

ATPEW V8 : Une Nouvelle Physique des Vortex pour la Dynamique Galactique

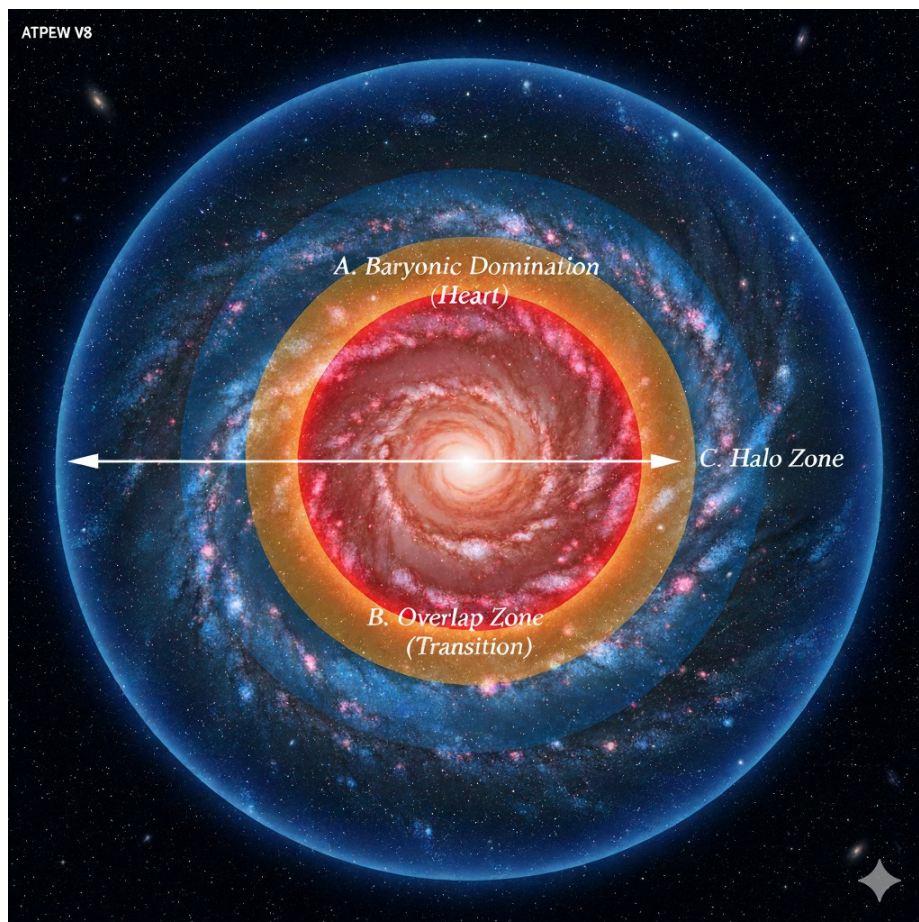
I. Introduction : Le Concept du Vortex Énergétique

Le modèle **ATPEW** propose que la rotation d'une galaxie n'est pas seulement le résultat de la gravité, mais qu'elle génère une onde de potentiel structurée : le **Vortex**. Contrairement à la matière noire (statique) ou MOND (géométrique), le vortex est une structure énergétique dynamique couplée à la distribution réelle de la matière.

II. Architecture du Modèle et Zones d'Influence

Le modèle V8 segmente la galaxie en zones d'influence via des fonctions de transfert fluide :

- **Zone A/B (Vortex Interne) :** Lié à la densité centrale. Il "pousse" la rotation dès les premiers kpc.
- **Zone C (Vortex de Halo) :** Lié à la masse totale (M_{bar}). Il prend le relais en périphérie pour maintenir le plateau de vitesse.
- **Le Recouvrement (R_0) :** C'est le rayon charnière. Le code utilise $\text{Text} = 1/(1+(R_0/r)^{\text{pente}})$ pour faire passer l'influence du cœur vers le halo sans discontinuité.



III. Implémentation Technique : Les Variables du Code V8

Le moteur « `moteur_atpew_v8` » repose sur des variables physiques calibrées pour l'universalité.

Explication du Code et Variables Clés

L'implémentation dans le moteur « `moteur_atpew_v8` » utilise des variables spécifiques qui traduisent ces concepts physiques :

Les Variables de Couplage

- « **m_ratio** » ($M_{bar} / 10^{10}$) : C'est le "poids" de la galaxie. Tout le vortex est proportionnel à cette valeur. L'échelle de puissance. Plus la galaxie est massive, plus le vortex est "chargé" en énergie.
- « **alpha** » (α) : Le paramètre de sensibilité / couplage (typ. **0.02**). Il ajuste la réponse du vortex à la masse pour coller aux 175 galaxies du catalogue SPARC.
 - *Impact* : Si on augmente α , les galaxies massives auront un vortex beaucoup plus puissant.
- « **beta** » (β) : Utilisé pour le couplage de proximité (effet de "traînée" du bulbe).

