



IAM: Impact Assessment Model for fisheries management

Mathieu Merzereaud¹, Claire Macher¹, Michel Bertignac², Marjolaine Fresard³, Olivier Guyader¹, Christelle Le Grand¹, Sophie Gourguet¹, Florence Briton¹, Maxime Jaunatre¹

¹ IFREMER | UMR AMURE, RBE/ Unité d'Economie Maritime

² IFREMER | RBE/ Unité Sciences et Technologies Halieutiques

³ UBO | UMR AMURE

13 mai 2022



Table des matières

1	Structure et mode opératoire du modèle	5
1.1	Notations et structure schématique du modèle	5
1.1.1	Notations utilisées	5
1.1.2	Schéma structurel du modèle bio-économique	6
1.2	Description et déroulement des modules	6
1.2.1	Module "Mortalité par pêche et Survie des rejets"	6
1.2.1.1	Sous-module "Allocation de la mortalité"	8
1.2.1.2	Sous-module "Calcul de capturabilité"	11
1.2.1.3	Module "Mortalité par pêche"	12
1.2.2	Module "Dynamique de populations"	16
1.2.2.1	Module principal	16
1.2.2.2	Module "Recrutement"	19
1.2.3	Module "Captures, débarquements et rejets"	19
1.2.4	Module "Marché"	24
1.2.5	Module "Economique"	26
1.2.6	Module "Scénario"	38
1.2.7	Module "Gestion"	39
1.2.8	Module "Réplicat"	39
2	Elaboration d'un fichier de paramétrage fonctionnel	42
2.1	Feuillet "Stock"	42
2.2	Feuillet "Market"	46
2.3	Feuillet "Flottille"	46
2.4	Feuillet "fm_matrix"	47
2.5	Feuillet "mm_matrix"	47
2.6	Feuillet "icat_matrix"	47
2.7	Feuillet "Scénarii"	48
2.8	Feuillet "Stochasticité"	49
2.8.1	Table "Samples"	50
2.8.2	Table "Random-variate"	50
3	Vignettes	52
3.1	Utilisation d'IAM dans R	54
3.1.1	Choix d'implémentation	54
3.1.2	Le package IAM (version 2.0.0)	54
3.1.2.1	Installation et chargement du package	54
3.1.2.2	Importation des paramètres (IAM.input)	55
3.1.2.3	Constitution des arguments (IAM.args)	57

3.1.2.4	Simulation (IAM.model)	60
3.1.2.5	Mise en forme des sorties (IAM.format & IAM.format_quant) .	62
3.1.3	Représentations graphiques des sorties avec ggplot2	64
3.1.4	Conseil de réplcation	66
3.1.5	Références	67
3.2	Gestion avec TAC	70
3.2.1	Scénario <i>statu quo</i>	71
3.2.2	Gestion TAC générale	72
3.2.2.1	Représentation graphiques	73
3.2.2.2	Pondérer les efforts avec la matrice mfm	76
3.3	Scénarios TAC, quotas et transition F_{msy}	80
3.3.1	Scénario <i>statu quo</i>	81
3.3.2	Gestion TAC générale	82
3.3.2.1	Représentation graphiques	83
3.3.3	TAC jusqu'au F_{msy}	86
3.3.3.1	Représentation graphiques	86
3.3.3.2	Calcul de la probabilité d'atteindre le F_{msy}	86

Introduction

IAM (Impact Assessment Model) est un modèle bio-économique de simulation de dynamiques de pêcheries, intégrant des outils spécifiques d'aide à la décision dans le cadre de mises en application théoriques de mesures de gestion. Ce document constitue un support d'utilisation du modèle, décrivant les étapes de paramétrage et de lancement des simulations, exposant les méthodologies et équations fonctionnelles utilisées en arrière-plan, et analysant les possibilités offertes par l'outil. Une première partie principalement théorique décrira l'architecture modulaire du modèle, détaillera les paramètres mis en jeu ainsi que les liens fonctionnels les unissant. Les différents outils de simulation greffés au modèle "basique" et les méthodologies associées seront également présentés dans cette partie. Plus techniquement, la méthode de constitution du fichier de paramétrage fera l'objet alors qu'une troisième et dernière partie reprendra l'ensemble des vignettes associées au package décrivant la mise en application des simulations au sein d'un environnement R.

Chapitre 1

Structure et mode opératoire du modèle

1.1 Notations et structure schématique du modèle

1.1.1 Notations utilisées

IAM est un modèle bio-économique à temps discret, multi-flottille, multi-métier, et multi-spécifique à composantes "âge" pour la partie biologique, et à composantes "catégorie commerciale" pour la partie économique. Les paramètres du modèle peuvent ainsi se décliner selon 7 indices de définition distincts qui sont décrits dans le tableau ci-dessous :

Indice	Description
t	Indice temporel
f	Indice flottille
m_{bio}	Indice métier (paramètres biologiques)
m_{eco}	Indice métier (paramètres économiques)
e	Indice espèce
ie	Indice âge (dépend de l'espèce)
ce	Indice catégorie (dépend de l'espèce)

TABLE 1.1 – Indices de déclinaison des variables numériques

En pratique, ces indices vont également déterminer la structuration des objets R incarnant les inputs du modèle (voir chapitre 3). Ces objets se présenteront en effet sous la forme de matrices multidimensionnelles formatées, incluant toute l'information nécessaire au modèle pour à la fois procéder aux calculs des indicateurs de sortie, mais aussi pour assurer la robustesse des implémentations. De surcroît, cette mise en forme spécifique, commune aux données d'entrée et de sortie, facilitera les traitements numériques ultérieurs qui pourront leur être appliqués. Notons que l'on peut distinguer deux types d'indices "métier", puisque le modèle prend en considération la possibilité de définir les paramètres biologiques et les paramètres économiques selon deux niveaux "métier" distincts. Dans ce cas, une matrice de correspondance entre ces deux niveaux de définition sera requise (matrice de dimension $m_{bio} \cdot m_{eco}$, voir chapitre 2). Précisons également que plusieurs types de dynamiques de population pourront être appliquées à un stock donné ; elles sont pour le moment au nombre de 3, et sont les suivantes :

- dynamique de population de type XSA (par défaut, annuelle et définie aux âges. Ex : Sole 8ab)
- dynamique de population de type SS3 (trimestrielle et définie par âge et cohorte. Ex : Merlu 8ab)

- dynamique de population de type SPiCT (modèle Pella-Tomlinson annuel)

On peut rajouter à ces trois options la possibilité de ne pas considérer de dynamique de population. Les productions seront alors calculées sur la base de débarquements par unité d'effort, considérés ou non comme constants au cours du temps. La mise en œuvre de chacune de ces dynamiques nécessitera un ensemble de paramètres spécifiques.

1.1.2 Schéma structurel du modèle bio-économique

Le schéma de la page suivante décrit de manière synthétique et simplifiée le mode de fonctionnement du modèle bio-économique. Il met à la fois en évidence la structure modulaire du modèle (chaque module étant représenté par un rectangle bleu), et les interactions entre ces modules au travers des relations existant entre les paramètres mis en jeu. Il permet aussi de distinguer les modules "utilisateur" (rectangles bleus avec intitulés en couleur), qui serviront à orchestrer l'exécution des modules de simulation tout en permettant à l'utilisateur d'intervenir au cours du processus de simulation, des modules "passifs" (rectangle bleus avec intitulés en noir) qui seront tributaires de l'action des précédents et qui composeront l'ossature du modèle. La dynamique de population ici considérée et illustrée est le modèle XSA. Pour plus de précisions concernant les interactions mises en jeu au sein des autres modèles de population, le lecteur pourra se référer aux équations détaillées de la partie 1.2.

L'objectif des sections qui vont suivre est de proposer une description exhaustive et organisée des processus mis en action durant l'étape de simulation. Il s'agira donc de fournir à la fois une synthèse des calculs effectués, mais aussi une vue des méthodologies utilisées et des différents outils mis à la disposition de l'utilisateur.

1.2 Description et déroulement des modules

La partie qui suit va s'attarder plus longuement sur les modules constituant le modèle bio-économique. Dans ce contexte, les paramètres d'entrée du module seront décrits (ils pourront tout aussi bien être des paramètres d'entrée du modèle que des paramètres de sortie d'autres modules), les sorties le seront également, et les équations décrivant le cheminement des calculs seront présentées.

Notons que les paramètres initiaux et les variables calculées sont décrits ici à leur niveau maximal de précision. Il est bien évidemment concevable que certaines données ne soient pas disponibles à ce niveau de définition : l'ajustement nécessaire est généralement pris en compte dans l'implémentation (tests sur la compatibilité des dimensions entre variables, et correction du format si besoin est).

Précisons pour finir que la dimension temporelle le plus souvent implicite dans les formulations suivantes peut être non seulement intégrée au stade initial de déclaration des paramètres d'entrées, mais également greffée à toute variable du modèle par le biais du module "scénario" permettant l'intervention de l'utilisateur à n'importe quel instant t .

1.2.1 Module "Mortalité par pêche et Survie des rejets"

Ce module fait appel à deux sous-modules. Le premier procède à la ventilation de la donnée "mortalité" initiale disponible par une variable auxiliaire supposée corrélée afin d'affiner le niveau de définition accessible. Ce processus s'effectue le plus souvent au moyen d'une variable de type "captures", ou à défaut, d'une donnée "débarquements". Le deuxième sous-module

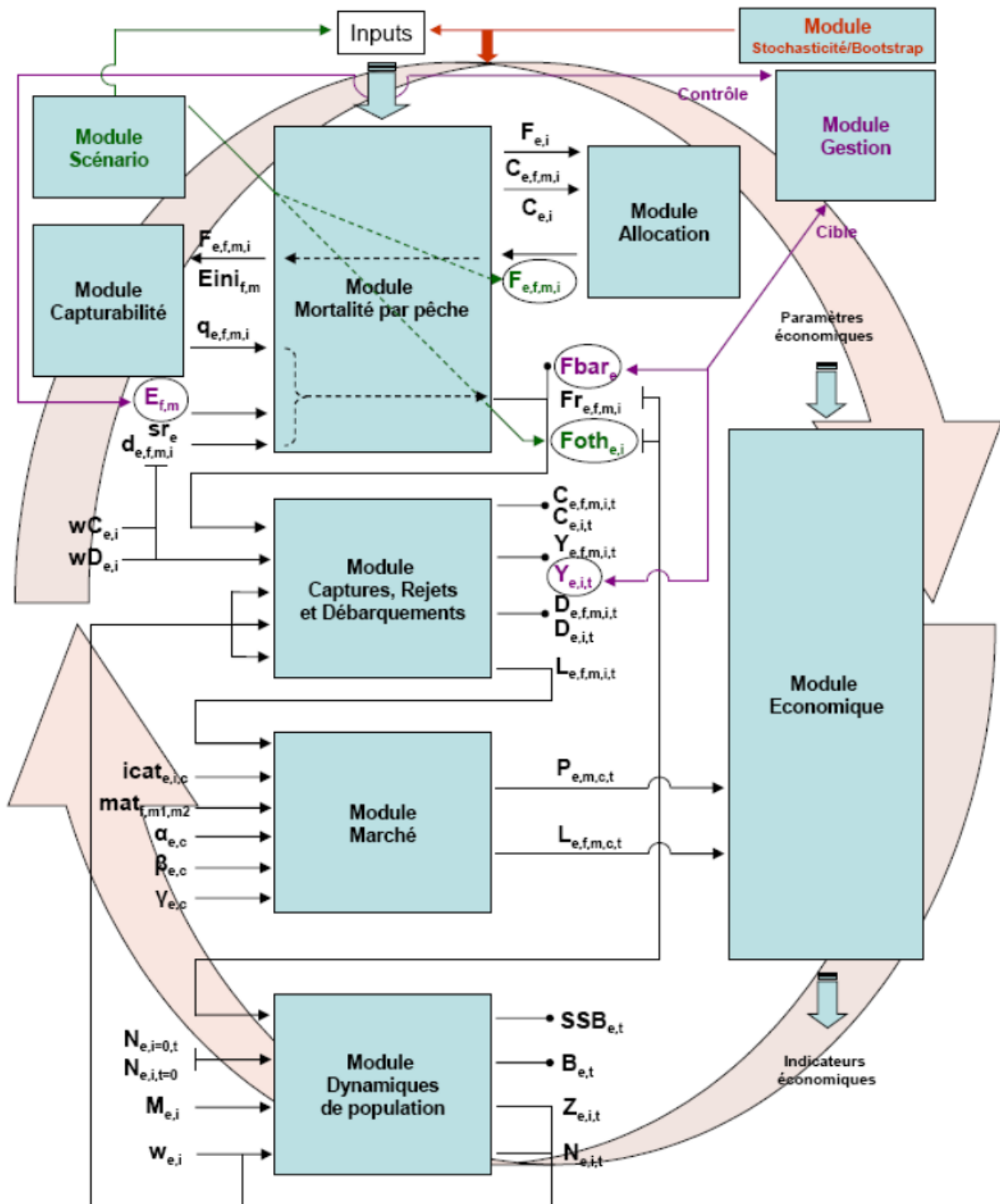


FIGURE 1.1 – Schéma structurel simplifié du modèle bio-économique

va estimer initialement une capturabilité associée afin de l'appliquer au cours du processus de modélisation à une variable de contrôle "effort".

1.2.1.1 Sous-module "Allocation de la mortalité"

Avant tout, précisons que ce qui va se rapporter à cette étape d'allocation ne concerne que les dynamiques **XSA** et **SPiCT**. La mortalité par pêche pour la dynamique SS3 est d'entrée renseignée déjà ventilée au niveau flottille-métier-âge-cohorte-saison requis, aucun travail d'allocation n'est donc nécessaire. Le modèle statique quant à lui s'affranchit de cet indicateur puisqu'il se base sur des données de LPUE déjà disponibles au niveau flottille-métier.

