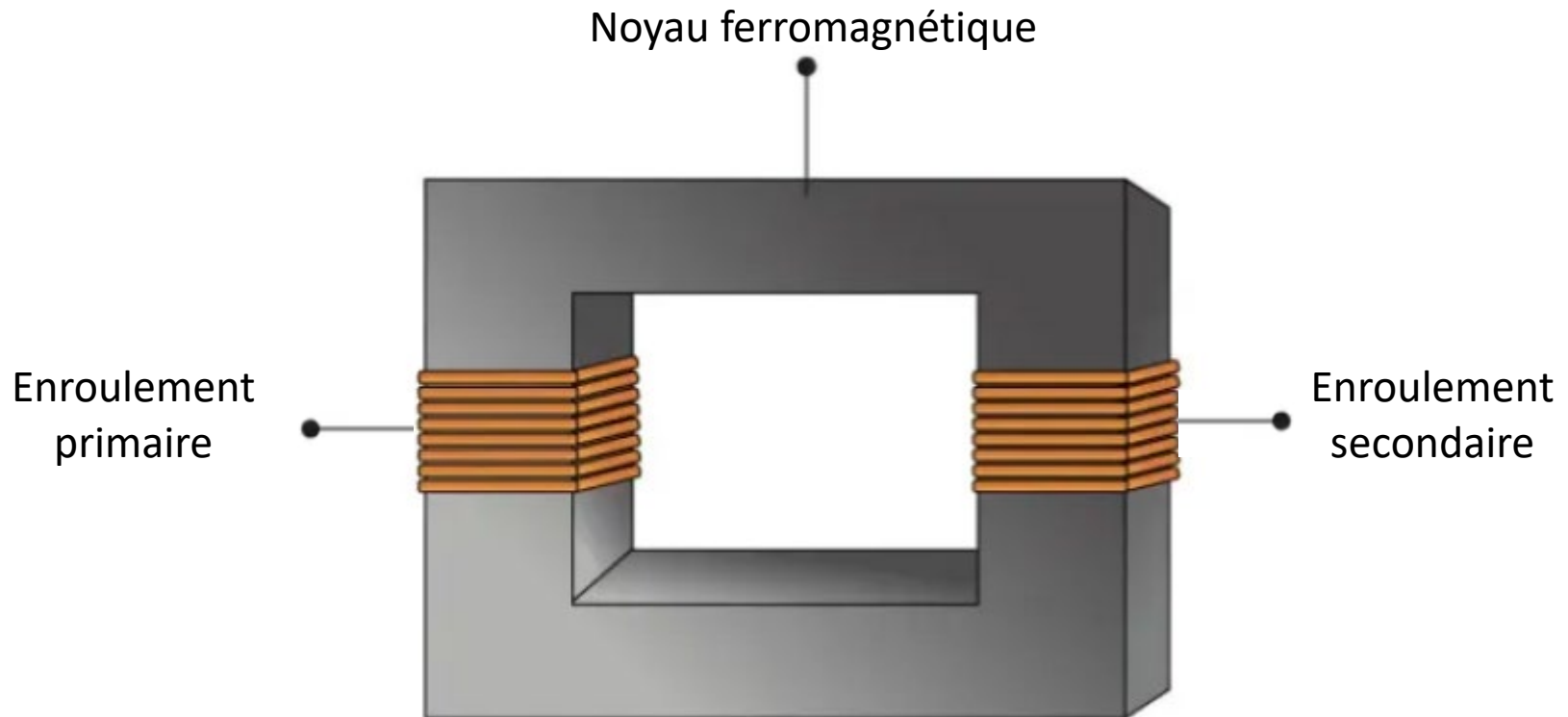


Schéma simplifié d'un transformateur de puissance

Annexe

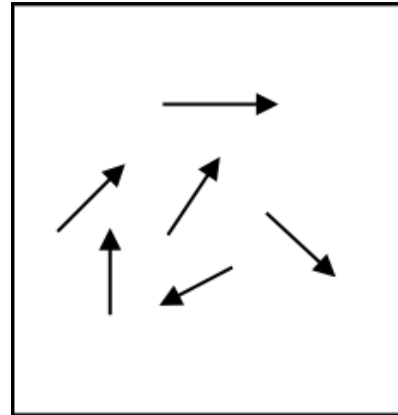
Le courant circulant dans l'enroulement des fils crée un champ magnétique qui est amplifié à l'aide du noyau