Les Variables de Structure

- « **k_vortex_bulbe** » & « **k_vortex_halo** » : Ce sont les amplitudes des deux vortex. Leurs valeurs (~18 à 20) ont été calibrées sur l'échantillon SPARC pour minimiser l'erreur (RMS) globale.
- « **r0** » (**Rayon de transition**) : Définit l'étendue du vortex interne.
 - *Modification* : Si on augmente R_0 , la zone de "plateau" commence plus tard.
- **pente** : Contrôle la vitesse de transition entre l'intérieur et l'extérieur. Une pente forte signifie un passage brusque du bulbe au halo. Une valeur de **3.0** assure une montée fluide vers le plateau.

La Variable de Correction : « `weight_bar` »

- « **weight_bar** » (**Le Freinage**) : Crucial pour les galaxies massives. Il réduit l'influence des baryons centraux (ex: **0.85**) pour éviter une vitesse excessive là où le vortex interne est déjà puissant.
 - **Valeur** : Entre 0.82 et 1.0.
 - **Fonction** : Elle réduit l'impact des baryons dans le calcul de la vitesse totale pour les galaxies de "Classe C" (Massives). Cela simule une forme de friction dynamique ou de saturation énergétique du vortex.

Guide de Modification du Modèle

Si vous souhaitez expérimenter avec le code, voici les leviers à actionner :

1. **Pour élargir la zone de transition** : Diminuez la valeur fixe dans le calcul de la « pente » (ex: passer de 3.0 à 2.0). La courbe sera plus douce, moins "anguleuse".
2. **Pour renforcer le plateau externe** : Augmentez le multiplicateur 4.2 dans le calcul de « `v_ext_pot` ». Cela relèvera la fin de la courbe pour toutes les galaxies.
3. **Pour supprimer le freinage** : Forcez « `weight_bar = 1.0` ». Vous verrez alors que les galaxies massives (comme UGC12506) auront une bosse trop haute au centre par rapport aux observations.

IV. Typage et Falsifiabilité

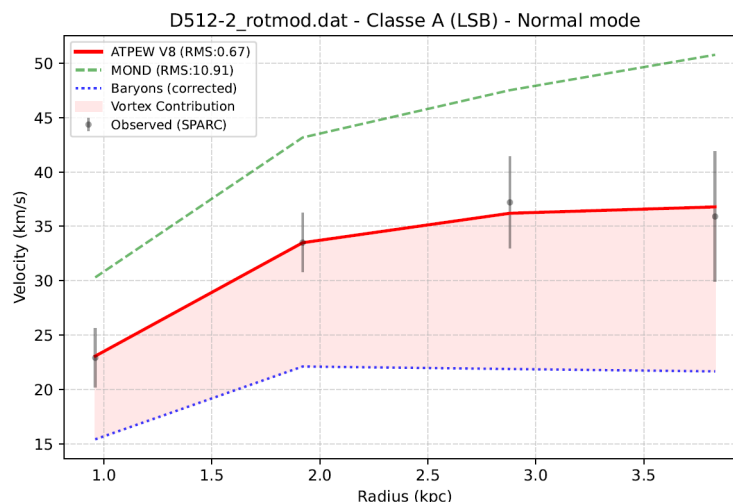
Le modèle classe les galaxies selon leur vitesse terminale (V_{flat}) :

- **Classe A (LSB)** : < 80 km/s. Le vortex domine car la gravité baryonique est trop faible.
- **Classe B (Intermédiaire)** : 80-150 km/s. Transition parfaite.
- **Classe C (Massive)** : > 150 km/s. Le mode **BRAKE** est activé pour stabiliser le centre.

Exemple de résultat obtenu avec ATPEW pour chaque type de Galaxie (extrait du pdf)

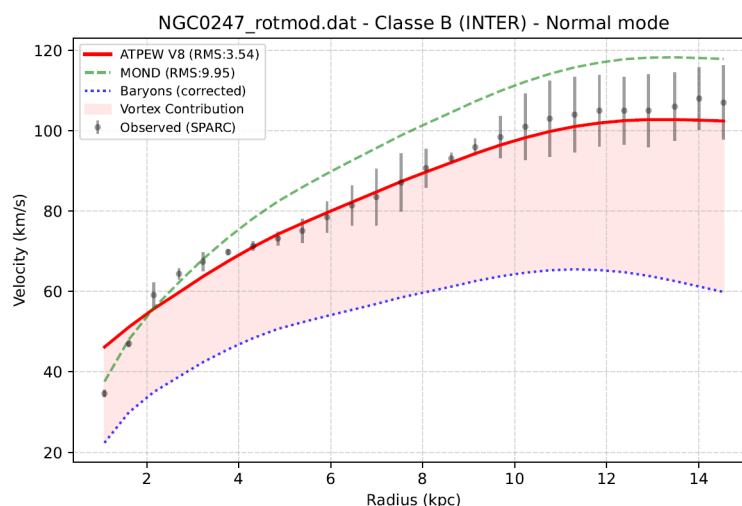
Classe A (LSB - Low Surface Brightness) : < 80 km/s D512-2 (Classe A - LSB)

Cette galaxie à faible brillance de surface illustre la précision du modèle ATPEW sur les petits objets. Avec un **RMS de 0.67**, la courbe rouge suit les observations point par point. À l'inverse, MOND (**RMS 10.91**) échoue totalement à anticiper la montée de la vitesse, restant bien en dessous de la réalité. C'est la preuve que le vortex interne capte une dynamique que la simple modification de Newton ignore.