L'objectif de ce module est d'aboutir à un niveau de définition maximal du paramètre de mortalité, à savoir idéalement une valeur par espèce, âge, flottille et métier. Il existe plusieurs manières d'obtenir ce niveau maximal (voir encadré 1.2.1.1 décrivant les procédés de ventilation implémentés). En partant d'une donnée "Mortalité" plus précise, on peut se contenter d'une variable de ventilation plus grossière. On peut également procéder en deux étapes successives avec deux variables distinctes si la variable initiale ne décline que la dimension "âge", et qu'il reste donc à intégrer les dimensions "métier", puis "flottille". Toutes ces possibilités sont offertes par le module implémenté, et celui-ci considérera au mieux les ventilations à opérer le cas échéant.

Si nécessaire, une première ventilation sera effectuée au moyen des variables C_{mi} et C_i préalablement renseignées dans les onglets "Espèce" du fichier de paramétrage (voir chapitre 2) afin d'intégrer au minimum la dimension "métier" à la donnée de mortalité par pêche. On rappelle qu'il s'agira idéalement de données de captures en nombre, mais qu'une autre donnée pourra à défaut être utilisée. Si à ce stade, la mortalité possède le niveau complet requis, la procédure s'arrêtera. Sinon, la seconde ventilation s'opérera avec les variables Y_{mi} et L_{refe} (variable présente dans les onglets de paramétrage économique et appliquée ici à la matrice "fm" d'allocation flottille-métier) afin d'inclure la dimension "flottille". Il revient à l'utilisateur de s'assurer de la pertinence des données assignées à chacune de ces variables afin que le processus de ventilation aboutisse à des résultats justes et optimaux (cf encadré 1.2.1.1 de la page suivante).

Note : la ventilation ne s'effectue plus en tant qu'étape préliminaire aux simulations lors du lancement du modèle, mais désormais lors de la phase d'intégration des indicateurs visant à créer l'objet R de paramétrage. Le paramètre calculé F_{fmi} apparaît donc maintenant parmi les paramètres "sources" au sein de l'objet, et est au même titre qu'eux directement utilisé par le modèle.

Procédures de ventilation : ventilation d'une variable Mortalité (F) par une variable Capture (C)

On décrit ici les procédures permettant de redéfinir le niveau d'agrégation d'une variable donnée en prenant appui sur une donnée externe corrélée, de niveau de définition supérieur ou complémentaire. On prendra pour cela l'exemple d'une variable de mortalité par pêche, qu'on ventilera à l'aide d'une variable de captures totales. Ces méthodes d'application porteront sur différentes combinaisons de dimensions. Les dimensions illustrées seront d'indice "f" (flottille), "m" (métier) et "i" (âge).

- Indice(s) en commun

Variable à ventiler (F)	Variable de ventilation (C)	Variable complémentaire requise (C _{tot})	Résultat (F _{ventil})
F_i	$C_{m,i}$	C_{tot_i}	$F_{m,i}$
F_i	$C_{f,m,i}$	C_{tot_i}	$F_{f,m,i}$
$F_{m,i}$	$C_{f,m}$	C_{tot_m}	$F_{f,m,i}$
$F_{m,i}$	$C_{f,m,i}$	$C_{tot_{m,i}}$	$F_{f,m,i}$

- Pas d'indices en commun

F_i	C_m	C_{tot}	$F_{m,i}$
F_i	$C_{f,m}$	C_{tot}	$F_{f,m,i}$
$F_{m,i}$	C_f	C_{tot}	$F_{f,m,i}$

Equation :

$$F_{ventil} = \frac{F \times C}{C_{apt_{tot}}} \text{ sur tous les indices en présence}$$

Méthodologie appliquée dans la procédure de ventilation

• **Paramètres initiaux** (la description des paramètres d'entrée et l'équation de calcul associée dépeignant ici la procédure d'allocation ne doivent être considérées que comme un cas particulier d'une ventilation en une étape)

- XSA, SPiCT

Notation	Description	Source
$F_{e,i}$	Coefficient de mortalité par pêche initial (ici, par espèce et par âge)	Stock Assessment ($i = \{all\}$ pour modèle global SpiCT, et $\{i\} = \{0, 1, \dots\}$ pour modèle aux âges XSA)
$C_{e,f,m,i}$	Variable de ventilation à un niveau de définition requis (ici, captures par espèce, flottille, métier et âge)	SACROIS

$C_{tot_{e,i}}$	Variable de ventilation totale (doit être définie sur l'intersection des niveaux des deux variables précédentes)	SACROIS
-----------------	--	---------

TABLE 1.2: Paramètres initiaux XSA, SPiCT pour le sous-module "allocation de la mortalité par pêche"

• **SS3** (toutes les variables décrites ci-dessous comprennent chacune 4x4 incarnations correspondants aux combinaisons "cohorte / saison"; par souci d'identification, les noms de ces variables seront indexées M1S1, M1S2, ..., M4S3, M4S4) :

Notation	Description	Source
$iniF_{q_{e,i}}$	Coefficient de mortalité par pêche total initial (par espèce, âge, saison et cohorte)	Stock Assessment
$F_{q_{e,i}}$	Coefficient de mortalité par pêche total à $t = 1$ (par espèce, âge, saison et cohorte)	Stock Assessment
$iniF_{q_{e,i}}$	Coefficient de mortalité par pêche total initial (par espèce, flottille, métier, âge, saison et cohorte)	Stock Assessment
$F_{q_{e,f,m,i}}$	Coefficient de mortalité par pêche total à $t = 1$ (par espèce, flottille, métier, âge, saison et cohorte)	Stock Assessment
$iniF_{qLwt_{e,i}}$	Coefficient de mortalité par pêche "poids débarqués" initial (par espèce, âge, saison et cohorte)	Stock Assessment
$F_{qLwt_{e,i}}$	Coefficient de mortalité par pêche "poids débarqués" à $t = 1$ (par espèce, âge, saison et cohorte)	Stock Assessment
$iniF_{qLwt_{e,f,m,i}}$	Coefficient de mortalité par pêche "poids débarqués" initial (par espèce, flottille, métier, âge, saison et cohorte)	Stock Assessment
$F_{qLwt_{e,f,m,i}}$	Coefficient de mortalité par pêche "poids débarqués" à $t = 1$ (par espèce, flottille, métier, âge, saison et cohorte)	Stock Assessment
$iniF_{qDwt_{e,i}}$	Coefficient de mortalité par pêche "poids rejetés" initial (par espèce, âge, saison et cohorte)	Stock Assessment
$F_{qDwt_{e,i}}$	Coefficient de mortalité par pêche "poids rejetés" à $t = 1$ (par espèce, âge, saison et cohorte)	Stock Assessment

$iniFqDwt_{e,f,m,i}$	Coefficient de mortalité par pêche "poids rejetés" initial (par espèce, flottille, métier, âge, saison et cohorte)	Stock Assessment
$FqDwt_{e,f,m,i}$	Coefficient de mortalité par pêche "poids rejetés" à $t = 1$ (par espèce, flottille, métier, âge, saison et cohorte)	Stock Assessment

TABLE 1.3: Paramètres initiaux SS3 pour le sous-module "allocation de la mortalité par pêche"

• Variables calculées (XSA et SPiCT)

Notation	Type	Description	Equation
$F_{e,f,m,i}$	Sortie	Coefficient instantané de mortalité par pêche ventilé initial (ici, par espèce, flottille, métier et âge)	$F_{e,f,m,i} = \frac{F_{e,i} \cdot C_{e,f,m,i}}{C_{tot_{e,i}}}$
$Foth_{e,i}$	Sortie	Mortalité par pêche initiale "autres flottilles" par espèce et âge. Entrée des modules <i>Captures</i> et <i>Dynamiques de populations</i>	$Foth_{e,i} = F_{e,i} - \sum_{f,m} F_{e,f,m,i}$

TABLE 1.4: Paramètres calculés pour le sous-module "allocation de la mortalité par pêche"

1.2.1.2 Sous-module "Calcul de capturabilité"

On utilise ici le terme "capturabilité" de façon générique pour désigner la mortalité par pêche ramenée à une unité d'effort donnée. Cet effort sera le produit de deux variables d'entrée effort1 et effort2, auxquelles on pourra affecter différents couples d'indicateurs (voir tableau 1.5 ci-dessous). La mortalité par pêche devient ainsi une fonction de la variable d'effort au cours du processus implémenté.

• Paramètres initiaux (XSA, SPiCT,SS3)

Notation	Description	Source
$effort1_{f,m}$	Première composante d'effort (ici au niveau flottille-métier), soit le nombre de marées moyen par an et par navire ($nbTrip$) soit le nombre de jours de mer moyen par an et par navire (nbd s).	SACROIS
$effort2_{f,m}$	Deuxième composante d'effort (ici au niveau flottille-métier), soit la durée moyenne des marées par an et par navire ($tripLgth$) soit l'élément neutre 1	SACROIS

TABLE 1.5: Paramètres initiaux SS3 pour le sous-module "calcul de capturabilité"

• Variables calculées (XSA et SPiCT, SS3 sur les variables mortalités)

Notation	Type	Description	Equation
$Q_{e,f,m,i}$	Sortie	Capturabilité initiale (par espèce, flottille, métier et âge)	$Q_{e,f,m,i} = \frac{F_{e,f,mi}}{effort1_{f,m} \times effort2_{f,m}}$

TABLE 1.6: Paramètres calculés pour le sous-module "calcul de capturabilité"

1.2.1.3 Module "Mortalité par pêche"

Ce module intègre la méthode de calcul des capturabilités décrite précédemment. Il va en outre évaluer les mortalités par pêche corrigées. Le paramétrage systématisant dorénavant pour une flottille donnée la description exhaustive des espèces pêchées et des métiers pratiqués (via une espèce "ZZZ" considérée et un métier "Autres"), la couverture des indicateurs de sortie du modèle définis par flottille est complète. Ainsi, les mortalités complémentaires, au même titre que les autres variables complémentaires que le modèle sera amené à évaluer, sont désormais uniquement celles engendrées par les flottilles non modélisées. à partir de la mortalité par pêche ventilée initialement. Le module procède également au calcul de l'indicateur "Fbar" pour chacune des espèces modélisées XSA ou SS3.

Ajoutons que les deux variables de sortie "Mortalité par pêche ventilée» (F) et "Mortalité par pêche Autres flottilles" (Foth) sont des paramètres internes (donc non renseignés dans le fichier de paramétrage car évalués lors de la construction de l'objet input) sur lesquels il est possible d'intervenir au moyen du module "Scénario". On décrira le mode opératoire de ce module dans le chapitre correspondant. Il faut simplement noter que de cette manière, l'utilisateur pourra par exemple intégrer dans la simulation la prise en compte d'un facteur de sélectivité à un niveau flottille-métier-âge pour une ou plusieurs espèces modélisées.

• XSA • Paramètres initiaux (inputs biologiques prélevés dans les feuillets de paramétrage Stocks)

Notation	Description	Source
$d_{e,f,m,i}$	Pourcentage de captures totales rejetées en nombre (par espèce, flottille, métier et âge)	Stock Assessment
$donth_i$	Pourcentage de captures totales rejetées en nombre (par âge sur l'ensemble des flottilles non modélisés)	Stock Assessment
sr_e	Taux de survie des rejets (par espèce)	Stock Assessment
$E_{f,m}$	Effort (par flottille et métier, produit d' $effort1$ et d' $effort2$)	SACROIS
$p_{e,i}$	Pondération pour le calcul du Fbar par espèce (ie 1 pour les âges à considérer, 0 pour les autres)	Stock Assessment

TABLE 1.7: Paramètres initiaux XSA pour le module "Mortalité par pêche"

• Variables calculées

Notation	Type	Description	Equation
$F_{e,f,m,i,t}$	Sortie	Mortalité par pêche (par espèce, flottille, métier et âge, à l'instant t). Entrée des modules <i>Captures</i> et <i>Dynamiques de populations</i>	$F_{e,f,m,i,t} = q_{e,f,m,i} \cdot E_{f,m,t}$
$K_{e,f,m,i}$	Interne	Facteur de correction de la mortalité par pêche lié à la survie des rejets.	$K_{e,f,m,i} = 1 - sr_e \cdot d_{e,f,m,i}$
$Fr_{e,f,m,i,t}$	Sortie	Mortalité par pêche corrigée (par espèce, flottille, métier et âge, à l'instant t). Entrée des modules <i>Captures</i> et <i>Dynamiques de populations</i>	$Fr_{e,f,m,i,t} = F_{e,f,m,i,t} \cdot K_{e,f,m,i}$
$Froth_{e,i}$	Sortie	Mortalité par pêche initiale "autres flottilles" corrigée par espèce et âge. Entrée des modules <i>Captures</i> et <i>Dynamiques de populations</i>	$Froth_{e,i} = Froth_{e,i} \cdot (1 - sr_e \cdot doth_i)$
$Fbar_{e,t}$	Sortie	Taux de mortalité par pêche par espèce.	$Fbar_{e,t} = \frac{1}{\sum_i p_{e,i}} \sum_i p_{e,i} \cdot (\sum_{f,m} Fr_{e,f,m,i} + Froth_{e,i})$

TABLE 1.8: iParamètres calculés XSA pour le module "Mortalité par pêche"

• SPiCT

• Paramètres initiaux (inputs biologiques prélevés dans les feuillets de paramétrage Stocks)

Notation	Description	Source
d_e	Pourcentage de captures totales rejetées sur les flottilles modélisées (par espèce)	Stock Assessment
$doth_e$	Pourcentage de captures totales rejetées sur l'ensemble des flottilles non modélisés (par espèce)	Stock Assessment

TABLE 1.9: Paramètres initiaux SPiCT pour le module "Mortalité par pêche"

• Variables calculées

Notation	Type	Description	Equation
----------	------	-------------	----------

$F_{e,f,m,i,t}$	Sortie	Mortalité par pêche (par espèce, flottille, métier et âge, à l'instant t). Entrée des modules <i>Captures</i> et <i>Dynamiques de populations</i>	$F_{e,f,m,i,t} = q_{e,f,m,i,t} \times F_{f,m,t}$
$Foth_{e,i}$	Sortie	Mortalité par pêche initiale "autres flottilles" par espèce et âge. Entrée des modules <i>Captures</i> et <i>Dynamiques de populations</i>	$Foth_{e,i} = F_{e,i} - \sum_{f,m} F_{e,f,m,i,0}$

