

PARTIE III : CHAPITRE 1

SIMULATION TURBULENTE ET CALCUL DE LOIS EXACTES

Sommaire

1.1	Simuler un plasma turbulent	35
1.2	Code de post-traitement pour le calcul numérique de lois exactes	37
1.3	Mode de représentation du résultat	39
1.4	Synthèse des méthodes et choix numériques	43

Dans cette partie, nous utiliserons des simulations turbulentes de deux modèles. Le premier est le modèle CGL prenant en compte les termes Hall et de pression électronique dans la loi d'Ohm (MCGL) et le deuxième (MLF) prend en compte le tenseur de flux de chaleur. Ces modèles seront détaillés dans les chapitres 3 et 3.4. Dans ce premier chapitre, seront abordées les méthodes numériques utilisées dans l'implémentation de ces modèles (section 1.1) ainsi que celles associées au calcul des lois exactes (section 1.2) et à leur visualisation (section 1.3).

1.1 Simuler un plasma turbulent

Le code de simulation utilisé est un code tridimensionnel pseudo-spectral et versatile développé en interne à l'Observatoire de la Côte d'Azur pour l'implémentation du modèle fluide proposé et étendu par PASSOT et SULEM 2007 ; PASSOT, SULEM et HUNANA 2012 ; SULEM et PASSOT 2015. Ce modèle prend en compte les termes gyrotrope et non gyrotrope des tenseurs de pression et de flux de chaleur des électrons et des ions. Le code permet à l'utilisateur de choisir quelles contributions garder.

Les quantités sont sans dimension, les longueurs sont normalisées par d_i la longueur inertielle des ions et les vitesses par la vitesse d'Alfvén. Cela induit une constante $\beta/2$, avec β fixé à un, présente dans tous les termes dépendant des pressions. Il faudra prendre en compte cette constante par la suite.

On va supposer une équation générique $\partial_t X = \mathbf{v} \cdot \nabla X$. La simuler via un code pseudo-spectral (algorithme schématisé sur la figure I.1.1) signifie que les dérivées spatiales telles que ∇X sont effectuées dans l'espace de Fourier, tandis que les produits tels que $\mathbf{v} \cdot \nabla X$ et

l'intégration temporelle de l'équation pour obtenir les quantités au pas de temps suivant, sont effectués dans l'espace réel. Ainsi, à chaque pas de temps, un aller-retour est effectué entre les espaces réel et fréquentiel. Leur discrétisation en un nombre de points finis (les grilles), induit un crénelage (aliasing) des termes non-linéaires. Cet effet est limité par une troncation, à chaque pas de temps, du spectre de chaque quantité. L'intégration temporelle est obtenue via un schéma de Runge-Kutta d'ordre 3 (RK3), choisi pour sa stabilité devant des termes dispersifs tels que le terme de Hall. Les conditions de bords de la grille spatiale sont choisies comme périodiques afin de pouvoir utiliser la transformation de Fourier et l'algorithme de transformée de Fourier rapide (FFT).