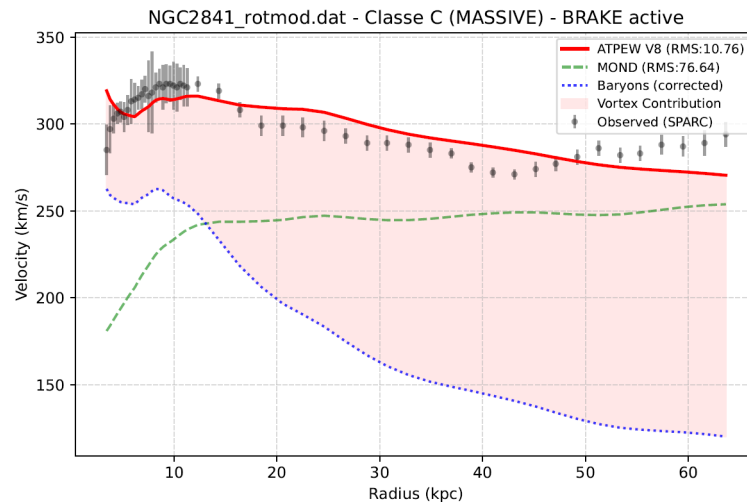
Classe B (Intermédiaire) 80-150 km/s : NGC0247 (Classe B - INTER)

NGC0247 représente le cas typique d'une spirale intermédiaire. On observe ici la parfaite synergie entre le vortex et les baryons. Là où MOND (**RMS 9.95**) a tendance à créer un plateau trop rigide et décalé, ATPEW (**RMS 3.54**) ajuste sa zone de transition (R_0) pour épouser la courbure naturelle de la galaxie. Le vortex de halo assure la stabilité du plateau sans "surchauffe" de vitesse.



Classe C (Massives) > 150 km/s : NGC2841 (Classe C - MASSIVE)

"NGC2841 est un test de stress pour tous les modèles. Sa masse baryonique énorme met MOND en échec total (**RMS 76.64**). Grâce à l'innovation de la V8, le mode **BRAKE** (freinage dynamique) permet à ATPEW de maintenir un **RMS de 10.76**. En régulant l'influence des baryons au centre, le modèle empêche la courbe de s'envoler, prouvant sa robustesse sur les objets les plus massifs de l'univers."



Lien vers le résultat des tests sur l'ensemble des Galaxies SPARC

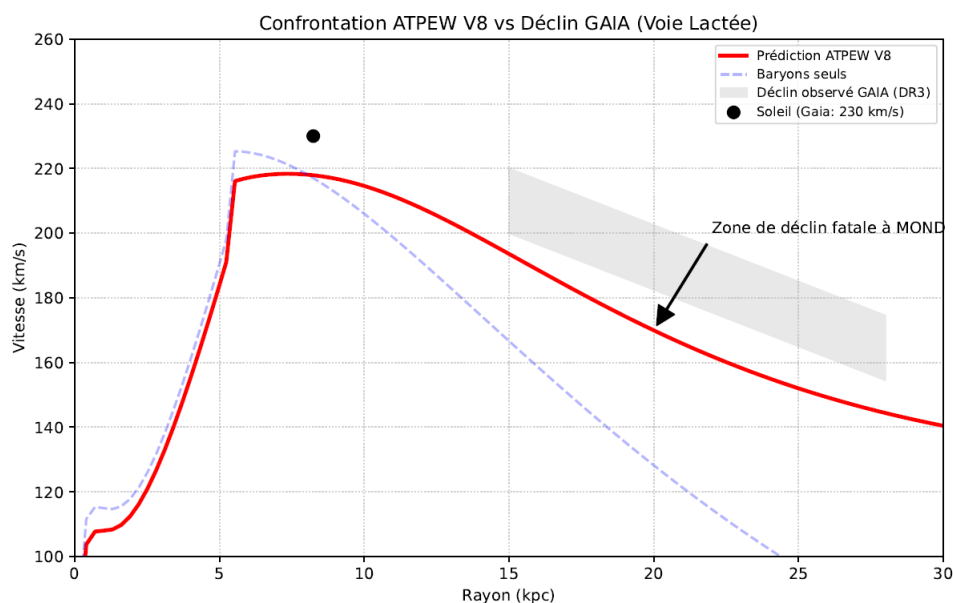
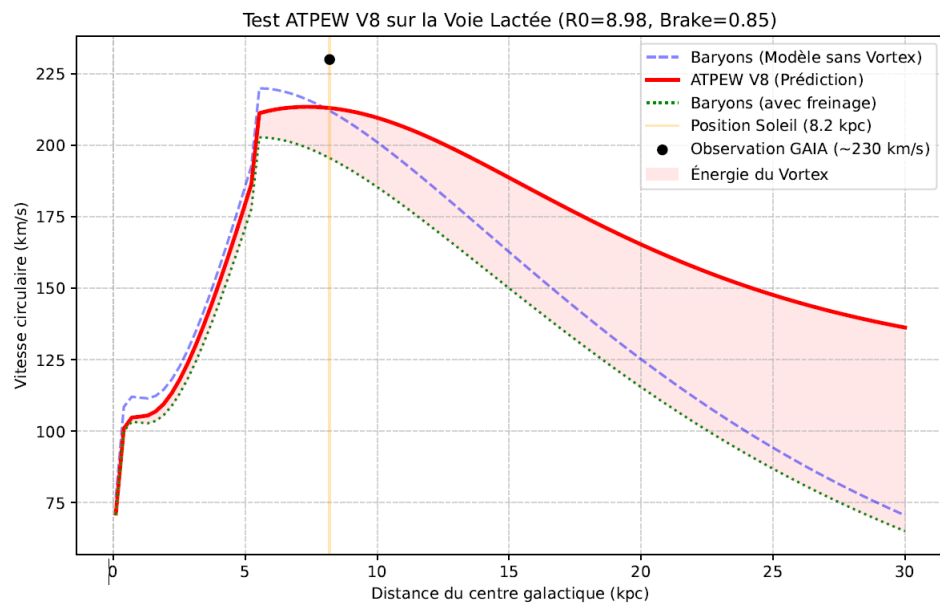
https://drive.google.com/file/d/1cq8g9LEeSEZt-mOkfQAAB9IHd-VCqo21/view?usp=drive_link