TABLE 1.10: Paramètres calculés SPiCT pour le module "Mortalité par pêche"

• SS3

• Paramètres initiaux (inputs biologiques prélevés dans les feuillets de paramétrage Stocks)

Note : les paramètres biologiques SS3 se définissent sur des dimensions supplémentaires "saison" (indice 's', 4 modalités) et "cohorte" (indice 'mo' pour 'morph', 4 modalités)

Notation	Description	Source
$iniFq_{e,i,s,mo}$	Mortalité par pêche initiale "captures en nombre" (par espèce, âge, saison et cohorte)	Stock Assessment
$Fq_{e,i,s,mo}$	Mortalité par pêche projection ($t > 0$) "captures en nombre" (par espèce, âge, saison et cohorte)	Stock Assessment
$iniFq_{e,f,m,i,s,mo}$	Mortalité par pêche initiale "captures en nombre" (par espèce, flottille, métier, âge, saison et cohorte)	Stock Assessment
$Fq_{e,f,m,i,s,mo}$	Mortalité par pêche projection ($t > 0$) "captures en nombre" (par espèce, flottille, métier, âge, saison et cohorte)	Stock Assessment
$iniFqLwt_{e,i,s,mo}$	Mortalité par pêche initiale "poids débarqués" (par espèce, âge, saison et cohorte)	Stock Assessment
$FqLwt_{e,i,s,mo}$	Mortalité par pêche projection ($t > 0$) "poids débarqués" (par espèce, âge, saison et cohorte)	Stock Assessment
$iniFqLwt_{e,f,m,i,s,mo}$	Mortalité par pêche initiale "poids débarqués" (par espèce, flottille, métier, âge, saison et cohorte)	Stock Assessment

$FqLwt_{e,f,m,i,s,mo}$	Mortalité par pêche projection ($t > 0$) "poids débarqués" (par espèce, flottille, métier, âge, saison et cohorte)	Stock Assessment
$iniFqDwt_{e,i,s,mo}$	Mortalité par pêche initiale "poids rejetés" (par espèce, âge, saison et cohorte)	Stock Assessment
$FqDwt_{e,i,s,mo}$	Mortalité par pêche projection ($t > 0$) "poids rejetés" (par espèce, âge, saison et cohorte)	Stock Assessment
$iniFqDwt_{e,f,m,i,s,mo}$	Mortalité par pêche initiale "poids rejetés" (par espèce, flottille, métier, âge, saison et cohorte)	Stock Assessment
$FqDwt_{e,f,m,i,s,mo}$	Mortalité par pêche projection ($t > 0$) "poids rejetés" (par espèce, flottille, métier, âge, saison et cohorte)	Stock Assessment

TABLE 1.11: Paramètres initiaux SS3 pour le module "Mortalité par pêche"

• Variables calculées

Notation	Type	Description	Equation
$iniFothq_{e,i,s,mo}$	-	Mortalité par pêche initiale "captures en nombre – Autres flottilles" (par espèce, âge, saison et cohorte)	$iniFothq_{e,i,s,mo} = iniFq_{e,i,s,mo} - \sum_{f,m} iniFq_{e,f,m,i,s,mo}$
$Fothq_{e,i,s,mo}$	-	Mortalité par pêche projection ($t > 0$) "captures en nombre – Autres flottilles" (par espèce, âge, saison et cohorte)	$Fothq_{e,i,s,mo} = Fq_{e,i,s,mo} - \sum_{f,m} Fq_{e,f,m,i,s,mo}$
$iniFothqLwt_{e,i,s,mo}$	-	Mortalité par pêche initiale "poids débarqués – Autres flottilles" (par espèce, âge, saison et cohorte)	$iniFothqLwt_{e,i,s,mo} = iniFqLwt_{e,i,s,mo} - \sum_{f,m} iniFqLwt_{e,f,m,i,s,mo}$
$FothqLwt_{e,i,s,mo}$	-	Mortalité par pêche projection ($t > 0$) "poids débarqués – Autres flottilles" (par espèce, âge, saison et cohorte)	$FothqLwt_{e,i,s,mo} = FqLwt_{e,i,s,mo} - \sum_{f,m} FqLwt_{e,f,m,i,s,mo}$

$iniFothqDwt_{e,i,s,mo}$	-	Mortalité par pêche initiale "poids rejetés – Autres flottilles" (par espèce, âge, saison et cohorte)	$iniFothqDwt_{e,i,s,mo} = iniFqDwt_{e,i,s,mo} - \sum_{f,m} iniFqDwt_{e,f,m,i,s,mo}$
$FothqDwt_{e,i,s,mo}$	-	Mortalité par pêche projection ($t > 0$) "poids rejetés – Autres flottilles" (par espèce, âge, saison et cohorte)	$FothqDwt_{e,i,s,mo} = FqDwt_{e,i,s,mo} - \sum_{f,m} FqDwt_{e,f,m,i,s,mo}$

TABLE 1.12: Paramètres calculés SS3 pour le module "Mortalité par pêche"

Précisons enfin que les espèces dites "statiques" ne s'inscrivent pas dans ce module.

1.2.2 Module "Dynamique de populations"

1.2.2.1 Module principal

Ce module se charge d'opérer le calcul des indicateurs relatifs aux stocks des espèces considérées dynamiquement dans la simulation (XSA et SS3). En partant de la mortalité par pêche appliquée au stock et estimée en sortie du module décrit dans le chapitre précédent, il estime pour chaque pas de temps, et à partir d'un état initial, la mortalité totale, les effectifs totaux aux âges, la biomasse et la biomasse reproductrice résultantes. Notons qu'il est possible de définir dans le fichier de paramétrage un recrutement a priori par espèce pour les projections (cf première ligne du tableau 1.13). Toutefois, si le module "Recrutement" est activé (voir 1.2.2.2), le module principal fera appel à lui au cours de la simulation afin d'estimer $N_{e,i=0,t}$ pour tout instant non initial (fonctionnel uniquement pour la dynamique XSA)

- XSA
- Paramètres initiaux

Notation	Description	Source
$N_{e,i,t=0}$ et $N_{e,i=0,t}$	Effectifs initiaux en nombre par espèce et par âge, et effectifs des recrutements par espèce à l'instant t .	Stock Assessment
$M_{e,i}$	Taux de mortalité naturelle par espèce et par âge	Stock Assessment
$w_{e,i}$	Poids moyen dans le stock par espèce à l'âge i	Stock Assessment
$mr_{e,i}$	Taux de maturité dans le stock par espèce à l'âge i	Stock Assessment

TABLE 1.13: Paramètres initiaux XSA pour le module "Dynamique de populations"

- Variables calculées

Notation	Type	Description	Equation
$Z_{e,i,t}$	Sortie	Coefficient de mortalité totale (par espèce et âge à l'instant t). Entrée du module Captures.	$Z_{e,i,t} = M_{e,i} + \sum_{f,m} Fr_{ef,f,i,t} + Froth_{e,i,t}$
$N_{e,i,t}$	Sortie	Effectif en nombre du groupe d'âge i à l'instant t par espèce.	$N_{e,i+1,t+1} = N_{e,i,t} \cdot e^{-Z_{e,i,t}}$ <hr/> $N_{e,i+1,t+1} = N_{e,i,t} \cdot e^{-Z_{e,i,t}} + N_{e,i+1,t} \cdot e^{-Z_{e,i+1,t}}$ <p>pour le groupe d'âge +</p>
$B_{e,t}$	Sortie	Biomasse à l'instant t par espèce.	$B_{e,t} = \sum_i N_{e,i,t} \cdot w_{e,i}$
$SSB_{e,t}$	Sortie	Biomasse reproductrice à l'instant t par espèce.	$SSB_{e,t} = \sum_i N_{e,i,t} \cdot w_{e,i} \cdot mr_{e,i}$

TABLE 1.14: Paramètres calculés XSA pour le module "Dynamique de populations"

• SPiCT

• Paramètres initiaux

Notation	Description	Source
$B_{e,t=0}$	Biomasse à l'instant initial par espèce.	Stock Assessment
r_e	Taux de croissance intrinsèque par espèce.	Stock Assessment
K_e	Capacité de charge par espèce.	Stock Assessment
n_e	Paramètre de détermination de la forme de la courbe de production, par espèce.	Stock Assessment

TABLE 1.15: Paramètres initiaux SPiCT pour le module "Dynamique de populations"

• Variables calculées

Notation	Type	Description	Equation
$B_{e,t}$	Sortie	Biomasse à l'instant t par espèce.	$B_{e,t} = B_{e,t-1} \times \left(1 + \frac{r_e}{n_e - 1} \cdot \left(1 - \left[\frac{B_{e,t-1}}{K_e}\right]^{n_e-1}\right) - F_{e,t-1}\right)$

TABLE 1.16: Paramètres calculés SPiCT pour le module "Dynamique de populations"

• SS3

• Paramètres initiaux

Notation	Description	Source
$IniN_{e,i,s,mo,t=0}$	Effectifs initiaux en nombre par espèce et par âge, saison et cohorte.	Stock Assessment
$N_{e,i,s=1,mo,t=1}$	Effectifs de projection initiaux ($t > 0$, saison 1) en nombre par espèce, par âge et cohorte.	Stock Assessment
$N_{e,i=0,s,mo=s}$	Recrutement par saison en nombre.	Stock Assessment
$matWt_{e,i,mo}$	Poids moyen pondéré pour calcul de la SSB par espèce, âge et cohorte.	Stock Assessment
$M_{e,i,mo}$	Taux de mortalité naturelle par espèce, par âge et cohorte.	Stock Assessment

TABLE 1.17: Paramètres initiaux SS3 pour le module "Dynamique de populations"

• **Variables calculées** (pour $t > 0$. A $t = 0$, les versions "ini" des variables de mortalité sont utilisées)

Notation	Type	Description	Equation
$Z_{e,i,s,mo,t}$	Sortie	Coefficient de mortalité totale (par espèce et âge, saison, cohorte à l'instant t). Entrée du module <i>Captures</i> .	$Z_{e,i,s,mo,t} = M_{e,i,mo} + \sum_{f,m} Fq_{e,f,m,i,s,mo,t} + Fq_{e,i,s,mo,t}$
$auxiA_{e,i,s,mo,t}$	Interne	Variable de calcul intermédiaire.	$auxiA_{e,i,s,mo,t} = N_{e,i,s,mo,t} \times \exp^{-Z_{e,i,s,mo,t}/4}$
$N_{e,i,s,mo,t}$	Sortie	Effectif en nombre par espèce et âge, saison, cohorte à l'instant t.	$N_{e,i,s+1,mo,t} = auxiA_{e,i,s,mo,t} + N_{e,i,s+1,mo,t} \cdot 1_{i=0 \cap mo=s+1}$ si $s < 4$ $N_{e,i,s=1,mo,t+1} = auxiA_{e,i-1,s=4,mo,t} + auxiA_{e,i,s=4,mo,t} \cdot 1_{i='gp+'}$ sinon
$SSB_{e,t}$	Sortie	Biomasse reproductrice à l'instant t par espèce.	$SSB_{e,t} = \sum_{i,mo} matWt_{e,i,mo} \times N_{e,i,s=1,mo,t}$
$Fss3_{e,f,m,i}$	Sortie	Mortalité par pêche annuelle aux âges "captures en nombre" (par espèce, flottille, métier, âge)	$Fss3_{e,f,m,i} = \sum_s \frac{\sum_{mo} Fq_{e,f,m,i,s,mo} \times N_{e,i,s,mo}}{4 \times \sum_{mo} N_{e,i,s,mo}}$
$Fothss3_{e,i}$	Sortie	Mortalité par pêche annuelle aux âges "captures en nombre - Autres flottilles" (par espèce et âge)	$Fothss3_{e,i} = \sum_s \frac{\sum_{mo} Fq_{e,i,s,mo}}{4 \times \sum_{mo} N_{e,i,s,mo}}$

$Fbar_{e,t}$	Sortie	Taux de mortalité par pêche par espèce et âge.	$Fbar_{e,t} = \frac{1}{\sum_i p_{e,i}} \sum_i p_{e,i} \cdot$ $\left(\sum_{f,m} Fss3_{e,f,m,i} + Fothss3_{e,i} \right)$
--------------	--------	--	---

TABLE 1.18: Paramètres calculés SS3 pour le module "Dynamique de populations"

1.2.2.2 Module "Recrutement"

Le module "Recrutement" est une composante du module composite "Stochasticité / Bootstrap" qui va donner la possibilité à l'utilisateur de générer un recrutement pour une espèce donnée, de manière stochastique, et ce à tout instant non initial, qui sera intégré en tant qu'input du modèle (voir illustration sur le schéma 1.1). Ce module propose pour l'instant deux méthodes de génération de recrutement dit "aléatoire". L'utilisateur peut ainsi assigner à chacune des espèces modélisées un type de recrutement donné avec des paramètres qui lui sont propres.

La première méthode va gérer les recrutements issus de relations stock-recrutement. Elle permet, pour le moment, de choisir entre 5 types de relations paramétrables (recrutement moyen, de type Hockey Stick, Beverton & Holt, Ricker, et Shepherd), et autorise l'adjonction d'un bruit paramétrable.

La seconde option permet de procéder aux générations par tirage aléatoire. Il est alors possible de réaliser un tirage aléatoire dans un historique de recrutement renseigné dans le fichier de paramétrage (le tirage aléatoire pouvant s'effectuer indépendamment ou non par espèce), ou encore d'envisager le recrutement comme la réalisation d'une variable aléatoire de loi et de paramètre(s) choisis par l'utilisateur.

Précisons enfin que ce module "Recrutement" n'est pour le moment applicable que sur les dynamiques de type XSA.

1.2.3 Module "Captures, débarquements et rejets"

A l'inverse du module précédent qui s'appliquait essentiellement du point de vue du stock, le module ici abordé se place du point de vue de l'engin de pêche. Des indicateurs de captures (en nombre et en poids) calculés sont déduites les quantités retenues et les quantités rejetées. En outre, les données de captures sont disponibles pour deux niveaux d'agrégation, l'une intégrant ainsi les captures des flottilles et métiers non modélisés.