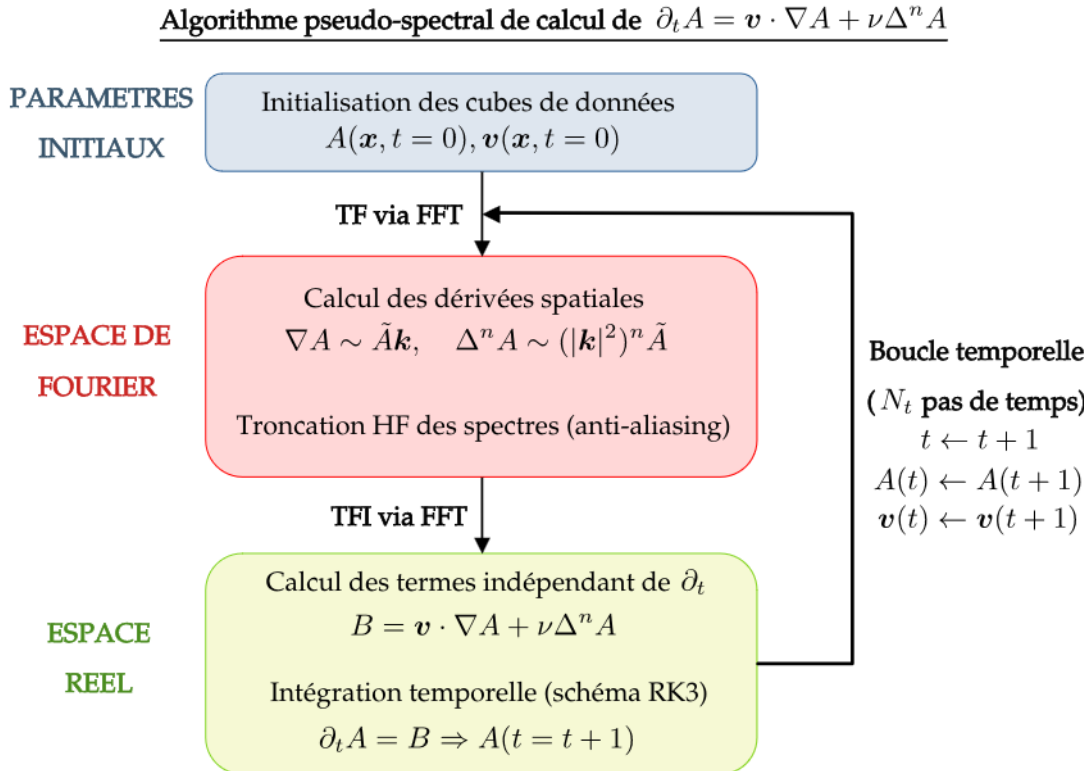


FIGURE I.1.1 – Algorithme d'intégration d'une équation d'évolution générique via une méthode pseudo-spectrale. Prise en compte des corrections anti-aliasing et hyperviscosité $\nu \Delta^n A$. TF(I) correspond à Transformée de Fourier (Inverse).

Afin de limiter l'apparition de fort gradients, ces discontinuités faisant planter la simulation, deux possibilités existent : appliquer un filtrage hautes fréquences de la quantité impliquée ou un terme d'hyperviscosité dans son équation. Le choix effectué est celui de l'hyperviscosité, c'est-à-dire imposer une décroissance graduelle et de plus en plus intense du spectre de la quantité. Ce terme de dissipation numérique s'écrit pour un champ X , $\nu \Delta^n X$. ν est une constante choisie initialement de manière empirique et n est un entier fixé à 4. Δ^n est effectué dans l'espace de Fourier où une décroissance du spectre en \mathbf{k}^{-8} est donc obtenue. En termes de turbulence, cette hyperviscosité, sera le terme de dissipation évacuant l'énergie aux petites échelles. On l'imposera donc plus important pour les champs de vitesse et magnétique que pour les champs de densité et de pression.

L'existence d'un champ magnétique moyen dans les simulations, induit une anisotropie du comportement. Sa direction est imposée suivant \mathbf{e}_z . Afin de refléter cette anisotropie, l'hyperviscosité est adaptée avec un paramètre α : Δ^n et calculée dans l'espace de Fourier

tel que $(k_x^2 + k_y^2 + \alpha k_z^2)^n$.

La grille spatiale est donc physiquement un parallélépipède allongé dans la direction \mathbf{e}_z mais nous parlerons tout de même de cubes de données. Similairement,

La cascade d'énergie est entretenue par un forçage permanent¹. Ce forçage, constitué de distribution de Dirac égales à un aux fréquences les plus petites telles que $\frac{\mathbf{k}}{|\mathbf{k}|} = \{(0, \pm 1, \pm 1); (\pm 1, \pm 1, \pm 1)\}$ est appliqué sur le champ de vitesse de telle sorte à maintenir l'énergie du système quasi-constante. Il s'éteint si l'énergie du système atteint une valeur supérieure à une certaine limite, E_{sup} , et se rallume lorsqu'elle atteint une limite inférieure, E_{inf} . Ces limites font partie des paramètres initiaux. L'angle d'injection de l'énergie, θ_i , sert à définir la forme de la grille spatiale, un parallélépipède allongé dans la direction \mathbf{e}_z , la taille physique de cette grille est fixée telle que $L_{x,y}/L_z = \tan \theta_i$.

La taille de l'espace des échelles accessibles via ces simulations dépend de la taille de la grille spatiale. L'échelle la plus petite dans une direction est la distance minimale entre deux points de la grille dans cette direction et l'échelle la plus grande la moitié de la taille de la grille. Pour une étude de turbulence, on a besoin de plusieurs ordres de grandeurs entre les échelles minimales et maximales. Afin d'obtenir un nombre de points suffisant, on part d'une grille de taille réelle fixée, mais contenant peu de points, par exemple 128^3 , puis, après avoir atteint un régime turbulent satisfaisant tel que les spectres se soient stabilisés, on augmente le nombre de points, et ainsi de suite jusqu'à avoir la taille voulue pour l'espace des échelles et un spectre stabilisé. Le nombre de points idéals serait de 1024^3 ou plus, mais plus il y a de points, plus le spectre prendra du temps à converger (cela peut atteindre plus d'un mois de temps de calcul) et plus, le calcul monopolisera de la mémoire. Similairement, le calcul du taux de cascade sera aussi plus contraignant. Un compromis doit donc être trouvé. La taille de cube minimale considérée dans le cadre des études de turbulence sera 512^3 et une partie des simulations auront une taille de 512^2 dans le plan $\{\mathbf{e}_x, \mathbf{e}_y\}$ et 1024 dans la direction \mathbf{e}_z .