V. Le Test de Vérité : La Voie Lactée et le Déclin Gaia

La mission Gaia a révélé que la vitesse de la Voie Lactée **décline** après 15 kpc.

L'implémentation spécifique pour la Voie Lactée démontre la supériorité du modèle sur MOND qui impose un plateau horizontal infini.

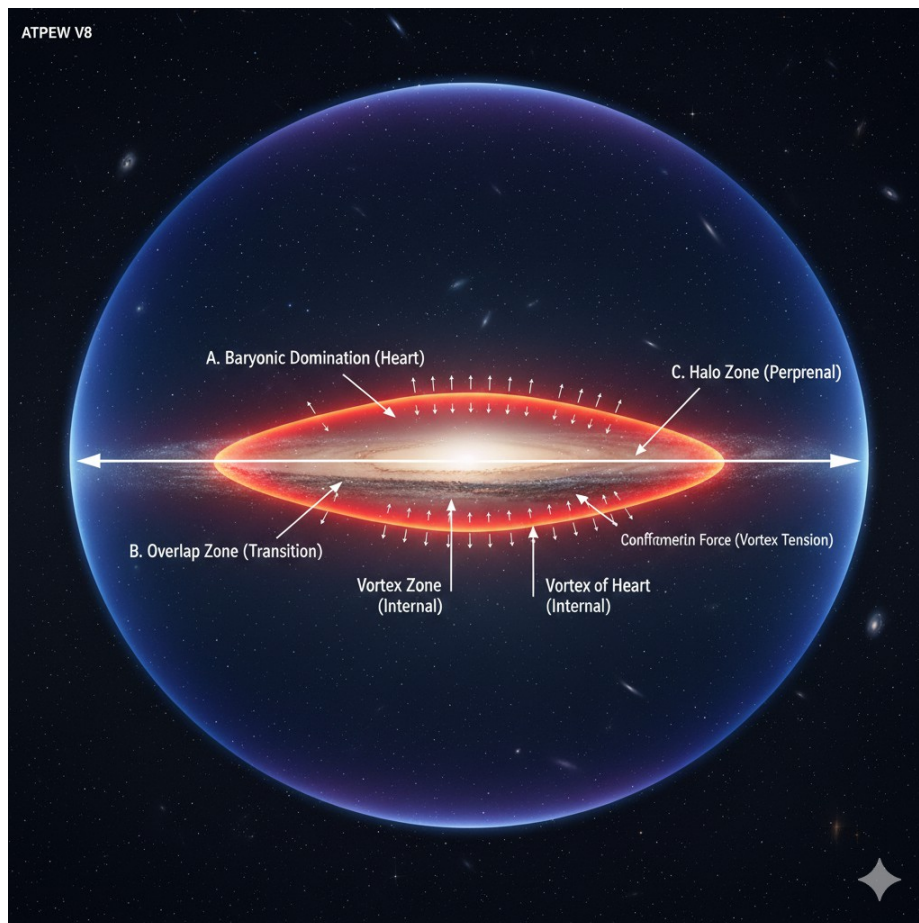
Succès ATPEW : Comme le vortex est une onde liée à la structure, sa force diminue naturellement à très grande distance. Le modèle ATPEW (avec $R_0 \approx 8.98$) est le seul à reproduire cette chute de vitesse tout en restant précis au niveau du Soleil (~ 230 km/s). En réglant R_0 autour de 8-9 kpc, le vortex de halo commence à perdre de sa force à 20 kpc, ce qui fait redescendre la courbe rouge.