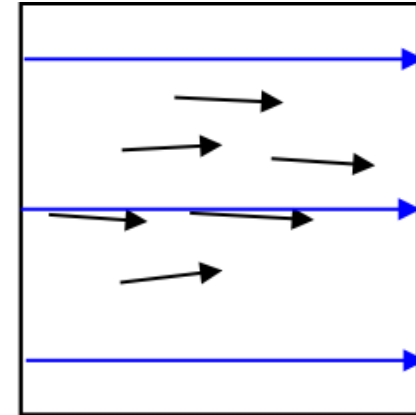
Si le champ magnétique est important $\|\vec{B}\| > B_{\text{sat}}$ tous les sous dipôles sont orientés dans la direction du champ, on a atteint la saturation.



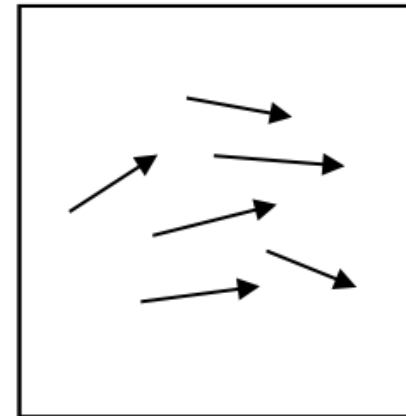
Le champ magnétique créé par le matériau ferromagnétique devient presque constant à la saturation



Milieu en l'absence de champ magnétique extérieur



Milieu soumis à un champ magnétique extérieur



On coupe le champ magnétique

❖ Le danger :

- La norme du champ créé par le noyau avant la saturation est de l'ordre de 100 à 1000 fois plus grande que celle par induction
- À la saturation , le noyau ferromagnétique ne contribue plus au flux , or le réseau impose sa tension (par exemple 735 kV en amplitude) qui entraîne une forte augmentation du courant primaire pour compenser la diminution de la contribution du noyau

Forte augmentation
du courant



Echauffement important
des enroulements

Annexe

❖ Trajectoire d'un électron :

Bilan des forces extérieures : $\vec{P} = m_e \vec{g}$ et $\vec{F}_L = -e \vec{v} \wedge \vec{B}$

$m_e \simeq 9,1 \times 10^{-31} \text{ kg}$: masse de l'électron

$e \simeq 1,6 \times 10^{-19} \text{ C}$: charge élémentaire

$g \simeq 9,8 \text{ m} \cdot \text{s}^{-2}$: intensité de la pesanteur

$B \simeq 5 \times 10^{-5} \text{ T}$: champ magnétique terrestre

$v \in [2,5 \times 10^5 ; 3 \times 10^6] \text{ m} \cdot \text{s}^{-1}$: vitesse d'un électron issu d'une tempête solaire

$$\frac{\|\vec{P}\|}{\|\vec{F}_L\|} \leq \frac{m_e g}{e v_{\min} B} \simeq 4,5 \times 10^{-12} \ll 1$$

On suppose donc que l'électron subit uniquement la force de Lorentz $\vec{F}_L$.

Annexe

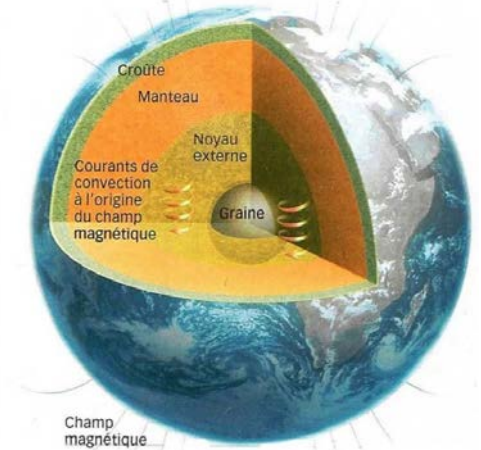
Le champ magnétique terrestre provient de la circulation de métaux en fusion dans le noyau terrestre.

On note $a \simeq 3,5 \times 10^6$ m le rayon du noyau terrestre.

La condition $r \gg a$ est vérifiée sur une grande partie du trajet.

