

Petite note sur les isotopes dans LMDZ

Camille Risi

15 janvier 2021

Table des matières

1	Quelques bases sur les isotopes	1
2	Contraintes spécifiques aux isotopes	1
3	Stratégies de pérennisation	1
3.1	Stratégie 1 : mise à jour régulière d'une branche parallèle	2
3.2	Stratégie 2 : insertion dans le trunk	2
4	Méthodes d'implémentation	2
4.1	Méthode actuelle	2
4.2	Méthode des traceurs chimiques	3
4.3	Méthode de l'indice supplémentaire à l'eau normale	3
4.4	Exemple de méthode élégante : dyn3dmem	4
4.5	Autre exemple de méthode élégante : les thermiques	4
4.6	Un panaché ?	4

1 Quelques bases sur les isotopes

- Les isotopes de l'eau sont : $H_2^{16}O$, $H_2^{18}O$, HDO , HTO , $H_2^{17}O$. On peut aussi y ajouter les "traceurs d'eau", aussi appelés "water tags" ou "couleurs de l'eau" : ex : eau bleue évaporée sur les océans, eau verte évaporée sur continents, eau rouge venant de l'évaporation des pluies... On peut activer ou désactiver indépendamment chaque isotope.
- Ce sont des **traceurs passifs** qui n'agissent pas sur le climat ni sur l'eau normale.

2 Contraintes spécifiques aux isotopes

Ils doivent respecter 2 contraintes :

1. L'isotope $H_2^{16}O$ doit être à tout moment égal à l'eau normale q , les autres isotopes représentant une infime proportion de l'eau.
2. On étudie des rapports isotopiques : ex : HDO/q . Au premier ordre, tous les isotopes de l'eau se comportent de manière très similaire à q . Ainsi, les rapports isotopiques restent presque constants et on étudie leurs variations en ‰, notées δ (ex : $\delta D = (HDO/q/R_{ocean} - 1) \cdot 1000$). En conséquence, **la moindre petite différence dans le traitement d'un isotope par rapport à l'eau normale q engendre des erreurs énormes sur le δ .**

3 Stratégies de pérennisation

J'appelle "pérennisation" le fait que les isotopes de l'eau suivent les évolutions de LMDZ sur le long terme. Dans cette section, je distingue bien "pérennisation" et "insertion dans le trunk" : on peut avoir des isotopes dans le trunk sans pérennisation, et inversement des isotopes pérennes dans une branche parallèle au trunk.

3.1 Stratégie 1 : mise à jour régulière d'une branche parallèle

C'est plus ou moins la stratégie utilisée depuis 14 ans. Les isotopes sont dans une version "informelle" de LMDZ. Régulièrement (tous les 1 à 3 ans), je transfère les isotopes dans des versions plus récentes du code. Une **batterie de tests** sont lancés. Le **code est quadrillé de vérifications**, correspondant aux 2 contraintes listées en section 2. Dès qu'un nouveau développement de la physique est incompatible avec un ancien code isotopique, la simulation plante, renvoyant exactement l'endroit du plantage. Il y a alors un travail de mise à jour du code isotopique pour re-devenir compatible avec la physique. Ca représente environ 1 mois de travail par an.

Avantages :

- Bon an, mal an, cette stratégie "artisanale" a permis la maintenance de LMDZ-iso **pendant 14 ans**, de LMDZ4 à LMDZ6 en passant par LMDZ5A et LMDZ5B. Ca a donc permis de fait une sorte de "pérennisation".
- Elle pourrait être améliorée avec une **automatisation de la batterie de tests**, qui pourraient être lancés de manière plus routinière, par exemple grâce à un support ingénieur permanent.
- La création d'un sous-répertoire physiso/ au côtés de phylmd/ permettrait de rendre la version récupérable publiquement sous svn.
- Cette stratégie permet de la flexibilité sur la façon de coder. Il n'y a pas besoin de se conformer aux normes de codage de LMDZ.

Inconvénients

- Pour l'instant, pas récupérable sous svn.
- La mise à jour prend beaucoup de temps, surtout quand on attend plusieurs années.
- Je peux devenir un point bloquant si je ne mets plus à jour.

3.2 Stratégie 2 : insertion dans le trunk

C'est par exemple le cas des traceurs chimiques dans LMDZ.

C'est aussi maintenant le cas de tous les traceurs, même isotopiques, dans dyn3dmem.

Avantages :

- Si c'est bien fait, certaines parties de la physique pourraient être modifiées sans que les isotopes ne soient affectés. Les isotopes suivraient alors naturellement les évolutions du code. Dans dy3dmem, par exemple, il est probable que le transport des traceurs et d'isotopes survivent à n'importe quelle évolution (voir sous-section 4.4).