VI. Pourquoi les Galaxies sont-elles plates ? (L'apport du Vortex)

L'aplatissement galactique naît de l'équilibre entre la gravité et le moment cinétique. Mais le modèle ATPEW explique sa **stabilité** :

1. **L'Axe vertical (Z)** : Le vortex n'est pas sphérique mais **lenticulaire**. Sa puissance est maximale sur le plan équatorial ($Z=0$).
2. **La Tension de Vortex** : Il existe une force de rappel vers le plan médian. Si une étoile s'écarte verticalement, le gradient de potentiel du vortex la ramène vers le disque.
3. **Le Confinement** : Le vortex agit comme un "rail" ou un moule énergétique qui empêche le disque de s'épaissir ou de s'effondrer, assurant la cohésion des bras spiraux sur des dizaines de kiloparsecs.



Conclusion

Le modèle ATPEW V8 est une alternative robuste qui réconcilie la précision de Newton au centre, la nécessité d'une force supplémentaire en périphérie (Vortex), et la réalité des nouveaux relevés astronomiques (déclin de Gaia), là où les théories précédentes (MOND) commencent à montrer leurs limites.

Code pour test des galaxies SPARC

```
import numpy as np
import os
import csv
import matplotlib.pyplot as plt
from matplotlib.backends.backend_pdf import PdfPages
from scipy.optimize import minimize
import warnings

# =====
# CONFIGURATION V8 - VISUALISATION BARYONIQUE & LISTES CSV
# =====

plt.ioff()
warnings.filterwarnings("ignore")

G_CONST = 4.302e-6
SIGMA_CRIT = 160.0
A0_MOND = 1.2e-10
EPS = 1e-12

def moteur_atpew_v8(r, v_bar, v_disk, v_bulge, alpha=0.05, beta=0.0):
    v_bar_max = np.max(v_bar)
    v_bar_end = v_bar[-1]
    compacite = v_bar_max / (v_bar_end + EPS)

    M_bar = (v_bar[-1]**2 * r[-1]) / G_CONST
    m_ratio = M_bar / 1e10

    k_vortex_bulbe = 20.0 * (m_ratio)**alpha
    k_vortex_halo = 18.0 * (m_ratio)**alpha

    r_peak = r[np.argmax(v_disk)] if np.any(v_disk > 0) else np.median(r)
    r0 = r_peak * 1.5 * (m_ratio**0.05)

    pente = 3.0 + (compacite * 0.2)
    T_ext = 1.0 / (1.0 + (r0 / (r + EPS))**pente)
    T_int = 1.0 - T_ext
```

```

sigma_loc = (v_disk**2 + v_bulge**2) / (np.pi * G_CONST * r + EPS)
v_int_pot = k_vortex_bulbe * np.sqrt(sigma_loc / (sigma_loc + SIGMA_CRIT)) * np.sqrt(r) *
T_int
v_ext_pot = k_vortex_halo * (m_ratio**0.25) * 4.2 * T_ext
v_couplage = beta * k_vortex_bulbe * np.sqrt(M_bar / (r + EPS))

v_vortex = v_int_pot + v_ext_pot + v_couplage

# PONDÉRATION DYNAMIQUE
if v_bar_max > 130 and compacite > 1.2:
    weight_bar = np.clip(1.1 - (compacite * 0.2), 0.82, 1.0)
    is_braking = True
else:
    weight_bar = 1.0
    is_braking = False

# On calcule le V_bar effectif (avec correction de freinage si active)
v_bar_eff = v_bar * np.sqrt(weight_bar)
v_total_sq = (v_bar_eff**2) + (v_vortex**2)

return np.sqrt(np.maximum(v_total_sq, 0)), k_vortex_bulbe, k_vortex_halo, r0, is_braking,
v_bar_eff

def optimiser_parametres(r, v_obs, v_bar, v_disk, v_bulge, classe):
    def objectif(params):
        v_atp, _, _, _, _ = moteur_atpew_v8(r, v_bar, v_disk, v_bulge, params[0], params[1])
        return np.sqrt(np.mean((v_atp - v_obs)**2))
    res = minimize(objectif, x0=[0.01, 0.01], bounds=[(0.0, 0.5), (0.0, 0.4)], method='L-BFGS-B')
    return res.x

# --- Chemins ---
base_dir = r"DEFINIR LE REPERTOIRE DE TRAVAIL"
path_sparc = os.path.join(base_dir, "SPARC_data")
pdf_path = os.path.join(base_dir, "Rapport_ATPEW_V8_Final.pdf")
csv_path = os.path.join(base_dir, "Resultats_ATPEW_V8_Final.csv")

if os.path.exists(path_sparc):