On peut donc utiliser le modèle dipolaire du champ magnétique terrestre :

$$\vec{B} = \frac{\mu_0}{4\pi} \left(\frac{3(\vec{M} \cdot \vec{r})\vec{r}}{r^5} - \frac{\vec{M}}{r^3} \right)$$



$\vec{M} = M \vec{e}_z$: moment magnétique terrestre $M \approx 8 \times 10^{22}$ A · m²

$\vec{e}_z$: pointe du pôle Sud magnétique vers le pôle Nord magnétique de la Terre.

$\vec{r}$: vecteur position de l'électron de norme r

<https://champmagnetique.canalblog.com/pages/a--son-origine/31650881.html>

Annexe

- ❖ Code python utilisé pour la trajectoire de l'électron (1/4) :

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from mpl_toolkits.mplot3d import Axes3D
4
5 mu_0 = 4 * np.pi * 1e-7
6 R_T = 6.371e6
7 M = 7.94e22
8 M_vec = np.array([0.0, 0.0, M])
9
10 def champ(r_vec):
11     r = np.linalg.norm(r_vec)
12     if r < R_T:
13         return np.array([0.0, 0.0, 0.0])
14     B = (mu_0 / (4 * np.pi)) * (3 * np.dot(M_vec, r_vec) * r_vec - r**2 * M_vec) / r**5
15     return B
```

Annexe

❖ Code python utilisé pour la trajectoire de l'électron (2/4) :

```
17 def tracage(r0, v0, q, m, dt, n):
18     r = r0.copy()
19     v = v0.copy()
20     positions = np.zeros((n, 3))
21     for i in range(n):
22         if i % (n//100) == 0:
23             print(f'Progression: {i/n * 100:.1f}%')
24             F1 = q * np.cross(v, champ(r))
25             a1 = F1 / m
26             rk1, vk1 = v * dt, a1 * dt
27             r2 = r + 0.5 * rk1
28             v2 = v + 0.5 * vk1
29             F2 = q * np.cross(v2, champ(r2))
30             a2 = F2 / m
31             rk2, vk2 = v2 * dt, a2 * dt
32             r += rk2
33             v += vk2
34             positions[i] = r
35
36             if np.linalg.norm(r) < R_T or np.linalg.norm(r) > 40 * R_T:
37                 break
38     print((np.linalg.norm(v) - np.linalg.norm(v0)) / np.linalg.norm(v0))
39     return positions[:i+1]
```

Annexe

- ❖ Code python utilisé pour la trajectoire de l'électron (3/4) :

```
41 r0 = np.array([-2*R_T, -2*R_T, 1*R_T])
42 v0 = np.array([2e5, 2e5, -2e6])
43 q = -1.6e-19
44 m = 9.1e-31
45 dt = 1e-6
46 n = 50000000
47 |
48 positions = tracage(r0, v0, q, m, dt, n)
49 r=positions[-1]
50 rayon_cube = 3*R_T
51 x_min, x_max = - rayon_cube, rayon_cube
52 y_min, y_max = - rayon_cube, rayon_cube
53 z_min, z_max = - rayon_cube, rayon_cube
54
55 grid_resolution = 7
56 x = np.linspace(x_min, x_max, grid_resolution)
57 y = np.linspace(y_min, y_max, grid_resolution)
58 z = np.linspace(z_min, z_max, grid_resolution)
59 X, Y, Z = np.meshgrid(x, y, z, indexing='ij')
60 Bx, By, Bz = np.zeros(X.shape), np.zeros(Y.shape), np.zeros(Z.shape)
```

Annexe

❖ Code python utilisé pour la trajectoire de l'électron (4/4) :