- XSA

- Paramètres initiaux

Notation	Description	Source
$wD_{e,i}$	Poids moyen des individus dans la fraction rejetée par espèce à l'âge i	Stock Assessment
$wC_{e,i}$	Poids moyen des individus dans la capture par espèce à l'âge i	Stock Assessment

TABLE 1.19: Paramètres initiaux XSA pour le module "Captures, débarquements et rejets"

- Variables calculées

Notation	Type	Description	Equation
$C_{e,f,m,i,t}$	Sortie	Captures en nombre (par espèce, flottille, métier et âge, sur la période t).	$C_{e,f,m,i,t} = \frac{F_{e,f,m,i,t}}{Z_{e,i,t}} \times N_{e,i,t} \times (1 - e^{-Z_{e,i,t}})$
$C_{e,i,t}$	Sortie	Captures en nombre (par espèce et âge, sur la période t).	$C_{e,i,t} = \frac{\sum_{f,m} F_{e,f,m,i,t} + F_{oth_{e,i}}}{Z_{e,i,t}} \times N_{e,i,t} \times (1 - e^{-Z_{e,i,t}})$
$Y_{e,f,m,i,t}$	Sortie	Captures en poids (par espèce, flottille, métier et âge, sur la période t).	$Y_{e,f,m,i,t} = wC_{e,i} \times C_{e,f,m,i,t}$
$Y_{e,i,t}$	Sortie	Captures en poids (par espèce et âge sur la période t).	$Y_{e,i,t} = wC_{e,i} \times C_{e,i,t}$
$D_{e,f,m,i,t}$	Sortie	Rejets en poids (par espèce, flottille, métier et âge, sur la période t).	$D_{e,f,m,i,t} = d_{e,f,m,i} \times wD_{e,i} \times C_{e,f,m,i,t}$ si $wD_{e,i}$ disponible $D_{e,f,m,i,t} = d_{e,f,m} \times Y_{e,f,m,t}$ sinon
$L_{e,f,m,i,t}$	Sortie	Débarquements en poids (par espèce, flottille, métier et âge, sur la période t). Entrée du module <i>Marché</i> .	$L_{e,f,m,i,t} = Y_{e,f,m,i,t} - D_{e,f,m,i,t}$

TABLE 1.20: Paramètres calculés XSA pour le module "Captures, débarquements et rejets"

• SPiCT

• Paramètres initiaux

Notation	Description	Source
$d_{e,f,m}$	Pourcentage de captures totales rejetées en poids (par espèce, flottille, métier)	Stock Assessment

TABLE 1.21: Paramètres initiaux SPiCT pour le module "Captures, débarquements et rejets"

• Variables calculées

Notation	Type	Description	Equation
$Y_{e,f,m,t}$	Sortie	Captures en poids (par espèce, flottille, métier sur la période t).	$Y_{e,f,m,t} = F_{e,f,m,t} \times B_{e,t}$
$Y_{e,t}$	Sortie	Captures en poids (par espèce sur la période t).	$Y_{e,t} = (\sum_{f,m} F_{e,f,m,t} + F_{roth_{e,t}}) \times B_{e,t}$

$D_{e,f,m,t}$	Sortie	Rejets en poids (par espèce, flottille, métier et âge, sur la période t).	$D_{e,f,m,t} = d_{e,f,m} \times Y_{e,f,m,t}$
$L_{e,f,m,t}$	Sortie	Débarquements en poids (par espèce, flottille, métier et âge, sur la période t). Entrée du module <i>Marché</i> .	$L_{e,f,m,t} = Y_{e,f,m,t} - D_{e,f,m,t}$

TABLE 1.22: Paramètres calculés SPiCT pour le module "Captures, débarquements et rejets"

• SS3

Notation	Type	Description	Equation
$C_{e,f,m,i,t}$	Sortie	Captures en nombre (par espèce, flottille, métier et âge, sur la période t).	$C_{e,f,m,i,t} = \sum_{s,mo} \left(\frac{Fq_{e,f,m,i,s,mo,t}}{Z_{e,i,s,mo,t}} \times N_{e,i,s,mo,t} \times (1 - e^{-Z_{e,i,s,mo,t}/4}) \right)$
$C_{e,i,t}$	Sortie	Captures en nombre (par espèce et âge, sur la période t).	$C_{e,i,t} = \sum_{s,mo} \left(\frac{Fq_{e,f,m,i,s,mo,t} - M_{e,i,mo}}{Z_{e,i,s,mo,t}} \times N_{e,i,s,mo,t} \times (1 - e^{-Z_{e,i,s,mo,t}/4}) \right)$
$L_{e,f,m,i,t}$	Sortie	Débarquements en poids (par espèce, flottille, métier et âge, sur la période t). Entrée du module <i>Marché</i> .	$L_{e,f,m,i,t} = \sum_{s,mo} \left(\frac{FqLwt_{e,f,m,i,s,mo,t}}{Z_{e,i,s,mo,t}} \times N_{e,i,s,mo,t} \times (1 - e^{-Z_{e,i,s,mo,t}/4}) \right)$
$D_{e,f,m,i,t}$	Sortie	Rejets en poids (par espèce, flottille, métier et âge, sur la période t).	$D_{e,f,m,i,t} = \sum_{s,mo} \left(\frac{FqDwt_{e,f,m,i,s,mo,t}}{Z_{e,i,s,mo,t}} \times N_{e,i,s,mo,t} \times (1 - e^{-Z_{e,i,s,mo,t}/4}) \right)$
$Y_{e,f,m,i,t}$	Sortie	Captures en poids (par espèce, flottille, métier et âge, sur la période t).	$Y_{e,f,m,i,t} = L_{e,f,m,i,t} + D_{e,f,m,i,t}$
$L_{e,i,t}$	Sortie	Débarquements en poids (par espèce, flottille, métier et âge, sur la période t). Entrée du module <i>Marché</i> .	$L_{e,i,t} = \sum_{s,mo} \left(\left(\frac{FqLwt_{e,f,m,i,s,mo,t}}{Z_{e,i,s,mo,t}} + \frac{FothqLwt_{e,i,s,mo,t}}{Z_{e,i,s,mo,t}} \right) \times N_{e,i,s,mo,t} \times (1 - e^{-Z_{e,i,s,mo,t}/4}) \right)$
$D_{e,i,t}$	Sortie	Rejets en poids (par espèce, flottille, métier et âge, sur la période t).	$D_{e,i,t} = \sum_{s,mo} \left(\left(\frac{FqDwt_{e,f,m,i,s,mo,t}}{Z_{e,i,s,mo,t}} + \frac{FothqDwt_{e,i,s,mo,t}}{Z_{e,i,s,mo,t}} \right) \times N_{e,i,s,mo,t} \times (1 - e^{-Z_{e,i,s,mo,t}/4}) \right)$

$Y_{e,i,t}$	Sortie	Captures en poids (par espèce et âge sur la période t).	$Y_{e,i,t} = L_{e,i,t} + D_{e,i,t}$
-------------	--------	--	-------------------------------------

TABLE 1.23: Paramètres calculés SS3 pour le module "Captures, débarquements et rejets"

- **Modèle statique**
 - **Paramètres initiaux**

Notation	Description	Source
$LPU E_{e,f,m}$	Débarquements moyens par unité d'effort en poids, par espèce, flottille et métier	SACROIS
$d_{e,f,m}$	Taux de rejets dans la partie capturée, par espèce, flottille et métier ($d_{e,f,m} < 1$)	OBSMER

TABLE 1.24: Paramètres initiaux "espèces statiques" pour le module "Captures, débarquements et rejets"

- **Variables calculées**

Notation	Type	Description	Equation
$statL_{e,f,m,t}$	Sortie	Débarquements en poids (par espèce, flottille, métier, sur la période t).	$statL_{e,f,m,t} = LPU E_{e,f,m,t} \times E_{f,m,t}$
$statD_{e,f,m,t}$	Sortie	Rejets en poids (par espèce, flottille, métier, sur la période t).	$statD_{e,f,m,t} = \frac{d_{e,f,m}}{1 - d_{e,f,m}} \times statL_{e,f,m,t}$
$statY_{e,f,m,t}$	Sortie	Captures en poids (par espèce, flottille, métier, sur la période t).	$statY_{e,f,m,t} = statL_{e,f,m,t} + statD_{e,f,m,t}$

TABLE 1.25: Paramètres calculés "espèces statiques" pour le module "Captures, débarquements et rejets"

- **Modélisation de l'obligation de débarquement (OD)**

Un modèle permettant de simuler l'application de mesures d'obligation de débarquement peut (ou non) être activé afin de compléter le module. Ce modèle et ses implications sur les indicateurs en sortie sont décrits ci-dessous.

- **Paramètres initiaux**

Notation	Description	Source
----------	-------------	--------

$dd1_{e,f,m}$	Taux de rejets "exemption" ou "non respect" par espèce, flottille, métier (en % de la capture totale de l'espèce considérée par flottille-métier)	-
$dd2_{e,f,m}$	Taux de rejets "exemption" ou "non respect" par espèce, flottille, métier (en % de la capture totale par flottille-métier)	-
$dd_{e,f,m}$	Taux de rejets "exemption" ou "non respect" finalement appliqué espèce, flottille, métier. qqs (e,f,m), $dd_{e,f,m} = dd1_{e,f,m}$ si $dd1_{e,f,m}$ renseigné, $dd_{e,f,m} = dd2_{e,f,m}$ sinon	-
OD_e	Indicatrice d'application de l'obligation de débarquement (1/0), par espèce. Prévaut sur la disponibilité de $dd1$ et $dd2$.	-
$dst_{e,f,m}$	Taux de rejets sous-taille par espèce statiques, en pourcentage du poids de rejets débarqués de l'espèce.	-

TABLE 1.26: Paramètres initiaux du modèle OD pour le module "Captures, débarquements et rejets"

• Variables calculées

Notation	Type	Description	Equation
$DD_{e,f,m,i,t}$	Sortie	Rejets "rejetés" en poids (par espèce modélisée, flottille, métier, âge, sur la période t).	$DD_{e,f,m,i,t} = \frac{D_{e,f,m,i,t}}{\sum_i D_{e,f,m,i,t}} \times dd_{e,f,m} \sum_i Y_{e,f,m,i,t}$ si $OD_e = 1$ et $dd1_{e,f,m}$ renseigné, <hr/> $DD_{e,f,m,i,t} = \frac{D_{e,f,m,i,t}}{\sum_i D_{e,f,m,i,t}} \times dd_{e,f,m} \sum_i Y_{e,f,m,i,t}$ si $OD_e = 1$, $dd1_{e,f,m}$ non renseigné et $dd2_{e,f,m}$ renseigné, <hr/> $DD_{e,f,m,i,t} = D_{e,f,m,i,t}$ sinon
$LD_{e,f,m,i,t}$	Sortie	Rejets "débarqués" en poids (par espèce modélisée, flottille, métier, âge sur la période t).	$LD_{e,f,m,i,t} = D_{e,f,m,i,t} - DD_{e,f,m,i,t}$

$statDD_{e,f,m,i,t}$	Sortie	Rejets "rejetés" en poids (par espèce statique, flottille, métier, sur la période t).	$statDD_{e,f,m,t} = dd_{e,f,m} \times Y_{e,f,m,t}$ si $OD_e = 1$ et $dd1_{e,f,m}$ renseigné, <hr/> $statDD_{e,f,m,t} = dd_{e,f,m} \times \sum_e Y_{e,f,m,t}$ si $OD_e = 1$, $dd1_{e,f,m}$ non renseigné et $dd2_{e,f,m}$ renseigné, <hr/> $statDD_{e,f,m,t} = statD_{e,f,m,t}$ sinon
$statLD_{e,f,m,t}$	Sortie	Rejets "débarqués" en poids (par espèce statique, flottille, métier, sur la période t).	$statLD_{e,f,m,t} =$ $statD_{e,f,m,t} - statDD_{e,f,m,t}$
$statLDst_{e,f,m,t}$	Sortie	Rejets sous-taille "débarqués" en poids (par espèce statique, flottille, métier, sur la période t).	$statLDst_{e,f,m,i,t} =$ $statLD_{e,f,m,t} \times dst_{e,f,m,t}$
$statLDor_{e,f,m,i,t}$	Sortie	Rejets autres "débarqués" en poids (par espèce statique, flottille, métier, sur la période t).	$statLDor_{e,f,m,i,t} =$ $statLD_{e,f,m,t} - statLDst_{e,f,m,i,t}$

TABLE 1.27: Paramètres calculés du modèle OD pour le module "Captures, débarquements et rejets"

1.2.4 Module "Marché"

Le module Marché n'est pas seulement voué à simuler un modèle de prix moyen plus ou moins élaboré, mais va également servir de passerelle entre la partie biologique et la partie économique du modèle. Cette étape va être en effet caractérisée par deux actions affectant la définition intrinsèque de certaines variables issues de modules biologiques. D'une part, la déclinaison par âge mutera en une déclinaison par catégorie commerciale grâce à l'apport d'une clé âge/catégorie ($icat_{e,i,c}$ voir chapitre 2). D'autre part, le niveau "métier" déclinant les paramètres économiques étant potentiellement différent de celui pour lesquelles les données biologiques sont définies, une matrice de recodage du niveau métier ($mat_{f,m1,m2}$) sera requise afin d'assurer la concordance des dimensions. L'utilisateur a le choix entre modéliser sa variable de prix ou la considérer constante sur un niveau de définition donné (par espèce et par catégorie, par exemple). Dans tous les cas de figure, les paramètres de prix seront déclarés dans un onglet spécifique "Marché" du fichier de paramétrage. Les tableaux ci-dessous décrivent un exemple de modèle de prix ; d'autres modèles s'ajouteront dans les développements futurs.