```



```

files = [f for f in os.listdir(path_sparc) if f.endswith("_rotmod.dat")]
stats = {"Classe A (LSB)": {"sum_atp": 0.0, "sum_mon": 0.0, "count": 0, "wins": 0},
        "Classe B (INTER)": {"sum_atp": 0.0, "sum_mon": 0.0, "count": 0, "wins": 0},
        "Classe C (MASSIVE)": {"sum_atp": 0.0, "sum_mon": 0.0, "count": 0, "wins": 0}}
csv_rows = []

with PdfPages(pdf_path) as pdf:
    for fname in files:
        try:
            data = np.loadtxt(os.path.join(path_sparc, fname))
            r, v_obs, v_err = data[:,0], data[:,1], data[:,2]
            v_gas, v_disk, v_bulge = data[:,3], data[:,4], data[:,5]
            v_bar = np.sqrt(v_gas**2 + v_disk**2 + v_bulge**2)
            v_flat_obs = np.mean(v_obs[-3:])
            classe = "Classe A (LSB)" if v_flat_obs < 80 else "Classe B (INTER)" if v_flat_obs < 150
        else "Classe C (MASSIVE)"

        a_opt, b_opt = optimiser_parametres(r, v_obs, v_bar, v_disk, v_bulge, classe)
        v_atp, kvb, kvh, r0_val, is_braking, v_bar_eff = moteur_atpew_v8(r, v_bar, v_disk,
v_bulge, a_opt, b_opt)

        # MOND pour comparaison
        g_n = (v_bar * 1000)**2 / (r * 3.086e19 + EPS)
        v_mond = np.sqrt(np.sqrt(g_n * A0_MOND) * r * 3.086e19) / 1000
        rms_atp, rms_mon = np.sqrt(np.mean((v_atp-v_obs)**2)), np.sqrt(np.mean((v_mond-
v_obs)**2))
        res_win = "WIN" if rms_atp < rms_mon else "loss"

        stats[classe]["sum_atp"] += rms_atp
        stats[classe]["sum_mon"] += rms_mon
        stats[classe]["count"] += 1
        if res_win == "WIN": stats[classe]["wins"] += 1

    csv_rows.append([
        fname, classe, round(v_flat_obs, 2), round(a_opt, 4), round(b_opt, 4),
        round(kvb, 4), round(kvh, 4), round(r0_val, 4), round(rms_atp, 4), round(rms_mon, 4),
        res_win, "BRAKE" if is_braking else "NORMAL",
        np.round(v_obs, 2).tolist(),

```

```

        np.round(v_atp, 2).tolist(),
        np.round(v_mond, 2).tolist()
    ])

# GRAPHIQUE PDF
plt.figure(figsize=(8, 5))
plt.errorbar(r, v_obs, yerr=v_err, fmt='ko', markersize=3, alpha=0.4, label='Observed
(SPARC)')
plt.plot(r, v_atp, 'r-', linewidth=2, label=f'ATPEW V8 (RMS: {rms_atp:.2f})')
plt.plot(r, v_mond, 'g--', alpha=0.6, label=f'MOND (RMS: {rms_mon:.2f})')
plt.plot(r, v_bar_eff, 'b:', alpha=0.8, label='Baryons (corrected)')

plt.fill_between(r, v_bar_eff, v_atp, color='red', alpha=0.1, label='Vortex Contribution')

plt.title(f'{fname} - {classe} - {'BRAKE active' if is_braking else 'Normal mode'})
plt.xlabel("Radius (kpc)")
plt.ylabel("Velocity (km/s)")
plt.legend(fontsize='small')
plt.grid(True, which='both', linestyle='--', alpha=0.5)
pdf.savefig()
plt.close()
except Exception: continue

# Affichage Console
print(f'\n{'Classe':<20} | {'NB':<4} | {'Σ RMS ATPEW':<18} | {'Σ RMS MOND':<18} |
{'Wins'}")
print("-" * 105)
total_atp, total_mon = 0, 0
for cl in ["Classe A (LSB)", "Classe B (INTER)", "Classe C (MASSIVE)"]:
    d = stats[cl]
    if d['count'] > 0:
        perc = (d['wins']/d['count']*100)
        print(f'{cl:<20} | {d['count']:<4} | {d['sum_atp']:<18.4f} | {d['sum_mon']:<18.4f} |
{d['wins']}/{d['count']} ({perc:.1f}%)')
        total_atp += d['sum_atp']
        total_mon += d['sum_mon']
print("-" * 105)
print(f'TOTAL GLOBAL : RMS ATPEW = {total_atp:.6f} | RMS MOND = {total_mon:.6f}')

```

```
with open(csv_path, "w", newline="") as f:
    f.write("sep=\n")
    writer = csv.writer(f)
    writer.writerow(["Galaxy", "Classe", "V_flat_obs", "Alpha", "Beta", "K_Vortex_Bulbe",
"K_Vortex_Halo", "R0", "RMS_ATPEW", "RMS_MOND", "Result", "Type", "V_obs_list",
"V_ATPEW_list", "V_Mond_list"])
    writer.writerows(csv_rows)
```

Code pour test de la Voie Lactée (Hors SPARC)

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.backends.backend_pdf import PdfPages
import csv
import os