Inconvénients :

- Moins de flexibilité dans la façon de coder les isotopes. Réflexion nécessaire sur les méthodes d'implémentation (voir section 4) pour se conformer aux normes de codage de LMDZ.
- Certaines modifications futures de la physique affecteront les isotopes, donc le travail de mise à jour du code isotopique pour re-devenir compatible avec la physique restera entier. Autrement dit, **la question du travail de pérennisation reste entière**. Idéalement, la méthode d'implémentation (voir section 4) doit permettre que ce travail soit le plus facile possible.

4 Méthodes d'implémentation

4.1 Méthode actuelle

Appelons "variable d'eau" toute variable qui représente de l'eau active dans le cycle de l'eau : ex : la concentration en vapeur d'eau, contenu en eau liquide ou glacée, eau du sol, précipitations, neige, etc.

Partout où il y a une variable d'eau $q(\text{grid}, n\text{lay})$, on **rajoute une variable isotopique équivalente** $x(n\text{grid}, n\text{lay}, n\text{traciso})$. $n\text{traciso}$ est le nombre d'isotopes et traceurs d'eau tels que définis dans infotrac. Il y a un tableau d'indices qui permet de savoir quel isotope est quoi, et donc quel isotope a quelles propriétés physiques (fractionnement, diffusivité, radioactivité éventuelle...).

Dans chaque routine, à chaque fois qu'il y a une "variable d'eau" dans la liste des arguments, on ajoute son équivalent isotopique dans la liste.

Les routines concernées sont nombreuses : évaporation totale de l'eau en entrée de physique, évaporation de surface, diffusion turbulente, thermiques, convection, poches, condensation de grande échelle, etc.

A chaque fois qu'une variable d'eau est modifiée, on modifie son équivalent isotopique avec une **boucle** $\text{do } i\text{xt}=1, n\text{traciso}$.

Quand il y a des fractionnements isotopiques, ou de la décroissance radioactive, ou autres processus spécifiques aux isotopes, alors on appelle une routine purement isotopique pour modifier la variable isotopique. Ainsi, dans les

routines habituelles de la physique, on ne voit jamais le code des processus isotopiques. Ils sont “cachés” dans les **modules isotopiques**.

Tout le code isotopique est encadré par une **clé CPP de pré-compilation**, `#ifndef ISO`. Ainsi, on peut compiler le code sans isotopes.

Le code est **quadrillé par des routines de vérifications** isotopiques, qui sont indispensables pour la pérennisation du code (voir section 3). Ces appels de routines pourraient être rendus moins encombrants en les encapsulant dans des routines génériques de vérification isotopiques (D Cugnet avait commencé à y réfléchir). Ces vérifications sont activées par une clé de précompilation, `#ifndef ISOVERIF`, et sont désactivées en mode production.

Avantages :

- Permet de suivre l’eau au plus près, et donc facilite le traitement des isotopes exactement comme l’eau normale
- En cas d’évolution de la physique incompatible avec le code isotopique, les routines de vérifications permettent de retrouver très vite exactement les lignes problématiques. Ça facilite donc le travail de pérennisation.

Inconvénients

- Encombrant

4.2 Méthode des traceurs chimiques

Les traceurs chimiques sont actuellement concaténés dans une variable `tr(ngrid,nlay,nqtot)`.

Dans la routines où il y a du transport ou des changements de phase, des variables qui diagnostiquent le transport et les changements de phase sont calculées et passées en arguments de sortie. Ces variables sont ensuite passées en entrée de **phytrac**. Dans `phytrac`, les champs de traceurs chimiques sont modifiés en n’utilisant que les variables diagnostiquées.

Avantages :

- Traceurs pas du tout encombrants car ils n’entrent jamais dans les routines habituelles de la physique.
- Pour les traceurs chimiques, ça permet de continuer à fonctionner même en cas d’évolution du code : ex : actuellement, le transport des traceurs par les poches froides n’a pas encore été implémenté, mais ça n’empêche qu’on peut quand même continuer à transporter les traceurs par les autres processus convectifs. Pour les traceurs chimiques, ce n’est pas très grave si le transport n’est pas complètement cohérent avec celui de l’eau.

Inconvénients

- Les isotopes doivent être traités exactement comme l’eau normale (voir contraintes en section 2). Or le transport de l’eau et ses changements de phase sont nombreux et souvent intriqués, comme par exemple dans la convection. Les effets sur l’eau ne s’ajoutent pas forcément de manière linéaire, car il y a par exemple des effets de seuil. L’ordre des processus a une importance. Mettre les isotopes dans `phytrac` demanderait de sortir des dizaines de variables diagnostiques par routines, pour mémoriser les différentes étapes des évolutions de l’eau.
- Externaliser les isotopes dans `phytrac` rendrait plus difficile le débogage. On ne pourrait plus quadriller le code de vérifications comme actuellement, à moins de mémoriser chaque étape de modification de l’eau.
- Externaliser les isotopes dans `phytrac` reviendrait pour la convection ou la condensation de grande échelle à pratiquement dupliquer les codes. Cela rendrait d’autant plus difficile la mise à jour du code isotopique par rapport aux évolutions de la physique. En terme de pérennisation, l’avantage par rapport à un simple ajout d’un dossier `physio/` aux côtés de `phylmd/` n’est pas évident.