```
62 for i in range(len(x)):
63     for j in range(len(y)):
64         for k in range(len(z)):
65             r_vec = np.array([X[i,j,k], Y[i,j,k], Z[i,j,k]])
66             B = champ(r_vec)
67             Bx[i,j,k], By[i,j,k], Bz[i,j,k] = B
68
69 fig = plt.figure()
70 ax = fig.add_subplot(projection='3d')
71
72 u = np.linspace(0, 2 * np.pi, 30)
73 v = np.linspace(0, np.pi, 30)
74 u, v = np.meshgrid(u, v)
75 x, y, z = R_T*np.cos(u)*np.sin(v), R_T*np.sin(u)*np.sin(v), R_T*np.cos(v)
76 ax.plot_surface(x, y, z)
77 ax.plot(positions[:,0], positions[:,1], positions[:,2], 'black', label='Trajectoire', linewidth=3)
78 ax.scatter(positions[0,0], positions[0,1], positions[0,2], color='green', s=15, label='Départ')
79 ax.scatter(positions[-1, 0], positions[-1, 1], positions[-1, 2], color='red', s=15, label='Arrivée')
80 ax.quiver(X, Y, Z, Bx, By, Bz, length=0.2*rayon_cube, normalize=True, color='blue', alpha=0.4, label='Lignes de champ')
81 ax.set_title('Trajectoire de la particule')
82 ax.set_xlabel('x (m)')
83 ax.set_ylabel('y (m)')
84 ax.set_zlabel('z (m)')
85 ax.legend()
86 plt.show()
87
```

Annexe

❖ Code python utilisé pour la trajectoire hélicoïdale de l'électron (1/3)

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from mpl_toolkits.mplot3d import Axes3D
4
5 mu_0 = 4 * np.pi * 1e-7
6 R_T = 6.371e6
7 M_magnitude = 7.94e22
8 M_vec = np.array([0.0, 0.0, M_magnitude])
9
10 def champ(r_vec):
11     r = np.linalg.norm(r_vec)
12     B = (mu_0 / (4 * np.pi)) * (3 * np.dot(M_vec, r_vec) * r_vec - r**2 * M_vec) / r**5
13     return B
14
15 def tracage(r0, v0, q, m, dt, n):
16     r = r0.copy()
17     v = v0.copy()
18     positions = np.zeros((n, 3))
19     for i in range(n):
20         F = q * np.cross(v, champ(r))
21         a = F / m
22         r = r + v * dt
23         v = v + a * dt
24         positions[i] = r
25     return positions
26
```

Annexe

❖ Code python utilisé pour la trajectoire hélicoïdale de l'électron (2/3)

```
27 r0 = np.array([-1.5*R_T, 0, -1.5*R_T], dtype=np.float64)
28 v0 = np.array([-9e5, -2e5, 2e6], dtype=np.float64)
29 q = -1.6e-19
30 m = 9.1e-31
31 dt = 1e-9
32 n = 100000
33 positions = tracage(r0, v0, q, m, dt, n)
34 r=positions[-1]
35 rayon_cube = (np.linalg.norm(r) - np.linalg.norm(r0))
36 x_min, x_max = (r0[0]+r[0])/2 - rayon_cube, (r0[0]+r[0])/2 + rayon_cube
37 y_min, y_max = (r0[1]+r[1])/2 - rayon_cube, (r0[1]+r[1])/2 + rayon_cube
38 z_min, z_max = (r0[2]+r[2])/2 - rayon_cube, (r0[2]+r[2])/2 + rayon_cube
39
40 grid_resolution = 4
41 x = np.linspace(x_min, x_max, grid_resolution)
42 y = np.linspace(y_min, y_max, grid_resolution)
43 z = np.linspace(z_min, z_max, grid_resolution)
44 X, Y, Z = np.meshgrid(x, y, z, indexing='ij')
45 Bx, By, Bz = np.zeros_like(X), np.zeros_like(Y), np.zeros_like(Z)
46 for i in range(len(x)):
47     for j in range(len(y)):
48         for k in range(len(z)):
49             r_vec = np.array([X[i,j,k], Y[i,j,k], Z[i,j,k]])
50             B = champ(r_vec)
51             Bx[i,j,k], By[i,j,k], Bz[i,j,k] = B
52
```

Annexe

❖ Code python utilisé pour la trajectoire hélicoïdale de l'électron (3/3)

```
53 fig = plt.figure()
54 ax = fig.add_subplot(111, projection='3d')
55
56 ax.plot(positions[:,0], positions[:,1], positions[:,2], 'black', label='Trajectoire', linewidth=1.5)
57 ax.scatter(positions[0,0], positions[0,1], positions[0,2], color='green', s=20, label='Départ')
58 ax.scatter(positions[-1,0], positions[-1,1], positions[-1,2], color='red', s=20, label='Arrivée')
59 ax.quiver(X, Y, Z, Bx, By, Bz, length=0.3*rayon_cube, normalize=True, color='blue', alpha=0.4, label='Lignes de champ')
60 ax.set_title("Trajectoire de la particule")
61 ax.set_xlabel("x (m)")
62 ax.set_ylabel("y (m)")
63 ax.set_zlabel("z (m)")
64 ax.legend()
65 plt.tight_layout()
66 plt.show()
```