

Documentation des Fonctions d'Analyse des Taux de Collision

Généré par IA (basé sur le script `auxtools.py`)

14 octobre 2025

Résumé

Ce document fournit une référence claire et structurée des fonctions Python incluses dans le script d'analyse des taux de collision. Chaque fonction est décrite en français avec ses paramètres d'entrée, son type de retour et son rôle principal dans la préparation ou la visualisation des données de niveaux quantiques.

Table des matières

1 Conventions Utilisées

- Les noms de fonctions, de DataFrames et de colonnes sont affichés en **bleu** (code).
- Les DataFrames Pandas sont désignés par `pd.DataFrame`.
- Les indices quantiques utilisés sont : v (vibrationnel) et J (rotationnel).

2 Fonctions de Préparation des Données

Ces fonctions sont essentielles pour enrichir les données de collision brutes avec des informations sur les niveaux quantiques et des valeurs effectives.

2.1 `merge_levels_simple`

TABLE 1: Description de la fonction `merge_levels_simple`

Rôle Principal	Fusionne les données de taux de collision avec les numéros quantiques (v , J) des niveaux supérieur et inférieur associés, en utilisant les indices numériques de niveau (<code>nu</code> , <code>nl</code>).
Entrée : <code>df_coll</code>	<code>pd.DataFrame</code> : Données de collision (doit contenir les indices numériques des niveaux, par défaut <code>nu</code> et <code>nl</code>).
Entrée : <code>df_level_co</code>	<code>pd.DataFrame</code> : Données détaillées des niveaux (doit contenir <code>n</code> , <code>v</code> , <code>J</code>).
Entrée : <code>cols</code>	list de str (Défaut : ['nu', 'nl']) : Noms des colonnes d'index de niveau dans <code>df_coll</code> .
Type de Retour	<code>pd.DataFrame</code> : DataFrame de collision enrichi avec les colonnes <code>vu</code> , <code>Ju</code> , <code>vl</code> et <code>Jl</code> (en tant que int64).

2.2 `sort_and_create_effective_values`

TABLE 2: Description de la fonction `sort_and_create_effective_values`

Rôle Principal	Trie le DataFrame (par v puis J pour les niveaux supérieur et inférieur) et calcule les valeurs quantiques effectives (V_{eff}) selon la formule : $V_{\text{eff}} = v + \frac{J}{J_{\text{max}}(v)}$
Entrée : <code>df_merged</code>	<code>pd.DataFrame</code> : Le DataFrame fusionné, contenant les colonnes <code>vu</code> , <code>Ju</code> , <code>vl</code> , et <code>Jl</code> .
Type de Retour	<code>pd.DataFrame</code> : Le DataFrame trié, enrichi des colonnes <code>V_eff_u</code> et <code>V_eff_l</code> .

2.3 `calculate_v_eff` (Fonction Auxiliaire)

TABLE 3: Description de la fonction `calculate_v_eff`

Rôle Principal	Fonction utilitaire interne pour calculer la valeur effective V_{eff} en fusionnant la valeur J_{max} correspondante.
Entrée : <code>df</code>	<code>pd.DataFrame</code> : DataFrame contenant les colonnes v et J (ex. <code>vu</code> , <code>Ju</code>).

Entrée : <code>prefix</code>	str : Préfixe du niveau (<code>u</code> ou <code>l</code>).
Entrée : <code>j_max_series</code>	pd.Series : Série des J_{\max} indexée par la valeur de v .
Type de Retour	pd.DataFrame : Le DataFrame mis à jour avec la colonne V_{eff} (ex. <code>V_eff_u</code>).

3 Fonctions de Détection des Données Manquantes

Ces fonctions permettent d'identifier les paires de niveaux pour lesquelles des données de collision sont théoriquement possibles mais non disponibles.

3.1 `get_missing_levels_df`

TABLE 4: Description de la fonction `get_missing_levels_df`

Rôle Principal	Génère un DataFrame des combinaisons de niveaux (v_u , J_u , v_l , J_l) qui sont manquantes dans le jeu de données actuel, en se basant sur l'ensemble des niveaux observés.
Entrée : <code>df_merged</code>	pd.DataFrame : DataFrame des données existantes.
Entrée : <code>c_col</code>	str (Défaut : '100.0') : Colonne des données de collision utilisée pour identifier les valeurs manquantes (NaN).
Type de Retour	pd.DataFrame : Un DataFrame trié contenant uniquement les colonnes v_u , J_u , v_l , J_l des niveaux manquants.