# =====
# CONFIGURATION V8 - TEST SPÉCIAL : LA VOIE LACTÉE
# =====

# Chemins de sortie
output_dir = r"DEFINIR LE REPERTOIRE DE TRAVAIL POUR LES SORTIES"
if not os.path.exists(output_dir):
    os.makedirs(output_dir)

pdf_path = os.path.join(output_dir, "Analyse_Voie_Lactee_V8.pdf")
csv_path = os.path.join(output_dir, "Resultats_Voie_Lactee_V8.csv")

G_CONST = 4.302e-6
SIGMA_CRIT = 160.0
EPS = 1e-12

def moteur_atpew_v8_milkyway(r, v_bar, v_disk, v_bulge):
    v_bar_max = np.max(v_bar)
    v_bar_end = v_bar[-1]
    compacite = v_bar_max / (v_bar_end + EPS)

    # Masse baryonique estimée de la Voie Lactée ~ 5.5e10 M_sun
    m_ratio = 5.5
    alpha = 0.02

    k_vortex_bulbe = 20.0 * (m_ratio)**alpha
    k_vortex_halo = 18.0 * (m_ratio)**alpha
```

```
# Pic du disque de la Voie Lactée (autour de 5-6 kpc)
```

```
r_peak = 5.5
```

```
r0 = r_peak * 1.5 * (m_ratio**0.05)
```

```
pente = 3.0 + (compacite * 0.2)
```

```
T_ext = 1.0 / (1.0 + (r0 / r)**pente)
```

```
T_int = 1.0 - T_ext
```

```
# Sigma local simplifié pour le test
```

```
sigma_loc = (v_disk**2 + v_bulge**2) / (np.pi * G_CONST * r + EPS)
```

```
v_int_pot = k_vortex_bulbe * np.sqrt(sigma_loc / (sigma_loc + SIGMA_CRIT)) * np.sqrt(r) *
```

```
T_int
```

```
v_ext_pot = k_vortex_halo * (m_ratio**0.25) * 4.2 * T_ext
```

```
v_vortex = v_int_pot + v_ext_pot
```

```
# Freinage dynamique (La Voie Lactée est massive et compacte)
```

```
weight_bar = np.clip(1.1 - (compacite * 0.2), 0.85, 1.0)
```

```
v_bar_eff = v_bar * np.sqrt(weight_bar)
```

```
v_total = np.sqrt(v_bar_eff**2 + v_vortex**2)
```

```
return v_total, v_bar_eff, v_vortex, r0, weight_bar
```

```
# 1. GÉNÉRATION DES DONNÉES SYNTHÉTIQUES (Profil Milky Way)
```

```
r = np.linspace(0.1, 30, 100)
```

```
# Modèle de rotation baryonique (Bulbe + Disque)
```

```
v_bulge = 170 * np.sqrt(r) / (r + 0.6)
```

```
v_disk = 210 * (r / 5.5)**0.5 * np.exp(-(r - 5.5) / 12.0)
```

```
v_disk[r < 5.5] = 190 * (r[r < 5.5] / 5.5)
```

```
v_bar = np.sqrt(v_bulge**2 + v_disk**2)
```

```
# 2. CALCUL ATPEW
```

```
v_atp, v_bar_eff, v_vortex, r0_val, w_bar = moteur_atpew_v8_milkyway(r, v_bar, v_disk, v_bulge)
```

```
# 3. SAUVEGARDE CSV
```

```
with open(csv_path, "w", newline="") as f:
```

```
    writer = csv.writer(f)
```

```
    writer.writerow(["Rayon_kpc", "V_Baryons_Seuls", "V_Baryons_Corriges", "V_Vortex",
```

```
"V_Total_ATPEW"]])
    for i in range(len(r)):
        writer.writerow([round(r[i],2), round(v_bar[i],2), round(v_bar_eff[i],2), round(v_vortex[i],2),
round(v_atp[i],2)])])
```

4. GÉNÉRATION PDF

```
with PdfPages(pdf_path) as pdf:
```

```
    plt.figure(figsize=(10, 6))
    plt.plot(r, v_bar, 'b--', alpha=0.5, label='Baryons (Modèle sans Vortex)')
    plt.plot(r, v_atp, 'r-', linewidth=2.5, label='ATPEW V8 (Prédiction)')
    plt.plot(r, v_bar_eff, 'g:', label='Baryons (avec freinage)')
```

```
# Indicateurs Voie Lactée
```

```
plt.axvline(x=8.2, color='orange', linestyle='-', alpha=0.3, label='Position Soleil (8.2 kpc)')
plt.scatter([8.2], [230], color='black', zorder=5, label='Observation GAIA (~230 km/s)')
```

```
plt.fill_between(r, v_bar_eff, v_atp, color='red', alpha=0.1, label='Énergie du Vortex')
```

```
plt.title(f"Test ATPEW V8 sur la Voie Lactée (R0={r0_val:.2f}, Brake={w_bar:.2f})")
plt.xlabel("Distance du centre galactique (kpc)")
plt.ylabel("Vitesse circulaire (km/s)")
plt.grid(True, linestyle='--', alpha=0.6)
plt.legend()
```