Les simulations utilisées et détaillées dans les chapitres 2 et 3 ont, pour la plupart, fait l'objet de FERRAND, SAHRAOUI, LAVEDER et al. 2021. Parmi, elles, une est de taille 1024^3 . Elle ne sera pas traitée ici, car sa taille est trop importante pour le code de post-traitement implémenté et les moyens de calcul (mésocentres) à disposition. Dans ce chapitre, nous utiliserons les simulations nommées CGL1 (512^3) et CGL3 ($512^2 \times 1024$) afin de valider le code de post-traitement. Les spécificités seront analysées plus en détail dans le chapitre 2.

1.2 Code de post-traitement pour le calcul numérique de lois exactes

On a vu qu'une loi exacte est une formule statistique donnant un résultat en fonction de l'échelle ℓ . Elle dépend de quantités évaluées localement en deux points puis combinées en une expression qui est ensuite moyennée. Une partie des termes doivent ensuite être dérivés dans l'espace des échelles si aucune hypothèse d'intégration n'est effectuée. Cette méthode pourrait être implémentée directement. On considérerait les quantités à dispositions, a priori des cubes évalués en \mathbf{x} , on les translaterait de telle sorte à obtenir des cubes évalués

1. HELLINGER et al. 2018; GÓMEZ, MININNI et DMITRUK 2005; MININNI et al. 2011 utilisent une autre méthode pour obtenir le développement d'une cascade turbulente : leur champ de vitesse et champ magnétique sont initialisés par une superposition de modes de phases aléatoires, puis leurs simulations évoluent librement.

en $\mathbf{x} - \boldsymbol{\ell}$ puis on les combinerait suivant l'expression voulue avant de les moyenner. On obtiendrait ainsi notre résultat évalué en un point de l'espace des échelles et il faudrait recommencer encore et encore afin d'obtenir l'ensemble de l'espace des échelles. Enfin, on dériverait ou intégrerait le résultat. Cet algorithme est schématisé sur la figure I.1.2.

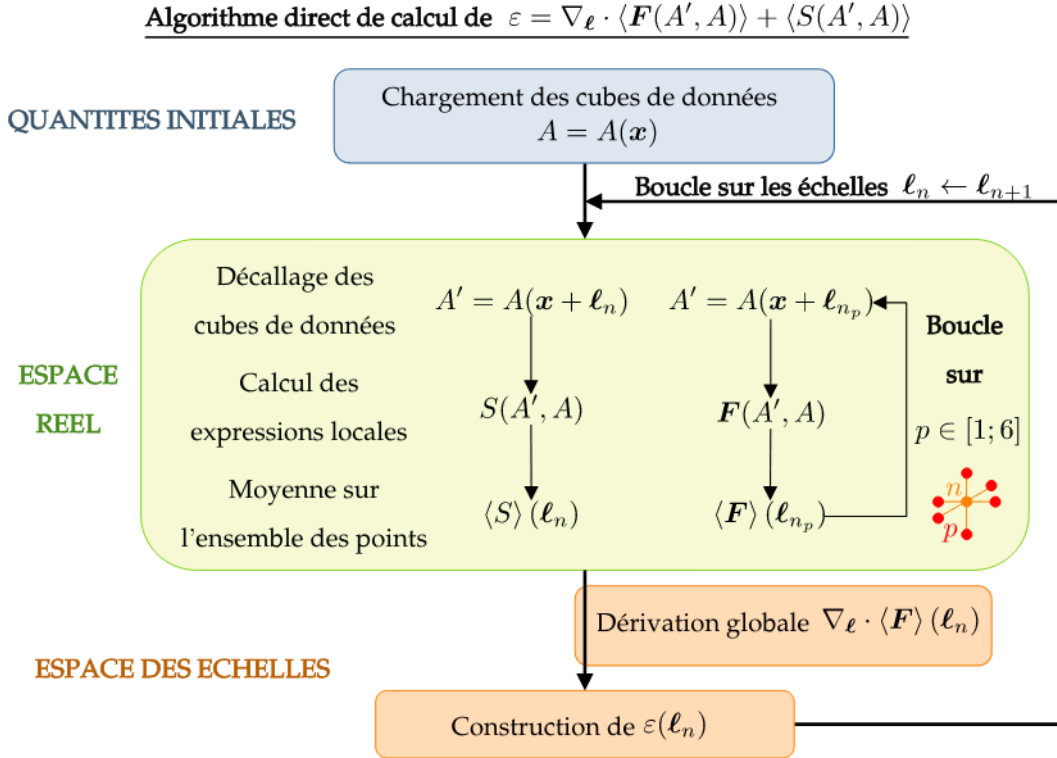


FIGURE I.1.2 – Algorithme de calcul du taux de cascade ε via la méthode directe. Les quantités impliquées sont des quantités génériques.