3.2 `get_missing_levels_df_2`

TABLE 5: Description de la fonction `get_missing_levels_df_2`

Rôle Principal	Identifie les combinaisons manquantes en appliquant la contrainte physique : le niveau supérieur effectif doit être strictement plus grand que le niveau inférieur effectif ($V_{\text{eff}_u} > V_{\text{eff}_l}$).
Entrée : <code>df_merged</code>	pd.DataFrame : DataFrame des données existantes (doit contenir les colonnes V_{eff}).
Entrée : <code>c_col</code>	str (Défaut : '100.0') : Colonne des données de collision.
Type de Retour	pd.DataFrame : DataFrame des niveaux manquants *physiquement valides*. Toutes les colonnes de données de collision sont définies à NaN.

4 Fonctions de Visualisation (Plotting)

Ces fonctions génèrent des graphiques statiques (Matplotlib) ou interactifs (Plotly) pour visualiser les données de collision en fonction des valeurs effectives V_{eff} .

4.1 `plot_static_colormap`

TABLE 6: Description de la fonction `plot_static_colormap`

Rôle Principal	Génère un nuage de points statique (<code>matplotlib</code>) de V_{eff_u} vs V_{eff_l} , coloré par une colonne de données spécifiée.
Entrée : <code>df</code>	pd.DataFrame : Données à tracer.

Entrée : <code>x_col</code> , <code>y_col</code>	str (Défauts : 'V_eff_l', 'V_eff_u').
Entrée : <code>c_col</code>	str (Défaut : '100.0') : Colonne pour la colormap (couleur des points).
Entrée : <code>cmap_name</code>	str (Défaut : 'viridis') : Nom de la colormap Matplotlib.
Entrée : <code>log_color_scale</code>	bool (Défaut : False) : Si True , utilise une échelle de couleur logarithmique (LogNorm).
Type de Retour	None : Affiche le graphique.

4.2 `plot_interactive_plotly`

TABLE 7: Description de la fonction `plot_interactive_plotly`

Rôle Principal	Génère un nuage de points interactif (Plotly) de V_{eff_u} vs V_{eff_l} . Les informations détaillées (vu , Ju , vl , Jl) sont affichées au survol de la souris.
Entrée : <code>df</code>	pd.DataFrame : Données à tracer.
Entrée : <code>x_col</code> , <code>y_col</code>	str (Défauts : 'V_eff_l', 'V_eff_u').
Entrée : <code>c_col</code>	str (Défaut : '100.0') : Colonne pour la colormap.
Type de Retour	None : Affiche le graphique interactif.

4.3 `plot_interactive_with_missing`

TABLE 8: Description de la fonction `plot_interactive_with_missing`

Rôle Principal	Combine sur un même graphique Plotly les données existantes (avec colormap) et les niveaux manquants possibles (en gris transparent) pour visualiser l'exhaustivité de la matrice.
Entrée : <code>df_merged</code>	pd.DataFrame : DataFrame des données existantes.
Entrée : <code>x_col</code> , <code>y_col</code> , <code>c_col</code>	str (Défauts : 'V_eff_l', 'V_eff_u', '100.0').
Type de Retour	None : Affiche le graphique interactif combiné.

4.4 `plot_interactive_combined`

TABLE 9: Description de la fonction `plot_interactive_combined`

Rôle Principal	Génère un graphique Plotly interactif en superposant les données existantes (colorées) et les niveaux manquants (en noir transparent), en utilisant des DataFrames distincts pour les deux ensembles de points.
Entrée : <code>df_merged</code>	pd.DataFrame : Données existantes (pour la couleur).
Entrée : <code>df_missing</code>	pd.DataFrame : Données des niveaux manquants (généralement issues de <code>get_missing_levels_df_2</code>).
Entrée : <code>x_col</code> , <code>y_col</code> , <code>c_col</code>	str (Défauts : 'V_eff_l', 'V_eff_u', '100.0').
Type de Retour	None : Affiche le graphique interactif combiné.