4.3 Méthode de l’indice supplémentaire à l’eau normale

Certains groupes de modélisation isotopique ont choisi d’ajouter un indice à l’eau normale : `q(grid,nlay,ntraciso)`. C’est aussi le cas maintenant dans `dyn3dmem`.

Avantages :

- Ça permet de passer en argument les isotopes avec l’eau normale de manière transparente.
- Les isotopes et l’eau normale se retrouvent dans les mêmes boucles, ce qui permet d’assurer que toute modification du code sur l’eau normale concerne aussi les isotopes.

Inconvénients

- Ça nécessite de gros changements de code.
- Ça demanderait peut-être de repenser les traceurs chimiques aussi.
- Il y a plein d’endroits où on utilise l’eau normale sans avoir besoin de ses isotopes : ex : pour calculer le dégagement de chaleur latente.
- Ça n’enlèvera pas les appels aux routines spécifiquement isotopiques lors des fractionnements, qui peuvent encore être jugés encombrants.

4.4 Exemple de méthode élégante : dyn3dmem

Tous les traceurs (eau, isotopes, traceurs chimiques) sont définis dans `traceur.def`. On en fait un tableau unique de type `q(grid,nlay,nqtot)`, contenant à la fois l'eau normale, les isotopes de l'eau et les traceurs chimiques.

Pour chaque traceur, on définit un "père", qui est le fluide transporteur. Par exemple, l'eau normale ou les traceurs chimiques sont "fils" de l'air, c'est à dire transportés par l'air. Les isotopes sont "fils" de l'eau, c'est à dire qu'ils sont transportés par l'eau. De même, n'importe quel traceur chimique pourrait théoriquement avoir des "fils", ouvrant la voie dans le futur au transport d'isotopes de traceurs chimiques. Les couleurs d'isotopes sont des "fils" d'isotopes.

Une fois les vents connus, on appelle la routine d'advection sur tous les "fils" directs de l'air. Puis à chaque fois qu'on advecte un fils, **on appelle de manière récursive la routine d'advection sur tous les fils** de ce fils. Et ainsi de suite. Ainsi, l'air transporte l'eau et les traceurs chimiques, et l'eau transporte ses isotopes.

Avantages :

- les isotopes sont transparents, traités de manière générique aux côtés de l'eau normale et des traceurs chimiques.
- Même si `dyn3dmem` évolue, il est probable qu'aucun travail de mise à jour des isotopes ne soit nécessaire.
- C'est maintenant dans le trunk.

Comment s'inspirer de ce succès pour les routines de la physique ?

- Difficulté : dans la physique, les choses sont plus complexes et les phénomènes de transport sont moins bien isolés du reste.

4.5 Autre exemple de méthode élégante : les thermiques

Dans le schéma des thermiques, le transport vertical de l'eau a été isolé du reste, par le biais de la routine `thermcell_dq`. Cette routine générique permet de **transporter n'importe quel traceur, dont l'eau, de manière générique**. Cette même routine est appelée dans `phytrac` pour le transport des traceurs chimiques. Pour les isotopes de l'eau, il suffit donc aussi d'appeler cette même routine en l'insérant dans une boucle `do ixt=1,ntraciso` et en entrant en argument chaque profil isotopique.

Avantages :

- Peu encombrant
- Satisfait la "conceptualisation" réclamée par Frédéric.

Peut-on généraliser cette méthode aux autres routines ?

- Pour les thermiques, ça marche bien car Frédéric a fait lui-même pour l'eau normale l'effort de "conceptualisation" qu'il réclame. Il a probablement pensé aux transports des traceurs bien en amont du codage. De plus, les changements de phase ont été exclus des thermiques et ont été "délégués" au schéma de condensation de grande échelle. Il n'y a donc pas le problème d'imbrication des changements de phase et du transport qu'on trouve dans la convection.
- Dans la convection, pour faire le même travail, il faudrait d'abord faire le même travail pour l'eau normale. Est-il envisageable de sortir le transport de l'eau du schéma de convection ? Ou est-ce que les changements de phase et le transport sont vraiment trop intriqués ? Je pense qu'il n'y a pas de raisons d'externaliser les isotopes tant que l'eau elle-même n'est pas externalisée.

4.6 Un panaché ?

Ces différentes méthodes ne sont pas forcément exclusives, et pourraient être panachées selon les routines, sachant qu'il y a des routines :

- "faciles", où les isotopes suivent exactement l'eau normale
- "difficiles", où il y a des changements de phase : en particulier convection, condensation de grande échelle, évaporation de surface, eau du sol, neige.