Cette méthode est coûteuse en temps de calcul et demande des compromis. Pour réduire le temps de calcul, on peut choisir intelligemment un certain nombre de vecteurs d'échelle. Tout d'abord, on peut jouer sur la parité de la loi exacte et ne calculer que les vecteurs tels que $\ell_z \leq 0$. FERRAND, SAHRAOUI, LAVEDER et al. 2021 ; FERRAND, SAHRAOUI, GALTIER et al. 2022, par exemple, utilise les hypothèses d'isotropie ou d'axisymétrie de l'espace d'échelles. Dans le cas isotrope, l'espace des échelles est alors vu comme une sphère et 73 vecteurs directeurs partant du centre de cette sphère sont considérés. Dans le cas axisymétrique, le découpage est similaire, mais effectués dans des disques pour chaque ℓ_z . La divergence dans l'espace des échelles est ensuite effectuée sphériquement (resp. cylindriquement) le long de $\boldsymbol{\ell} = |\boldsymbol{\ell}|$ (resp. $\ell_{\perp} = \sqrt{\ell_x^2 + \ell_y^2}$) en supposant les dérivées angulaires nulles. Je n'ai pas voulu faire de même, n'étant pas convaincue de l'isotropie ou de l'axisymétrie de l'espace d'échelle. Une autre possibilité est de choisir les vecteurs en fonction du mode de représentation final. Si, au final, le mode de représentation est logarithmique, on peut ne choisir qu'un nombre limité de vecteurs à grande échelle tels qu'ils soient régulièrement espacés en représentation logarithmique. Un problème de cette méthode est l'irrégularité de la grille résultante. La divergence dans l'espace des échelles doit se faire vecteur par vecteur à partir des six échelles les plus proches (au minimum). Ce choix-là ne semblait toujours pas satisfaisant, car il implique de devoir potentiellement refaire le calcul en fonction du mode de représentation final et un biais en cas de moyenne

dans l'espace des échelles. Ces compromis doivent en plus être accompagnés d'une optimisation, voire d'une parallélisation du calcul numérique.

Après maintes versions et tentatives d'optimisation de mon code de post-traitement tel qu'essayant de respecter explicitement la forme de la loi exacte, j'ai décidé de changer radicalement de point de vue. Mathématiquement, les opérations de corrélation, \otimes , et de convolution, $*$, sont liés. En effet, si l'on considère deux quantités s et r , et qu'on les corrèle :

$$\begin{aligned} C_{sr}(\ell) = s(\mathbf{x}) \otimes r(\mathbf{x}) &= \frac{1}{V} \iiint_V s(\mathbf{x} + \ell) r^*(\mathbf{x}) d\mathbf{x} = \frac{1}{V} \iiint_V s(\mathbf{x}) r^*(\mathbf{x} - \ell) d\mathbf{x} \\ &= \frac{1}{V} \iiint_V s(\mathbf{x}) r^*(-(\ell - \mathbf{x})) d\mathbf{x} = \frac{1}{V} s(\ell) * r^*(-\ell) \end{aligned} \quad (\text{I.1.1})$$

avec V le volume d'intégration. Ainsi appliquer l'opération de corrélation sur s et r revient à convoluer s évaluée en ℓ avec le conjugué de r évaluée en $-\ell$. Dans notre situation, $r^* = r$ puisque r n'est pas un complexe. Une autre propriété mathématique intéressante est que l'opération de convolution correspond à effectuer un simple produit dans l'espace de Fourier. Ainsi en notant $\text{TF}[\cdot]$ la transformée de Fourier :

$$\text{TF}[C_{sr}(\ell)] = \frac{1}{V} \text{TF}[s(\ell)] \cdot \text{TF}[r(-\ell)] = \frac{1}{V} \text{TF}[s](\mathbf{k}) \cdot \text{TF}[r]^*(\mathbf{k}) \quad (\text{I.1.2})$$

L'obtention de l'ensemble de l'espace des échelles est donc possible mais demande de développer tous les termes factorisés de la loi exacte. Par exemple, pour la fonction de structure $\langle \delta \mathbf{v} \cdot \delta \mathbf{v} \delta \mathbf{v} \rangle$:

$$\langle \delta \mathbf{v} \cdot \delta \mathbf{v} \delta \mathbf{v} \rangle = \langle \mathbf{v}' \cdot \mathbf{v}' \mathbf{v}' - \mathbf{v} \cdot \mathbf{v} \mathbf{v} + \mathbf{v} \cdot \mathbf{v} \mathbf{v}' + 2\mathbf{v}' \cdot \mathbf{v} \mathbf{v} - \mathbf{v}' \cdot \mathbf{v}' \mathbf{v} - 2\mathbf{v}' \cdot \mathbf{v} \mathbf{v}' \rangle \quad (\text{I.1.3})$$