• Paramètres initiaux

Notation	Description	Source
----------	-------------	--------

$\alpha_{e,c}$	Constante du modèle de prix spécifique à chaque catégorie c	-
$\beta_{e,c}$	Elasticité prix de la catégorie c	-
$\gamma_{e,c}$	Elasticité de substitution de la catégorie c	-
$icat_{e,i,c}$	Matrice de conversion âge/catégorie par espèce : $P_e(c/i)$	-
$mat_{f,m1,m2}$	Matrice binaire d'appartenance : $(f \cap mtier1) \subset (f \cap mtier2)$	-
$P_{e,f,m,(c),t}$	Prix moyen par unité de poids (par espèce, flottille, métier (catégorie si espèce XSA ou SS3) à l'instant t). Entrée du module Economique utilisée si le modèle de prix n'est pas appliqué pour l'espèce en question.	SACROIS

TABLE 1.28: Paramètres initiaux pour le module "Marché"

• Variables calculées

Notation	Type	Description	Equation
$L_{e,f,m1,c,t}$	Interne	Captures en nombre (par espèce, flottille, métier "bio" et catégorie, sur la période t).	$L_{e,f,m1,c,t} = \sum_i (L_{e,f,m1,i,t} \times icat_{e,i,c})$
$L_{e,f,m,c,t}$	Sortie	Captures en nombre (par espèce, flottille, métier "éco" et catégorie, sur la période t). Entrée du module Economique	$L_{e,f,m,c,t} = \sum_{(f,m1) \perp mat_{f,m1,m}=1} L_{e,f,m1,c,t}$
$P_{e,c,t}$	Sortie	Prix moyen modélisé par unité de poids (par espèce et catégorie à l'instant t). Entrée du module Economique	$P_{e,c,t} = \alpha_{e,c} + \beta_{e,c} \times \ln\left(\sum_{f,m} L_{e,f,m,c,t}\right) + \gamma_{e,c} \times \ln\left(\sum_{f,m} L_{e,f,m,c,t}\right)$

TABLE 1.29: Paramètres calculés pour le module "Marché"

1.2.5 Module "Economique"

Hypothèses générales du module économique :

- modèle dynamique en temps discret (années)
- modèle déterministe (pour l'instant)
- le modèle utilise en entrée des paramètres issus des fiches "flottes" et des fiches "produits", ainsi que les outputs du modèle biologique. Une autre partie des paramètres d'entrée est générée par les calculs internes au modèle.
- le modèle utilise en entrée des paramètres exprimés en termes nominaux. Ces paramètres pourront aussi être exprimés en termes réels une fois l'inflation déduite (2.8% en 2008 ; indice général des prix à la consommation INSEE, coefficient de transformation de l'euro ou du franc d'une année en euro ou franc d'une autre année¹).

Description des variables en jeu et mise en équation du modèle économique :

- Notation : intitulé exact de la variable tel qu'il sera utilisé dans l'implémentation et à l'intérieur des fichiers de paramétrage
- Type(s) : statut de la variable au sein de l'organisation du modèle (E : entrée, C : calculée, EC : entrée calculée, I : calcul interne, S : sortie...)
- Description : description succincte de la variable
- Unité : unité utilisée dans le modèle (peut différer de l'unité employée dans un feuillet de paramétrage de stock, pourvu que le champ "multiplicateur" de ce même feuillet soit ajusté en conséquence)
- Équation : formulation de la variable en fonction de paramètres déjà déclarés (ne concerne que les variables calculées)

N	Notation	Type	Description	Unité	Equation
i1	$L_{e,f,m,c}$	EC (_{bio})	Débarquements totaux par flotte, métier, espèce (et catégorie)	tonnes	
i2	$P_{e,f,m,c}$	EC (_{marché})	Prix moyen par flotte, métier, espèce (et catégorie)	euro/kg	

1. <http://www.insee.fr/fr/themes/indicateur.asp?id=29&page=achatfranc.htm>

i3	θ_e	E	Multiplicateur (par espèce) à appliquer aux prix "débarquements" afin d'aboutir aux prix "rejets débarqués de taille commerciale". Inclus dans $[0;1]$.	-	
i4	$LD_{e,f,m,c}$	EC _(bio)	Rejets débarqués en poids, par flottille, métier, espèce modélisée et catégorie (taille et sous-taille)	tonnes	
i5	$statLDor_{e,f,m}$	EC _(bio)	Rejets débarqués "autres" en poids, par flottille, métier et espèce statique	tonnes	
i6	$statLDst_{e,f,m}$	EC _(bio)	Rejets débarqués "sous taille" en poids, par flottille, métier et espèce statique	tonnes	
i7	pst_e	E	Prix au kg de la partie "rejets débarqués sous taille" pour les espèces statiques	euro/kg	
i8	nbv_f	E	Nombre de navires par flottille	nombre	

i9	$nbv_{f,m}$	E	Nombre de navires par flottille et métier	nombre	
i10	$lc_{f,m}$	E	Taxes de débarquements (en % du CA) par flottille et métier	%	
i11	$lcd_{f,m}$	E	Taxes de débarquements des rejets "sous taille" (en % du CA) par flottille et métier	%	
i12	$tripLgth_f$	E	Durée moyenne d'une marée par navire d'une flottille par an	heures	
i13	$tripLgth_{f,m}$	E	Durée moyenne d'une marée par navire d'une flottille-métier	heures	
i14	$nbTrip_f$	E	Nombre de marées annuel par navire d'une flottille	nombre	
i15	$nbTrip_{f,m}$	E	Nombre de marées annuel par navire d'une flottille-métier	nombre	
i16	$nbds_f$	E	Nombre de jours de mer par navire d'une flottille par an	jours	
i17	$nbds_{f,m}$	E	Nombre moyen de jours de mer par navire d'une flottille-métier et par an	jours	

i18	$effort1_f$	E	Première composante d'effort par navire d'une flottille par an	-	$nbTrip_{f(i14)}$ ou $nbd_{sf(i16)}$
i19	$effort1_{f,m}$	E	Première composante d'effort par navire d'une flottille-métier par an	-	$nbTrip_{f,m(i15)}$ ou $nbd_{sf,m(i17)}$
i20	$effort2_f$	E	Deuxième composante d'effort par navire d'une flottille par an	-	$tripLgth_{f(i12)}$ ou 1
i21	$effort2_{f,m}$	E	Deuxième composante d'effort par navire d'une flottille-métier par an	-	$tripLgth_{f,m(i13)}$ ou 1
i22	$Lref_{f,m}$	E	Débarquements totaux de référence par flottille et métier	tonnes	
i23	$cnb_{f,m,ini}$	E	Effectif moyen par navire d'une flottille-métier	nombre	
i24	$ovc_{f,m}$	E	Autres coûts variables DCF déduits des coûts de débarquements ($lc_{f,m(i10)}$) par navire d'une flottille-métier	euro/an	
i25	$fc_{f,m}$	E	Coûts de carburant par navire d'une flottille-métier	euro/an	

i26	$v_{f,m}$	E	Prix du carburant par navire d'une flottille-métier	euro/L	
i27	$csr_{f,m}$	E	Part équipage (ratio du RAP) par navire d'une flottille-métier	%RAP	
i28	csr_f	E	Part équipage (ratio du RAP) par navire d'une flottille	%RAP	
i29	$cnb_{f,ini}$	E	Effectif moyen par navire d'une flottille	nombre	
i30	$persc_f$	E	Coûts de personnel initiaux par navire d'une flottille	euro/an	
i31	eec_f	E	Cotisations salariales totales par navire d'une flottille	euro/an	
i32	mwh	E	Salaire brut horaire minimum national	euro/an	
i33	rep_f	E	Coûts entretien et réparation par navire d'une flottille	euro/an	
i34	gc_f	E	Coût total engin par navire d'une flottille	euro/an	
i35	$fixc_f$	E	Autres coûts fixes par navire d'une flottille	euro/an	
i36	FTE_f	E	ETP par navire d'une flottille	hommes	
i37	dep_f	E	Amortissements par navire d'une flottille	euro/an	

i38	ic_f	E	Coût d'opportunité du capital par navire d'une flottille	euro/an	
i39	K_f	E	Valeur d'assurance par navire d'une flottille	euro/an	
i40	inv_f	E	Coût d'investissement annuel par navire d'une flottille	euro/an	
i41	$FTE_{f,m}$	E	ETP par navire d'une flottille-métier	hommes	
i42	$perscCalc$	E	Mode de calcul de la variable "coût du personnel"	-	0 : salaires par marin fixés 1 : part équipage constante (cshr) 2 : part équipage constante calculée (ccwr) 3 : part équipage constante + salaire marin supplémentaire fixé (cshr) 4 : part équipage constante calculée + salaire marin supplémentaire fixé (ccwr)
i43	$GVLref_{f,m}$	E	CA moyen initial par navire d'une flottille-métier	euro/an	

TABLE 1.30: Paramètres initiaux pour le module "Economique"

N	Notation	Type	Description	Unité	Equation
p1	$ETini_{f,m}$	I	Efficacité initiale de tri en tonnage par heure et homme, par flottille-métier	tonnes/h.homme	$\frac{Lref_{f,m(i22)}}{nbv_{f,m,ini(i9)} \cdot nbTrip_{f,m,ini(i15)} \cdot \frac{1}{tripLgth_{f,m,ini(i13)} \cdot cnb_{f,m,ini(i23)}}} \times$
p2	$ue_{f,m}$	EC	Unité d'effort annuelle par navire d'une flottille-métier	-	$effort1_{f,m(i19)} \cdot effort2_{f,m(i21)}$

p3	ue_f	EC	Unité d'effort annuelle par navire d'une flottille	-	$effort1_{f(i18)} \cdot effort2_{f(i20)}$
p4	$fvalue_{f,m}$	EC	Volume de carburant par unité d'effort et par navire d'une flottille-métier	L/ue	$\frac{fc_{f,m(i25)}}{vf_{f,m(i26)} \cdot ue_{f,m(p2)}}$
p5	$ovc_{ue_{f,m}}$	EC	Autres coûts variables par unité d'effort et par navire d'une flottille-métier	€/ue	$\frac{ovc_{f,m(i24)}}{ue_{f,m(p2)}}$
p6	$rtbs_{f,ini}$	I	Reste à partager par navire d'une flottille	€/an	$\sum_m (GVLref_{f,m(i43)} \cdot nbv_{f,m(i9)} \cdot (1 - lc_{f,m(i10)}) \cdot ovc_{f,m(i24)} \cdot fc_{f,m(i25)}) \times \frac{1}{nbv_{f(i8)}}$
p7	$ccwr_f$	EC	Part du coût de personnel en % du RAP par flottille	%	$\frac{persc_{f(i30)}}{rtbs_{f,ini(p6)}}$
p8	$opersc_f$	EC	Autres coûts de personnel par navire d'une flottille	€/an	$persc_{f(i30)} \cdot cshr_{f(i28)} \cdot rtbs_{f,ini(p6)}$

TABLE 1.31: Paramètres initiaux calculés pour le module "Economique"

• Équations régissant le modèle pour tout instant t : table "t"

On ajoute qu'un taux d'actualisation est appliqué aux variables de coût pour $t > t_{init}$. Pour simplifier l'écriture, on notera $X_{(act)}$ une variable de sortie du modèle dont une version actualisée est également disponible. On rappelle qu'une variable actualisée est réajustée selon la formule suivante :

$$X_{(act)} = \frac{x}{(1 + dr)^{t'}}$$
 à l'instant t

dr est le taux d'actualisation en entrée du module économique (voir fonction $IAM.args()$)

N	Notation	Type	Description	Unité	Equation
---	----------	------	-------------	-------	----------

t1	$GVLcom_{e,f,m}$	C	CA total issu des espèces commercialisables, par flottille, métier et espèce	euro/an	$\frac{\sum_{c \neq 999} P_{e,f,m,c(i2)} \times L_{e,f,m,c(i1)} + \sum_{c \neq 999} \theta_{e(i3)} \times P_{e,f,m,c(i2)} \times LD_{e,f,m,c(i4)} \text{ si e modélisée}}{P_{e,f,m,c(i2)} \times L_{f,m,e(i1)} + \theta_{e(i3)} \times P_{e,f,m,c(i2)} \times statLDor_{f,m,e(i5)} \text{ si e statique}}$
t2	$GVLst_{e,f,m}$	C	CA total issu des espèces sous-taille, par flottille, métier et espèce	euro/an	$\frac{\sum_{c \in 999} P_{e,f,m,c(i2)} \times L_{e,f,m,c(i1)} + \sum_{c \in 999} \theta_{e(i3)} \times P_{e,f,m,c(i2)} \times LD_{e,f,m,c(i4)} \text{ si e modélisée}}{pst_{e(i7)} \times statLDst_{e,f,m(i6)} \text{ si e statique}}$
t3	$GVLtot_{e,f,m}$	C	CA total par flottille, métier et espèce	euro/an	$GVLcom_{e,f,m(t1)} + GVLst_{e,f,m(t2)}$
t4	$GVLtot_{f,m}$	C	CA total par flottille et métier	euro/an	$\sum_e GVLtot_{e,f,m(t3)}$
t5	$GVLav_{f,m}$	C	CA moyen par navire d'une flottille-métier	euro/an	$\frac{GVLtot_{f,m(t4)}}{nbv_{f,m(i9)}}$
t6	$GVLtot_f$	C	CA total par flottille	euro/an	$\sum_m GVLtot_{f,m(t4)}$
t7	$GVLav_f$	C	CA moyen par navire d'une flottille	euro/an	$\frac{GVLtot_{f(t6)}}{nbv_{f(i8)}}$
t8	$NGVLav_{f,m}$	I	CA net par navire d'une flottille-métier	euro/an	$\frac{\sum_e GVLcom_{e,f,m(t1)} \times (1 - lc_{f,m(i10)})}{nbv_{f,m(i9)}} + \frac{\sum_e GVLst_{e,f,m(t2)} \times (1 - lcd_{f,m(i11)})}{nbv_{f,m(i9)}}$
t9	$cnb_{f,m}$	C	Réévaluation de l'équipage en fonction de la production, par navire d'une flottille-métier	hommes	$\frac{\sum_{e,c} (L_{e,f,m,c(i1)} + LD_{e,f,m,c(i4)})}{ETini_{f,m(p1)} \times nbv_{f,m(i9)} \times 1} \times \frac{1}{nbTrip_{f,m(i15)} \times tripLgth_{f,m(i13)}}$