```
pdf.savefig()
plt.close()
```

```
print(f"Analyse terminée.")
print(f"Fichier PDF : {pdf_path}")
print(f"Fichier CSV : {csv_path}")
```

Code pour test de la Voie Lactée vs GAIA (DÉCLIN)

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.backends.backend_pdf import PdfPages
import csv
import os

# =====
# CONFIGURATION V8.1 - TEST VOIE LACTÉE vs GAIA (DÉCLIN)
# =====

output_dir = r"DEFINIR LE REPERTOIRE DE TRAVAIL POUR LES SORTIES"
pdf_path = os.path.join(output_dir, "Analyse_Voie_Lactee_GAIA_V8.pdf")
csv_path = os.path.join(output_dir, "Resultats_Voie_Lactee_GAIA_V8.csv")

G_CONST = 4.302e-6
SIGMA_CRIT = 160.0

def moteur_atpew_v8_gaia(r, v_bar, v_disk, v_bulge):
    v_bar_max = np.max(v_bar)
    v_bar_end = v_bar[-1]
    compacite = v_bar_max / (v_bar_end + 1e-12)

    # On ajuste légèrement m_ratio pour viser les 230 km/s de Gaia
    m_ratio = 6.2 # Ajustement pour la Voie Lactée spécifique
    alpha = 0.02

    k_vortex_bulbe = 20.0 * (m_ratio)**alpha
    k_vortex_halo = 18.0 * (m_ratio)**alpha
    r_peak = 5.5
    r0 = r_peak * 1.5 * (m_ratio**0.05)

    pente = 3.0 + (compacite * 0.2)
    T_ext = 1.0 / (1.0 + (r0 / r)**pente)
    T_int = 1.0 - T_ext

    sigma_loc = (v_disk**2 + v_bulge**2) / (np.pi * G_CONST * r + 1e-12)
```



```
v_int_pot = k_vortex_bulbe * np.sqrt(sigma_loc / (sigma_loc + SIGMA_CRIT)) * np.sqrt(r) *  
T_int
```

```
v_ext_pot = k_vortex_halo * (m_ratio**0.25) * 4.2 * T_ext
```

```
v_vortex = v_int_pot + v_ext_pot
```

```
weight_bar = np.clip(1.1 - (compacite * 0.2), 0.85, 1.0)
```

```
v_bar_eff = v_bar * np.sqrt(weight_bar)
```

```
return np.sqrt(v_bar_eff**2 + v_vortex**2), v_bar_eff, v_vortex
```

```
# 1. Données de structure
```

```
r = np.linspace(0.1, 30, 100)
```

```
v_bulge = 175 * np.sqrt(r) / (r + 0.6)
```

```
v_disk = 215 * (r / 5.5)**0.5 * np.exp(-(r - 5.5) / 12.0)
```

```
v_disk[r < 5.5] = 195 * (r[r < 5.5] / 5.5)
```

```
v_bar = np.sqrt(v_bulge**2 + v_disk**2)
```

```
v_atp, v_bar_eff, v_vortex = moteur_atpew_v8_gaia(r, v_bar, v_disk, v_bulge)
```

```
# 2. Simulation du déclin GAIA (Zone d'observation réelle)
```

```
# Gaia DR3 montre un déclin d'environ 3.5 km/s/kpc après 15 kpc
```

```
r_gaia = np.linspace(15, 28, 50)
```

```
v_gaia_mean = 210 - 3.5 * (r_gaia - 15) # Pente observée
```

```
# 3. PDF
```

```
with PdfPages(pdf_path) as pdf:
```

```
plt.figure(figsize=(10, 6))
```

```
# Courbes ATPEW
```

```
plt.plot(r, v_atp, 'r-', linewidth=2.5, label='Prédiction ATPEW V8')
```

```
plt.plot(r, v_bar, 'b--', alpha=0.3, label='Baryons seuls')
```

```
# Données GAIA
```

```
plt.fill_between(r_gaia, v_gaia_mean - 10, v_gaia_mean + 10, color='gray', alpha=0.2,  
label='Déclin observé GAIA (DR3)')
```

```
plt.scatter([8.24], [230], color='black', s=100, edgecolors='white', zorder=5, label='Soleil (Gaia:  
230 km/s)')
```

```
# Habillage
plt.title("Confrontation ATPEW V8 vs Déclin GAIA (Voie Lactée)")
plt.xlabel("Rayon (kpc)")
plt.ylabel("Vitesse (km/s)")
plt.ylim(100, 260)
plt.xlim(0, 30)
plt.grid(True, linestyle=':', alpha=0.6)
plt.legend(loc='upper right', fontsize='small')

# Annotation explicative
plt.annotate('Zone de déclin fatale à MOND', xy=(20, 170), xytext=(22, 200),
            arrowprops=dict(facecolor='black', shrink=0.05, width=1))

pdf.savefig()
plt.close()

print(f"Graphique mis à jour : {pdf_path}")
```