$$= \frac{1}{N} \text{TFI}[\text{TF}[C_{\mathbf{v},\mathbf{v},\mathbf{v}}] - \text{TF}[C_{\mathbf{v},\mathbf{v},\mathbf{v}}] + 2\text{TF}[C_{\mathbf{v},\mathbf{v}\mathbf{v}}] - 2\text{TF}[C_{\mathbf{v}\mathbf{v},\mathbf{v}}]] \quad (\text{I.1.4})$$

$$\begin{aligned} &= \frac{1}{N} \text{TFI}[\text{TF}[\mathbf{v}]^* \cdot \text{TF}[\mathbf{v} \cdot \mathbf{v}] - \text{TF}[\mathbf{v}] \cdot \text{TF}[\mathbf{v} \cdot \mathbf{v}]^*] \\ &\quad + \frac{2}{N} \text{TFI}[\text{TF}[\mathbf{v}]^* \cdot \text{TF}[\mathbf{v}\mathbf{v}] - \text{TF}[\mathbf{v}] \cdot \text{TF}[\mathbf{v}\mathbf{v}]^*] \end{aligned} \quad (\text{I.1.5})$$

avec $C_{\mathbf{v},\mathbf{v},\mathbf{v}} = \langle \mathbf{v}' \cdot \mathbf{v}' \mathbf{v}' \rangle$, $C_{\mathbf{v}\mathbf{v},\mathbf{v}} = \langle \mathbf{v}' \cdot \mathbf{v} \mathbf{v}' \rangle$, $\text{TFI}[\cdot]$ la transformée de Fourier inverse et N le nombre de points moyennés et sachant que par homogénéité statistique $\langle \mathbf{v}' \cdot \mathbf{v}' \mathbf{v}' \rangle = \langle \mathbf{v} \cdot \mathbf{v} \mathbf{v} \rangle$. Cette méthode est utilisable, car il est possible de développer les expressions en des produits de deux quantités générales, une évaluée au point \mathbf{x} et l'autre en \mathbf{x}' , et parce que la simulation est périodique. L'algorithme associé à cette méthode est schématisé sur la figure I.1.3 Cette méthode demande quelques précautions lors de l'implémentation, car elle peut vite devenir coûteuse en mémoire ainsi que de développer l'ensemble des termes présents dans une loi exacte. Cependant, elle permet d'obtenir un résultat complet, indépendant du mode de représentation final des résultats. C'est donc la méthode qui a été choisie. En usant de l'algorithme de FFT, elle s'avère, de plus, particulièrement rapide (moins d'une heure pour calculer séparément les trois termes de PP98).

1.3 Mode de représentation du résultat

Le résultat de l'algorithme de calcul par convolution est, pour chaque quantité, un parallélépipède couvrant une gamme d'échelles physiques dans la direction \mathbf{e}_z différente de