t10	cnb_f	C	Réévaluation de l'équipage en fonction de la production, par navire d'une flottille	hommes	$\frac{\sum_m (cnb_{f,m(t9)} \times nbv_{f,m(i9)} \times nbTrip_{f,m(i15)} \times tripLgth_{f,m(i13)}) \times \frac{1}{nbv_{f(i8)}} \times \frac{1}{nbTrip_{f(i14)} \times tripLgth_{f(i12)}}}{1}$
t11	$rtbs_{f,m}$	C	Reste à partager par navire d'une flottille-métier	euro/an	$NGVLav_{f,m(t8)} - (ovcue_{f,m(p5)} + fvolve_{f,m(p4)} \times vff_{f,m(i26)}) \times ue_{f,m(p2)}$
t12	$rtbs_f$	C	Reste à partager par navire d'une flottille	euro/an	$\frac{\sum_m (rtbs_{f,m(t11)} \times nbv_{f,m(i9)})}{nbv_{f(i8)}}$
t13	$cshrT_{f,m}$	C	Part équipage par navire d'une flottille-métier	euro/an	$cshr_{f,m(i27)} \times rtbs_{f,m(t11)} \text{ si } perscCalc = 0 \text{ ou } 1, \text{ NA sinon)}$
t14	$cshrT_f$	C	Part équipage par navire d'une flottille	euro/an	$cshr_{f(i28)} \times \frac{rtbs_{f,ini(p6)}}{cnb_{f,ini(i29)}} \times cnb_{f(t10)} \text{ si } perscCalc_{(i42)} = 0 \text{ (salaires par marin fixé)}$ <hr/> $cshr_{f(i28)} \times rtbs_{f(t12)} \text{ si } perscCalc_{(i42)} = 1 \text{ (part équipage constante)}$ <hr/> $ccwr_{f(p7)} \times rtbs_{f(t12)} \text{ si } perscCalc_{(i42)} = 2 \text{ (part équipage constante)}$ <hr/> $cshr_{f(i28)} \times (rtbs_{f(t12)} + \frac{rtbs_{f,ini(p6)}}{cnb_{f,ini(i29)}} \times (cnb_{f(t10)} - cnb_{f,ini(i29)})) \text{ si } perscCalc_{(i42)} = 3 \text{ (salaires marin suppl. fixé)}$ <hr/> $ccwr_{f(p7)} \times (rtbs_{f(t12)} + \frac{rtbs_{f,ini(p6)}}{cnb_{f,ini(i29)}} \times (cnb_{f(t10)} - cnb_{f,ini(i29)})) \text{ si } perscCalc_{(i42)} = 4 \text{ (salaires marin suppl. fixé)}$
t15	$ncshr_f$	C	Part équipage nette par navire d'une flottille	euro/an	$cshrT_{f(t14)} - eec_{f(i31)}$

t16	ocl_f	C	Coût d'opportunité du travail par navire d'une flottille	euro/an	$mwh_{(i32)} \times cnb_{f(t10)} \times nbTrip_{f(i14)} \times tripLgth_{f(i12)}$
t17	cs_f	C	Surplus de l'équipage par navire d'une flottille	euro/an	$ncshr_{f(t15)} - ocl_{f(t16)}$
t18	$csTot_f$	C	Surplus total de l'équipage d'une flottille	euro/an	$cs_{f(t17)} \times nbv_{f(i8)}$
t19	gva_f	C	Valeur Ajoutée Brute pour un navire d'une flottille	euro/an	$rtbs_{f(t12)} - rep_{f(i33)} - gc_{f(i34)} - fixc_{f(i35)}$
t20	$gvamargin_f$	C	Ratio valeur ajoutée brute / CA d'un navire moyen d'une flottille	-	$\frac{gva_{f(t19)}}{GVLav_{f(t7)}}$
t21	$gvaFTE_f$	C	Ratio valeur ajoutée brute / ETP d'un navire moyen d'une flottille	-	$\frac{gva_{f(t19)}}{FTE_{f(i36)}}$
t22	ccw_f	C	Coûts de personnel moyens par navire d'une flottille	euro/an	$\frac{cshrT_{f(t14)} + oper_{sc_{f(p8)}}}{\text{si } perscCalc_{(i42)} = 0, 1 \text{ ou } 3}$ $\frac{cshrT_{f(t14)}}{\text{si } perscCalc_{(i42)} = 2 \text{ ou } 4}$
t23	$ccwCr_f$	C	Coûts du personnel par marin	euro/an	$\frac{ccw_{f(t22)}}{cnb_{f(t10)}}$
t24	$wageg_f$	C	Salaire brut par marin	euro/an	$\frac{cshrT_{f(t14)}}{cnb_{f(t10)}}$
t25	$wagen_f$	C	Salaire net par marin	euro/an	$\frac{ncshr_{f(t15)}}{cnb_{f(t10)}}$
t26	$wagegFTE_f$	C	Salaire brut par marin / ETP	-	$\frac{wageg_{f(t24)}}{FTE_{f(i36)}}$
t27	$wagegh_f$	C	Salaire horaire brut par marin	euro/h	$\frac{wagegFTE_{f(t26)}}{nbTrip_{f(i14)} \times tripLgth_{f(i12)}}$

t28	gp_f	C	Excédent brut d'exploitation par navire d'une flottille	euro/an	$gva_{f(t19)} - ccw_{f(t22)}$
t29	$gpmargin_f$	C	Ratio excédent brut d'exploitation / CA d'un navire moyen d'une flottille	-	$\frac{gp_{f(t28)}}{GVLav_{f(t7)}}$
t30	ncf_f	C	Excédent net d'exploitation par navire d'une flottille	euro/an	$gp_{f(t28)} - dep_{f(i37)}$
t31	np_f	C	Résultat net d'exploitation par navire d'une flottille	euro/an	$ncf_{f(t30)} - ic_{f(i38)}$
t32	$npmargin_f$	C	Ration résultat net d'exploitation / CA d'un navire moyen d'une flottille	-	$\frac{np_{f(t31)}}{GVLav_{f(t7)}}$
t33	$prof_f$	C	Profitabilité totale par navire d'une flottille	-	<div>"Hight" : si $npmargin_{f(t32)} > 10\%$</div> <div>"Reasonable" : si $npmargin_{f(t32)} \in [0\%; 10\%]$</div> <div>"Weak" : si $npmargin_{f(t32)} < 0\%$</div>
t34	$npmargin - Trend_f$	C	Development trend sur le RNE	-	<div>$\frac{npmargin_{f(t32)}}{\frac{1}{5} \sum_{T \in \{T-5, \dots, t-1\}} npmargin_{f(t32)}}$</div> <div>"Improved" : si $> 6\%$</div> <div>"Stable" : si $\in [-5\%; 6\%]$</div> <div>"Deterioration" : si $< -5\%$</div>
t35	$ssTot_f$	C	Surplus total de l'armateur d'une flottille	euro/an	$gp_{f(t28)} \times nbv_{f(i8)}$
t36	ps_f	C	Surplus total du producteur d'une flottille	euro/an	$nbv_{f(i8)} \times (cs_{f(t17)} + gp_{f(t28)})$

t37	sts_f	C	Surplus de l'État associé à une flottille	euro/an	$\sum_m lc_{f,m(i10)} \times GVLav_{f,m(t5)} \times nbv_{f,m(i9)}$
t38	BER_f	C	Break Even Revenue	euro/an	$\frac{GVLtot_f(t6) \times nbv_{f(i8)}}{\sum_m rtbs_{f,m(t11)} \times nbv_{f,m(i9)} (fixc_{f(i35)} + ic_{f(i38)} + dep_{f(i37)})}$
t39	$CRBER_f$	C	Ratio Revenu / BER d'un navire moyen d'une flottille	-	$\frac{GVLav_{f(t7)}}{BER_{f(t38)}}$
t40	$fuelEff_f$	C	Ratio Volume de carburant / Débarquements totaux d'un navire d'une flottille	-	$\frac{\sum_m fvalue_{f,m(p4)} \times ue_{f,m(p2)} \times nbv_{f,m(i9)}}{1 / (\sum_{m,e,c} (L_{e,f,m,c(i1)} + LD_{e,f,m,c(i4)}) + \sum_{m,e} (L_{e,f,m,c(i1)} + statLDor_{e,f,m(i5)} + statLDst_{e,f,m(i6)}))}$
t41	$fvolGVA_f$	C	Ratio volume de carburant / valeur ajoutée brute d'un navire moyen d'une flottille	-	$\frac{\sum_m fvalue_{f,m(p4)} \cdot ue_{f,m(p2)}}{gva_{f(t19)}}$
t42	$gpGVA_f$	C	Ratio excédent brut d'exploitation / valeur ajoutée brute d'un navire moyen d'une flottille	-	$\frac{gp_{f(t28)}}{gva_{f(t19)}}$
t43	$gvlK_f$	C	Ratio CA / valeur d'assurance d'un navire moyen d'une flottille	-	$\frac{GVLav_{f(t7)}}{K_{f(i39)}}$
t44	gpK_f	C	Ratio excédent brut d'exploitation / valeur d'assurance d'un navire moyen d'une flottille	-	$\frac{gp_{f(t28)}}{K_{f(i39)}}$

t45	$RoFTA_f$	C	Rate of Return on Fixed Tangible Assessed	-	$\frac{ncf_f(t_{30})}{K_f(t_{39})}$
t46	ROI_f	C	Return Of Investment	-	$\frac{gp_f(t_{28}) - inv_f(t_{40})}{inv_f(t_{40})}$
t47	$ratio_{npK_f}$	C	Ratio résultat net d'exploitation / valeur d'assurance d'un navire moyen d'une flottille	-	$\frac{np_f(t_{31})}{K_f(t_{39})}$
t48	$gvlCNBue_f$	C	Ratio CA / effectif moyen / unité d'effort d'un navire moyen d'une flottille	-	$\frac{GVLav_f(t_7)}{cnb_f(t_{10}) \times ue_f(p_3)}$

TABLE 1.32: : Paramètres calculés pour le module "Economique"

1.2.6 Module "Scénario"

Le module "Scénario" va permettre à l'utilisateur d'intervenir à sa convenance sur le cours de la simulation en modulant un certain nombre de paramètres. L'action peut s'opérer sur n'importe quel paramètre initial ainsi que sur quelques paramètres internes au modèle, et ce à n'importe quel instant de la projection (hormis l'instant initial). On rappelle en outre que chaque scénario est renseigné dans un onglet spécifique du fichier de paramétrage (voir chapitre 2 sur la paramétrisation), accompagné d'une déclaration des actions à opérer (quelle(s) modulation(s) sur quelle(s) dimension(s) de quel(s) paramètre(s) et à quel(s) instant(s) de la simulation). Les paramètres internes sur lesquels il est, pour le moment, possible d'agir via le module "Scénario" incluent la mortalité ventilée ($F_{e,f,m,i}$), la mortalité "autres flottilles et autres métiers" ($F_{oth_{e,i}}$), et le chiffre d'affaires "autres espèces" ($CA_{othsf,meco}$ et CA_{othsf}).

Les différentes actions sur les variables sont renseignées comme une variable du modèle à part entière, c'est-à-dire sous la forme d'une table de valeurs (dans ce cas, des multiplicateurs appliqués aux données) intégrant les dimensions d'application (flottille, métier, âge,...), le nom de la variable impliquée, et enfin le nom du scénario dans lequel l'action devra s'opérer. Bien entendu, il est nécessaire que les dimensions de l'action coïncident avec les dimensions de la variable ciblée : on ne définira pas une action sur la dimension "métier" pour intervenir sur une variable de simple dimension "flottille". L'inclusion (mais pas nécessairement l'égalité) des dimensions "action" dans les dimensions "cible" sera donc requise. Seule la dimension temporelle pourra être intégrée dans la définition des actions sans pour autant que la variable cible n'en possède une (variable constante au cours du temps). On pourra en complément se référer au chapitre 2) dans lequel on décrira plus précisément l'onglet "Scénario" du fichier de paramétrage.

La revalorisation d'une variable dans le cadre d'un scénario est une action irrémédiable :

cela signifie que la variable conserve sa nouvelle valeur à partir de cet instant, et ce jusqu'à une hypothétique prochaine intervention. Pour illustrer ce mode de fonctionnement, prenons un exemple simple. Soit une variable "prix moyen" initiale d'une valeur de 10 euros. Supposons qu'on veuille simuler deux scénarios distincts : le premier simulant une baisse de 10% du prix moyen à l'instant t et $t + 1$, et le second simulant une baisse de 1 euro du prix moyen aux mêmes instants. Ainsi, le premier scénario fera intervenir un multiplicateur de valeur 0,9 à t et $t + 1$, alors que le second appliquera un multiplicateur de valeur 0,9 à t et de valeur 0,8888 à $t + 1$.

1.2.7 Module "Gestion"

Le module permet en l'état actuel de moduler l'effort de pêche total simulé (en jouant sur les variables de contrôle que sont l'effort (en nombre de jours de mer) ou le nombre de navires), ainsi que la mortalité par pêche des flottilles non modélisées. On peut de cette manière exercer un contrôle sur toutes les composantes de la mortalité par pêche appliquée à un stock donné. Cette modulation s'opère avec pour objectif d'atteindre des valeurs données d'une variable "cible". Il est pour le moment possible d'ajuster les variables de contrôle en vue d'atteindre une valeur de TAC (captures) ou une valeur de F_{bar} , moyenne pondérée de la mortalité par pêche (voir 1.2.1.3). L'ajustement peut se faire de façon homogène sur toutes les flottilles modélisées (variations de même proportion quelles que soient les flottilles), ou bien être obtenu par des variations distinctes de l'effort par flottille (la répartition des "efforts" consentis par chaque flottille étant renseignée par l'utilisateur au moyen d'une donnée de pondération), permettant par exemple d'allouer la réduction de mortalité par pêche par flottille en fonction de leur dépendance à une espèce ou de leur contribution à la mortalité par pêche d'une espèce. On exposera en détail les multiples options disponibles dans la partie 3 consacrée à l'utilisation du modèle.

La représentation schématique de la figure 1.2 retrace le mode de fonctionnement actuel du module de gestion pour les options "atteinte de TAC" et "atteinte de F_{bar} ". Le module de gestion intègre un algorithme de résolution qui vise à estimer le multiplicateur μ qui sera appliqué aux pondérations des variables de contrôles mF (par flottille, appliquées au nombre de navires ou au nombre de jours de mer) et $mOth$ (valeur unique appliquée à la mortalité par pêche "autres flottilles, autres métiers"), avant qu'elles mêmes soient appliquées aux variables de contrôles "initiales" (en fait issues de la simulation à l'instant précédent). Des variables de contrôle réévaluées (out) sont déduites les valeurs des variables "cible", qui peuvent être comparées avec les valeurs demandées. Les réitérations s'interrompent une fois la convergence de l'algorithme d'ajustement obtenue.

Les perspectives quant à l'enrichissement du module par de nouvelles options de gestion (potentiellement apportées par l'intermédiaire d'un module complémentaire de simulation de comportement) sont très nombreuses. Il est ainsi envisageable d'offrir la possibilité de définir des règles de variation des multiplicateurs (par exemple, en fonction de la rentabilité), ou de multiplier les paramètres "cible" (ajout de la SSB,...), ou encore d'intégrer des contraintes sur les variations d'une année sur l'autre ou sur les valeurs limites des paramètres affectés (variation annuelle majorée de TAC, seuil minimal acceptable pour l'EBE,...).

1.2.8 Module "Réplicat"

Le rôle de ce module est d'intégrer une variabilité dans les sorties du modèle en tirant parti des incertitudes considérées pour certaines variables (par le biais du module "Stochasticité",

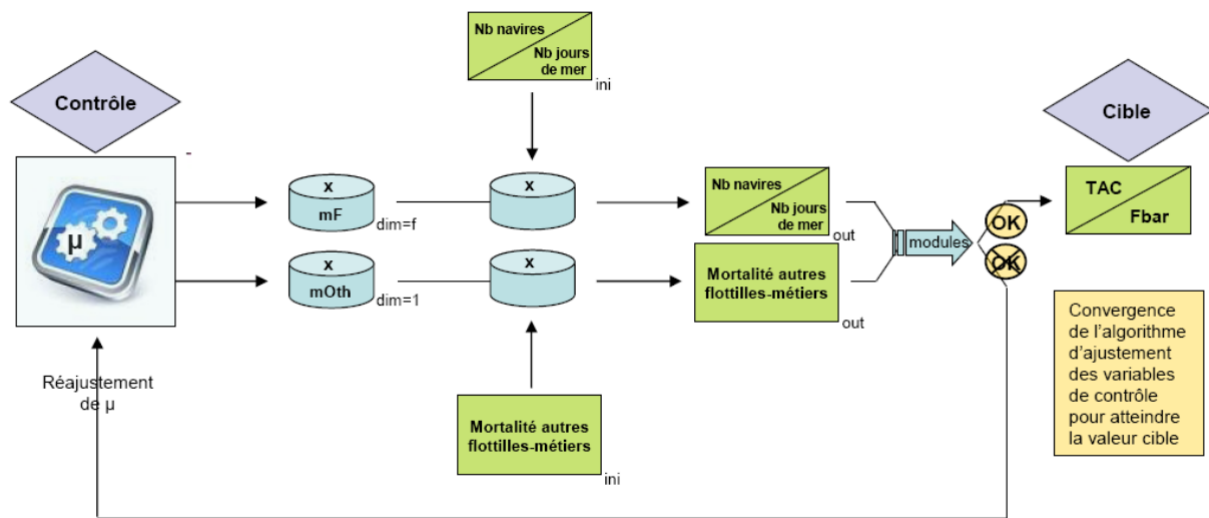


FIGURE 1.2 – Fonctionnement du module "Gestion" pour les options d'atteinte de TAC et de Fbar

l'illustration évidente étant la donnée de recrutement). Comme son nom l'indique, il va procéder par réplication multiple des simulations afin d'obtenir pour chaque indicateur un jeu de valeurs permettant, par exemple, d'aboutir à l'estimation d'une moyenne, d'une variance, ou bien encore d'un intervalle de confiance des projections.

Une schématisation de la façon dont le modèle actuel fait collaborer les modules "Recrutement" et "Réplicat" (auss appelé abusivement "Bootstrap") est représentée en figure 1.3. On distingue d'abord parmi les paramètres d'entrée les variables qui évolueront entre deux répétitions (variables stochastiques) de celles qui resteront inchangées (variables non stochastiques). Comme on l'a vu précédemment, les variables stochastiques peuvent être de trois types : des variables issues d'un tirage aléatoire dans un historique (avec possibilité de pondérer les éléments le composant), des variables incarnant une réalisation d'une variable aléatoire donnée avec des paramètres donnés, et un cas particulier de variable (de recrutement) issue d'une relation Stock-Recrutement avec addition d'un bruit paramétré. Toutes ces variables prennent part à la simulation opérée par le modèle bio-économique, et si le module "Bootstrap" est activé, un certain nombre d'itérations (déterminé par l'utilisateur) est effectué, chaque sortie étant ainsi conservée en vue des traitements post visant à considérer (ou simplement à estimer) leur variabilité.

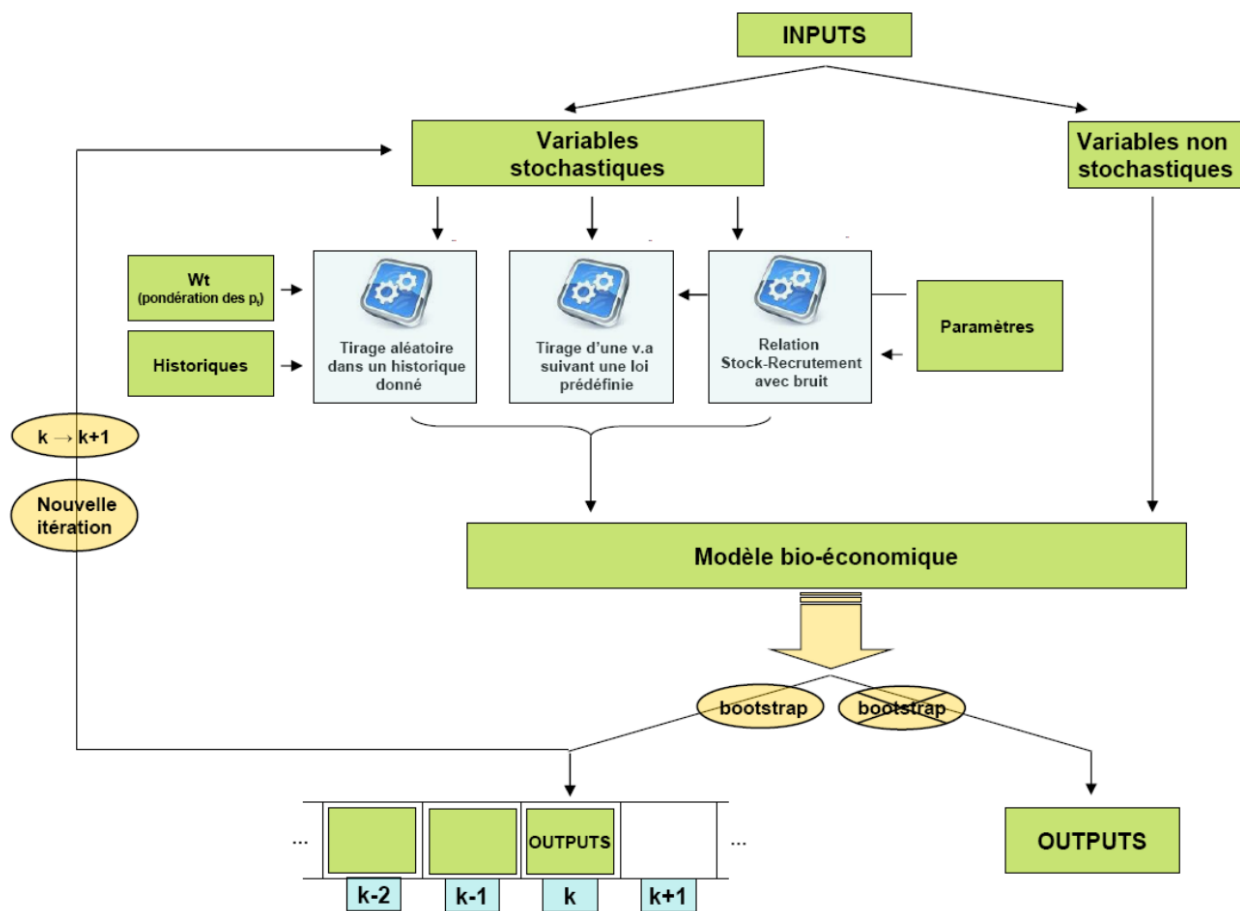


FIGURE 1.3 – Fonctionnement des modules "Recrutement" et "Bootstrap"

Chapitre 2

Elaboration d'un fichier de paramétrage fonctionnel

Un effort particulier a été mis en œuvre afin que le modèle IAM possède un système de paramétrage relativement simple à aborder, et ce malgré la diversité d'inputs qu'un modèle bio-économique multi-spécifique, multi-flottille et multi-métier induit. Pour cela, il a été convenu que l'intégralité des données d'entrée du modèle seraient contenues dans un fichier Excel d'extension ".xlsx", possédant une structure spécifique, et lisible par une procédure d'importation intégrée à l'outil (rappelons que le modèle IAM est utilisable dans un environnement R, et qu'un fichier de paramétrage type est disponible dans le package).

Le fichier d'entrée se compose des feuillets suivants : un feuillet "Stock" par espèce modélisée, un feuillet "Market", un feuillet "Flottille" par flottille modélisée, trois feuillets contenant les tables de conversions, un feuillet "Scénarii" et un feuillet "Stochasticité". Les seules données non intégrées dans le fichier et intervenant dans le modèle sont les options d'utilisation du modèle que l'on spécifiera séparément lors du lancement des simulations .

2.1 Feuillet "Stock"

Les feuillets de type "Stock" vont permettre de réunir tous les paramètres propres à chaque stock modélisé (paramètres biologiques, biomasses, effectifs totaux, mortalités par pêche,...). Il est possible de renseigner chaque paramètre de manière exhaustive au niveau historique puisque l'utilisateur peut décider du temps initial de la modélisation : ce fichier a donc aussi pour vocation d'être complété et remis à jour année après année afin que l'utilisateur aie à sa disposition une source de données à la fois évolutive et fonctionnelle.

Chacun de ces feuillets doit être nommé (click droit sur l'onglet → Renommer) selon le schéma suivant : "Stock__" + nom de l'espèce sans espace (par exemple, "Stock__Langoustine" ou "Stock__Merlu_commun").

⚠ La dénomination choisie sera également la caractérisation du stock dans le modèle (dans l'exemple, "Langoustine" ou "Merlu_commun").

En haut de la feuille, un cadre jaune en partie protégé en écriture fait office de listing des variables considérées. La colonne "Variable" n'est pas modifiable : il s'agit des intitulés de chacun des paramètres tels qu'ils sont utilisés dans l'implémentation du modèle. L'autre champ fixe est le champ "Description" qui comme son nom l'indique présente rapidement les variables. La colonne "Alias" permet d'intégrer des tables de données dans lesquelles les variables ont

un autre nom que le nom standard. Les deux colonnes "Variable" et "Alias" assurent ainsi la correspondance. Enfin, le champ "Multi" est un multiplicateur utile dans le cas où les variables rentrées ne sont pas à l'unité standard décrite dans le champ "Description". Si une variable doit s'exprimer en tonnes, et que les valeurs inscrites sont en kilogrammes, il suffit d'insérer 0,001 dans la case correspondant à la variable en question, et la correction sera opérée lors de l'importation.

⚠ Il n'est pas obligatoire de renseigner toutes les variables déclarées ici (lire le champ Description pour plus de détails). Notons également que, par souci de facilité de traitement, la liste de variables intègre aussi les variables du feuillet "Market".

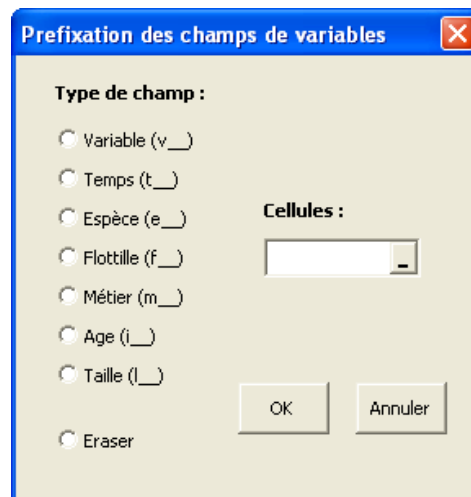
Il ne reste plus qu'à regrouper sur ce feuillet les tables susceptibles de décrire, pour l'espèce en question, les variables présentées dans le cadre jaune. Afin de faciliter ce processus, l'objectif fixé était de pouvoir gérer, lire et assurer le reformatage de tableaux de sources diverses, et de ce fait de multiples formats, tout en limitant au maximum les modifications à opérer sur les tableaux importés (ou copiés-collés) par l'utilisateur. Une macro développée spécifiquement pour ces fichiers permet de transformer les tables de paramètres insérées afin qu'elles soient lues par la procédure d'importation R. Prenons un exemple pour illustrer l'application ; soit un tableau historique d'effectifs totaux d'un stock quelconque inséré dans la feuille de calcul :

Initial Numbers (* 10 ³)											
AGE	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008
0	74985	41970	44005	57723	79148	37304	35615	34788	29203	54083	112418
1	15822	16023	13210	14162	16347	15550	10841	11411	11059	11279	22800
2	3459	4097	3881	3642	2849	3532	3085	3244	3397	3532	2982
3	986	1192	1110	1006	709	658	694	1049	1035	943	1068
4	218	283	347	278	269	214	130	220	369	298	303
5	52	68	91	99	83	93	70	44	79	126	106
6	27	15	23	22	29	25	34	30	17	28	48
7+	17	8	12	8	17	48	18	13	9	12	34

Puisque les tables peuvent se succéder sur la feuille sans ordre prédéfini, il faut indiquer quelle variable est décrite par ce tableau. Il suffit pour cela de considérer cette appellation comme une variable, au même titre que la variable "âge" déjà déclinée, et insérer un nouveau champ comme ci-dessous (nous l'appelons "ini_numbers") :