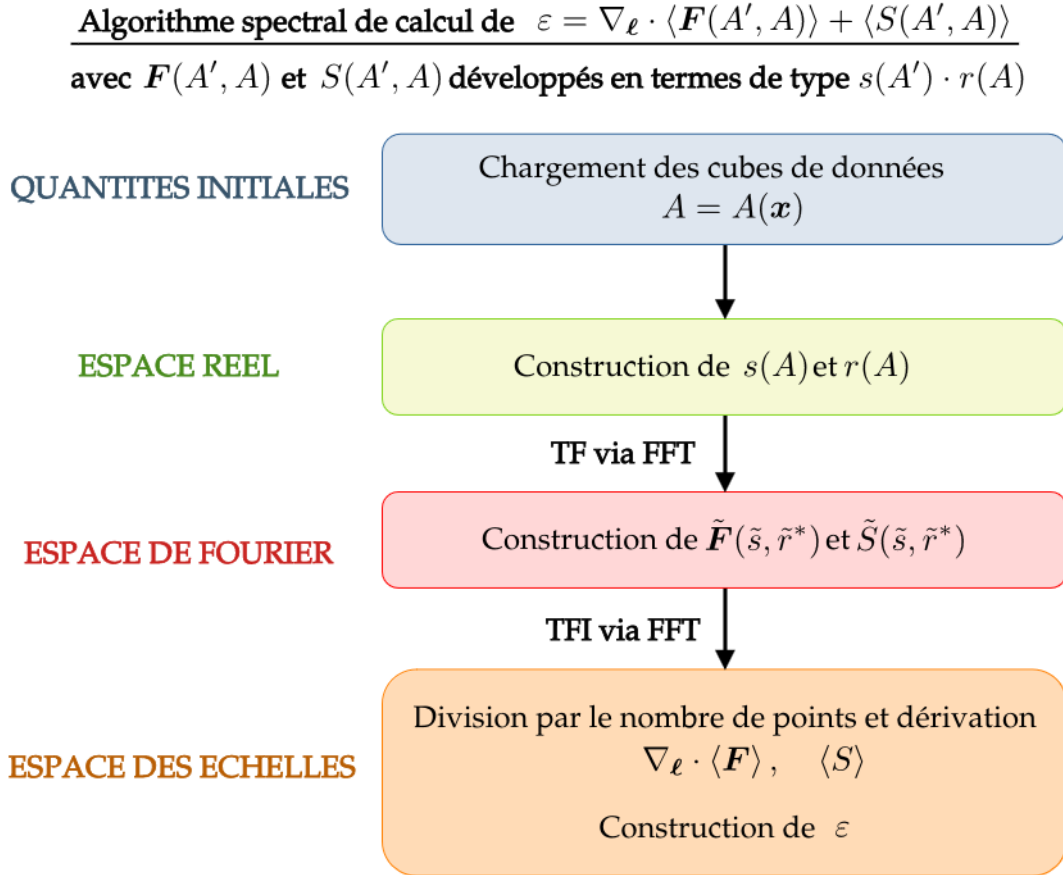


FIGURE I.1.3 – Algorithme de calcul du taux de cascade ε via la convolution. Les quantités impliquées sont des quantités génériques et par exemple pour S , $\text{TF}[S] = \tilde{S}$.

la gamme d'échelles couverte dans les directions perpendiculaires, \mathbf{e}_x ou \mathbf{e}_y . Ces gammes d'échelles couvrant différents ordres de grandeurs, une représentation logarithmique sera adoptée. Le caractère tridimensionnel de la grille parallélépipédique, impose de choisir une méthode de réduction (3D vers 2D ou 1D) afin de pouvoir visualiser facilement les quantités. Différents types de réductions sont possibles et illustrées sur la figure I.1.4 :

- R1 : 1D en fonction de $\ell = |\boldsymbol{\ell}|$ en moyennant la quantité sur des coquilles de rayon moyen ℓ ,
- R2 : 2D en fonction de $\ell_{\perp} = \sqrt{\ell_x^2 + \ell_y^2}$ et $\ell_{\parallel} = \ell_z$ en moyennant la quantité sur des couronnes de rayon moyen ℓ_{\perp} dans chaque plan perpendiculaire à \mathbf{e}_z ,
- R3 : 1D en fonction de ℓ_{\perp} en moyennant la quantité sur des coquilles cylindriques de rayon moyen ℓ_{\perp} , la moyenne suivant ℓ_{\parallel} peut s'effectuer de diverses manières qui seront détaillées par la suite,
- R4 : 1D en fonction de ℓ_{\parallel} en moyennant chaque plan perpendiculaire à \mathbf{e}_z , la moyenne suivant ℓ_{\perp} peut s'effectuer de diverses manières qui seront détaillées par la suite.

Sachant que la grille parallélépipédique couvre des gammes d'échelles différentes dans la direction \mathbf{e}_z et les directions perpendiculaires (voir I.1.4R2), notre géométrie est fondamentalement axisymétrique. La représentation de type R2 est donc la plus adaptée. L'échelle symlog permet de repérer les changements de signe et les ordres de grandeurs couverts par ε_{P98} tandis que les courbes de niveau linéaires révèlent les variations plus

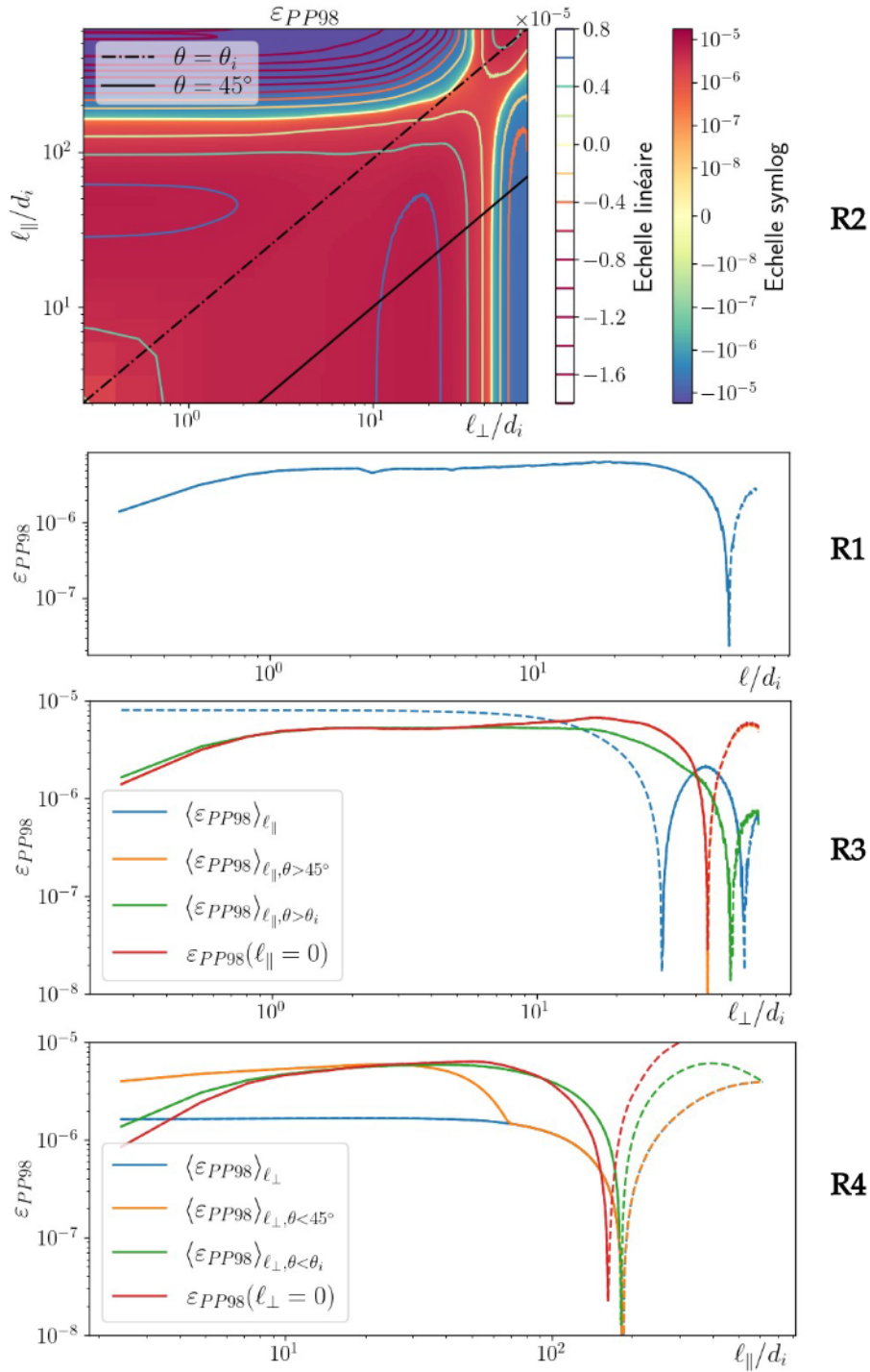


FIGURE I.1.4 – Différents modes de représentations du taux de cascade ε_{PP98} calculé avec PP98 dans les données de la simulation CGL1. R2 : 2D en fonction de ℓ_{\perp} et ℓ_{\parallel} , avec deux échelles de couleurs, une échelle symlog, linéaire entre -10^8 et 10^8 (continue), une échelle linéaire (discontinue). Frontières $\theta = \theta_i$ (noire discontinue) et $\theta = 45^\circ$ (noire continue). R1 : 1D en fonction de ℓ . R3 : 1D en fonction de ℓ_{\perp} , pour $\ell_{\parallel} = 0$ (rouge), moyenne sur l'ensemble des ℓ_{\parallel} (bleue), moyennes sur les ℓ_{\parallel} tels que $\theta > 45^\circ$ (orange), et tels que $\theta > \theta_i$ (vert). R4 : 1D en fonction de ℓ_{\parallel} , pour $\ell_{\perp} = 0$ (rouge), moyenne sur l'ensemble des ℓ_{\perp} (bleue), moyenne sur les ℓ_{\perp} tels que $\theta < 45^\circ$ (orange) et tels que $\theta < \theta_i$ (vert). Le caractère continu ou discontinu des courbes 1D reflète le signe de ε_{PP98} .

spécifiques tel qu'un affaiblissement aux petites échelles ou des bosses (courbes de niveau bleues) aux échelles intermédiaires. On pourrait définir une zone inertielle entre les courbes de niveaux associées à la valeur 0.4. On remarque que cette zone semble carrée, cela est dû aux axes logarithmiques. Avec des axes linéaires, on observerait un quart d'ellipse liant $\ell_{\parallel}/d_i = 10^2$ à $\ell_{\perp}/d_i \simeq 30$. Le problème des cartes est la difficulté de comparer de multiples quantités. Nous aurons donc besoin d'une représentation 1D.

R1 peut donner un résultat biaisé : sur I.1.4R1, le résultat correspond quasiment entièrement (sauf aux très grandes échelles communes aux directions parallèle et perpendiculaire) à $\varepsilon_{PP98}(\ell_{\parallel} = 0)$ en rouge sur I.1.4R3. De plus, le manque de points, pour effectuer la moyenne en chaque ℓ , induit des variations non-physiques du résultat (sursauts à intervalles réguliers sur I.1.4R1). Ce n'est donc pas le type de représentation 1D le plus adapté dans notre cas.

Originellement, les petites échelles sont définies telles que ℓ petit et les grandes échelles telles que ℓ grand. Cependant, visualiser la cascade via R3 en moyennant l'ensemble des ℓ_{\parallel} à ℓ_{\perp} fixé (courbe bleue), vient mixer les petits et grands ℓ . La zone négative à grand ℓ_{\parallel} vient écraser la zone inertielle présumée et plus encore la variation des petites échelles. Le même phénomène apparaît pour R4. Un autre possibilité serait de ne regarder qu'une direction $\ell_{\parallel} = 0$ pour R3 (en rouge) ou $\ell_{\perp} = 0$ pour R4. Le résultat n'est alors pas très lisse et peu représentatif de la variation d'ensemble. La troisième possibilité correspond à appliquer un filtre angulaire. En définissant θ , l'angle entre ℓ et \mathbf{e}_z , on pourrait considérer que les ℓ contribuant à la dynamique parallèle sont les ℓ quasi-parallèles tels que $\theta < 45^\circ$ et ceux contribuant à la dynamique perpendiculaire, les ℓ quasi-perpendiculaires tels que $\theta > 45^\circ$. La frontière $\theta = 45^\circ$ est représentée par une ligne noire continue sur la carte I.1.4R2 et les résultats apparaissent en orange sur I.1.4R3 et I.1.4R4. Pour R3, le résultat coïncide avec $\varepsilon_{PP98}(\ell_{\parallel} = 0)$. En effet, aux petites échelles, le plan tel que $\ell_{\parallel} = 0$ est la seule contribution à la moyenne. Similairement, pour R4, on assiste à un écroulement de la courbe qui rejoint $\langle \varepsilon_{PP98} \rangle_{\ell_{\perp}}$. Ce filtre angulaire n'est donc pas adapté.

La réduction 1D qui sera adoptée par la suite correspond à un filtrage angulaire basé sur l'angle d'injection de l'énergie, θ_i . Ce dernier impose la géométrie de la grille et les gammes d'échelles accessibles. Dans l'espace d'échelle, l'injection à lieux aux échelles telles que ℓ est maximal, c'est-à-dire dans l'angle supérieur droit de I.1.4R2. $\theta = \theta_i$ correspond à la diagonale représentée par une ligne noire discontinue sur I.1.4R2. Appliquer cette réduction nous donne les courbes vertes sur I.1.4R3 et I.1.4R4. N'y apparaissent, ni les artéfacts visibles avec R1, ni les saturations visibles sur les courbes bleues ou orange et elles sont plus représentatives du comportement de ε_{PP98} dans l'ensemble de l'espace des échelles que les courbes rouges.

1.4 Synthèse des méthodes et choix numériques

Code de simulation d'un plasma turbulent :

- Méthode d'intégration pseudo-spectrale (voir figure I.1.1) des équations fluides.
- Des termes correctifs d'hyperdissipation qui joueront le rôle de la dissipation aux petites échelles.
- Un forçage permanent maintenant l'énergie totale du système quasiment constante.
- Une géométrie périodique dépendant de l'angle d'injection de l'énergie, θ_i .

Je n'ai pas participé dans l'écriture de ce code mais je l'ai utilisé pour compléter le lot de simulations analysées par FERRAND, SAHRAOUI, LAVEDER et al. 2021.

Calcul des termes des lois exactes : Obtention rapide de l'ensemble de l'espace d'échelles accessible grâce à une méthode de calcul basée sur le lien entre corrélation et convolution, et sur la périodicité des simulations. L'algorithme est schématisé sur la figure I.1.3.

Visualisation des résultats : représentation cylindrique

- Représentation 2D en fonction de ℓ_{\parallel} et ℓ_{\perp} avec des échelles de couleurs de type chaud/froid (indiquant facilement le signe du résultat) associées aux variations logarithmique (fond) et linéaire (courbes de niveaux) du résultat.
- Représentation 1D en fonction de ℓ_{\perp} . Réduction du résultat 2D en moyennant sur ℓ_{\parallel} pour $\theta > \theta_i$.
- Représentation 1D en fonction de ℓ_{\parallel} . Réduction du résultat 2D en moyennant sur ℓ_{\perp} pour $\theta < \theta_i$.

avec θ l'angle entre ℓ et la direction moyenne du champ magnétique \mathbf{e}_z .

J'ai implémenté les codes de calcul et de visualisation des termes des lois exactes. Le code de calcul est disponible sur GitHub : *paulesimon97/2022-07_Simu_Exact_laws* : *Package usefull to compute exact laws* 2023.