	Initial Numbers (° 10 ³)											
	AGE	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008
ini_numbers	0	74985	41970	44005	57723	79148	37304	35615	34788	29203	54083	112418
ini_numbers	1	15822	16023	13210	14162	16347	15550	10841	11411	11059	11279	22800
ini_numbers	2	3459	4097	3881	3642	2849	3532	3085	3244	3397	3532	2982
ini_numbers	3	986	1192	1110	1006	709	658	694	1049	1035	943	1068
ini_numbers	4	218	283	347	278	269	214	130	220	369	298	303
ini_numbers	5	52	68	91	99	83	93	70	44	79	126	106
ini_numbers	6	27	15	23	22	29	25	34	30	17	28	48
ini_numbers	7+	17	8	12	8	17	48	18	13	9	12	34

Il faut maintenant le convertir en un tableau que la méthode d'importation saura lire. Appelons la macro "Préfixation" (Outils → Macro → Macros → Préfixation → Exécuter). Elle se présente sous la forme d'une petite interface :



Il faut alors caractériser les variables décrites dans le tableau au moyen d'un système de préfixation. Il s'agit de sélectionner les cellules à préfixer, puis de cocher la case correspondant au type de variable en jeu, et enfin de valider en cliquant sur OK. Pour recommencer, il suffit de cliquer sur le bouton dans le champ "Cellules :". La case "Eraser" permet de corriger les erreurs et d'effacer le(s) préfixe(s) de la plage sélectionnée. Cliquer sur "Annuler" lorsque le processus est terminé. On aboutit alors au tableau suivant :

Initial Numbers (* 10 ³)												
	AGE	t_1998	t_1999	t_2000	t_2001	t_2002	t_2003	t_2004	t_2005	t_2006	t_2007	t_2008
v_ini_numbers	i_0	74985	41970	44005	57723	79148	37304	35615	34788	29203	54083	112418
v_ini_numbers	i_1	15822	16023	13210	14162	16347	15550	10841	11411	11059	11279	22800
v_ini_numbers	i_2	3459	4097	3881	3642	2849	3532	3085	3244	3397	3532	2982
v_ini_numbers	i_3	986	1192	1110	1006	709	658	694	1049	1035	943	1068
v_ini_numbers	i_4	218	283	347	278	269	214	130	220	369	298	303
v_ini_numbers	i_5	52	68	91	99	83	93	70	44	79	126	106
v_ini_numbers	i_6	27	15	23	22	29	25	34	30	17	28	48
v_ini_numbers	i_7+	17	8	12	8	17	48	18	13	9	12	34

Le préfixe est composé d'une lettre spécifique à chaque type de variable, suivi de deux symboles "_". Si besoin est, on peut rentrer l'appellation choisie pour la variable (ici "ini_numbers") dans le champ "Alias" du cadre jaune, face à la variable correspondante (ici "Nit0"). Ce tableau est maintenant lisible par la méthode d'importation et sera affecté à la variable Nit0.

Il faut encore énoncer quelques règles au sujet du format et de l'organisation des tables de paramètres.

- Ne seront considérés que les champs de paramètres préfixés, ainsi que les numériques s'y rapportant. Dans la table précédente, les cellules "AGE" et "Initial Numbers (*10³)" ne seront pas lues.
- "___" est un champ de caractères réservé aux variables : ne pas l'utiliser ailleurs dans le document (à part dans le nom des feuillets "Stock" et "Flottille", comme on va le voir).
- La dimension "Longueur" préfixée "l___" ne peut définir des variables que si une clé tailles-âges est renseignée en tant que variable "alk".
- Les sauts de lignes (de taille quelconque) servent de séparateurs entre les tables à considérer.
- Une table peut se développer sur plusieurs successions de lignes ; par exemple, cette disposition est possible :

		t_1978	t_1979	t_1980	t_1981	t_1982	t_1983	t_1984
v_Wlan_i	i_0	0.021	0.023	0.021	0.015	0.013	0.014	0.013
v_Wlan_i	i_1	0.067	0.071	0.083	0.068	0.058	0.065	0.070
v_Wlan_i	i_2	0.177	0.179	0.179	0.173	0.154	0.169	0.183
v_Wlan_i	i_3	0.357	0.354	0.357	0.358	0.360	0.340	0.337
v_Wlan_i	i_4	0.570	0.570	0.570	0.570	0.560	0.562	0.566
v_Wlan_i	i_5	0.836	0.834	0.830	0.829	0.840	0.838	0.843
v_Wlan_i	i_6	1.153	1.153	1.156	1.155	1.149	1.152	1.149
v_Wlan_i	i_7	1.513	1.517	1.516	1.519	1.517	1.514	1.516
v_Wlan_i	i_+gp	2.979	2.735	2.815	2.925	2.899	2.935	2.894

		t_1985	t_1986	t_1987	t_1988	t_1989	t_1990
v_Wlan_i	i_0	0.028	0.015	0.014	0.020	0.014	0.013
v_Wlan_i	i_1	0.077	0.086	0.058	0.070	0.091	0.065
v_Wlan_i	i_2	0.199	0.199	0.195	0.177	0.196	0.178
v_Wlan_i	i_3	0.363	0.346	0.353	0.337	0.347	0.356
v_Wlan_i	i_4	0.562	0.565	0.565	0.564	0.567	0.564
v_Wlan_i	i_5	0.835	0.837	0.836	0.835	0.839	0.841
v_Wlan_i	i_6	1.146	1.155	1.155	1.156	1.152	1.156
v_Wlan_i	i_7	1.514	1.510	1.512	1.512	1.519	1.504
v_Wlan_i	i_+gp	2.620	2.895	2.926	2.562	2.557	2.464

- au moins une variable doit être déclinée sur une colonne (comme "v" ou "i" dans la table précédente). La table suivante n'est, par exemple, pas acceptée :

v_var1	v_var2	v_var3
0.021	0.023	0.021

mais celle-ci l'est :

	v_var1	v_var2	v_var3
t_2008	0.021	0.023	0.021

Il est possible de définir les pas de temps à partir desquels définir la valeur initiale d'une variable pour laquelle un historique est disponible. La valeur initiale sera alors la moyenne sur les pas de temps choisis. Pour cela, il faut dédoubler le préfixe "t_" sur les variables temporelles constituant le recul. Par exemple, si on veut une valeur initiale égale à la moyenne des 3 derniers instants (dans notre cas, les 3 dernières années) pour un temps initial égal à 2008, on a ce tableau :

		t_2002	t_2003	t_2004	t_2005	t_t_2006	t_t_2007	t_t_2008
v__ini_numbers	i_1	560402	691143	863661	1082193	764080	887070	642616
v__ini_numbers	i_2	435448	407149	502319	620484	770523	546147	650008
v__ini_numbers	i_3	265339	223112	207306	264341	310834	348242	328988
v__ini_numbers	i_4	100194	101697	86509	87471	102982	134396	180528
v__ini_numbers	i_5	34447	43972	40123	36228	36146	44656	65279
v__ini_numbers	i_6	15082	17261	18701	17475	16549	16670	23173
v__ini_numbers	i_7	6747	8128	8156	9013	8251	8236	8304
v__ini_numbers	i_8	3963	4019	4050	3794	4770	3980	4257
v__ini_numbers	i_+gp	4089	5426	6406	5580	7858	6174	4679

⚠ S'il a lieu d'être, le dédoublement du préfixe temporel doit au moins être affecté à l'instant initial "t_init", et potentiellement à des instants qui lui sont antérieurs. Pas de dédoublement signifie que le recul ne portera que sur l'instant initial. Notons que l'instant initial considéré sera déclaré lors de l'importation des paramètres. Toutes ces tables seront alors lues par la méthode d'importation et regroupées dans un objet R de type "liste".

2.2 Feuillet "Market"

Le feuillet "Market" diffère sensiblement des feuillets "Stock" par sa présentation générale, qui est similaire au format des feuillets "Flottille". Ceci s'explique par le fait que contrairement aux données de type "stock" qui proviennent le plus souvent de sources diverses (groupes d'évaluation, expertises,...), les variables "Market" sont générées (au même titre que les variables économiques regroupées dans les feuillets "Flottille") par des routines d'extraction et de mise en forme standardisées. Il était donc plus facile de conserver ce format au sein du fichier de paramétrage. On se référera donc au paragraphe décrivant le feuillet "Flottille" pour plus de précision. Notons simplement que ce feuillet comprend une colonne supplémentaire intitulée "categorie", pour des variables préfixées "c__".

2.3 Feuillet "Flottille"

Ces feuillets vont regrouper les données économiques et d'activités des flottilles modélisées. Comme on vient de le préciser pour le feuillet "Market", toutes ces informations sont disponibles grâce à une routine développée en langage R, qui restitue la donnée par flottille dans un unique format stable et reconnu par la méthode d'importation. Il suffira alors de compiler toutes les tables ainsi obtenues dans le fichier Excel, en conservant la distinction "flottille" (1 feuillet par flottille) ou non, la seule restriction étant de conserver l'architecture standard des tables "flottille". Nous présentons maintenant ce format spécifique.

Ce tableau comporte 7 colonnes principales d'en-têtes *nom_variable*, *libelle*, *type d'indicateur*, *flottille*, *metier*, *espece*, et *indicateur*, dont voici la description :

- **nom_variable** : intitulé (préfixé) de la variable décrite tel qu'il sera utilisé dans l'implémentation du modèle (ôté du préfixe caractéristique des variables "v__")
- **libelle** : descriptif de la variable
- **type d'indicateur** : catégorisation de l'indicateur
- **flottille** : champ facultatif. Variable flottille, préfixée "f__".
- **metier** : champ facultatif. Variable métier, préfixée "m__". Le niveau considéré est le niveau "métier éco", potentiellement différent du niveau considéré dans les feuillets "Stock".
- **espece** : champ facultatif. Variable espèce, préfixée "e__".
- **indicateur** : champ facultatif. Valeur numérique de la variable.
- ... : D'autres champs peuvent être ajoutés à droite de ces 7 premières colonnes, mais ils n'interviendront pas dans le processus de modélisation.

Les cellules non renseignées doivent rester vides. Toutes ces informations lues par la routine d'importation seront regroupées dans un nouvel élément "Fleet" de la liste en sortie. Seules les variables déclinées par espèce seront intégrées à l'élément "Stock" généré par les feuillets correspondants (cf section 2.1). Au même titre que les variables flottille, métier, temps,... les variables espèces doivent être consistantes dans l'intégralité du fichier. En conséquence, les dénominations des variables préfixées "e__..." doivent être similaires à celle utilisées dans les noms des feuillets "Stocks" (par exemple, "Stock__Sole_commune" et "e__Sole_commune").

A titre indicatif, un document "FleetSpreadsheet" de tables types regroupant toutes les variables d'activité à renseigner est disponible au sein du package IAM. Il comprend une table complète assurant le paramétrage du modèle économique complet, ainsi qu'une deuxième table pour le modèle économique DCF.

2.4 Feuillet "fm_matrix"

Ce feuillet réunit les tables de distribution flottilles-métiers "bio" qui seront utilisées lors de l'étape de ventilation de la variable de mortalité par pêche. Ce processus pourra nécessiter en effet l'apport de la donnée de débarquements totaux par espèce au niveau flottille-métier "stock". Cette donnée n'étant disponible qu'au niveau flottille et flottille-métier "éco" dans les feuillets "Flottille" (variable *Lref*), une table d'allocation des débarquements par flottille pour chaque métier "stock" sera utilisée. Le feuillet contiendra une table par espèce modélisée (les métiers pouvant différer entre les espèces considérées), utilisant les notations habituelles décrites précédemment. Les valeurs peuvent être considérées comme les probabilités $P(\text{metier}|\text{flottille} \cdot \text{espece})$.

			m_metier1	m_metier2	m_metier3
v_fm	e_species	f_fleet1	1		
v_fm	e_species	f_fleet2	1		
v_fm	e_species	f_fleet3		1	
v_fm	e_species	f_fleet4		1	
v_fm	e_species	f_fleet5		0.5	0.5
v_fm	e_species	f_fleet6			1
v_fm	e_species	f_fleet7			1
v_fm	e_species	f_fleet8			1

FIGURE 2.1 – Exemple illustratif d'une table "fm" pour une espèce donnée

2.5 Feuillet "mm_matrix"

Cette feuille intègre la matrice de transition entre le niveau métier "stock" utilisé par le modèle biologique, et le niveau métier "éco" qui caractérise ensuite le modèle économique. Elle sera utilisée au cours du processus engagé par le module "Market" (qui, on le rappelle, assure le lien entre les deux modèles, en redéfinissant notamment la dimension "métier"). La matrice contient ainsi des valeurs apparentées aux probabilités $P(\text{espece} \cdot \text{flottille} \cdot \text{metier}_{eco} | \text{espece} \cdot \text{flottille} \cdot \text{metier}_{stock})$.

2.6 Feuillet "icat_matrix"

Tout comme le feuillet "mm_matrix", la donnée incluse dans cette feuille sera utilisée par le module "Market" pour procéder à un redimensionnement des variables. En l'occurrence, les tables "icat" vont participer à la conversion de certaines variables aux âges en variable par catégorie commerciale, permettant ainsi leur intégration ou leur application aux modèles de prix. Ces tableaux vont définir la distribution des catégories aux âges, et ceci par espèce. En d'autres termes, ils contiendront les probabilités $P(\text{espece} \cdot \text{categoriecommerciale} | \text{espece} \cdot \text{age})$.

				m_mEco1	m_mEco2	m_mEco3
v_mm	e_spp1	f_fleet1	m_mBio1	1		
v_mm	e_spp1	f_fleet1	m_mBio2	1		
v_mm	e_spp1	f_fleet1	m_mBio3		1	
v_mm	e_spp1	f_fleet1	m_mBio4		1	
v_mm	e_spp1	f_fleet2	m_mBio1		0.5	0.5
v_mm	e_spp1	f_fleet2	m_mBio2			1
v_mm	e_spp1	f_fleet2	m_mBio3			1
v_mm	e_spp1	f_fleet2	m_mBio4			1
v_mm	e_spp2	f_fleet1	m_mBio1	1		
v_mm	e_spp2	f_fleet1	m_mBio2	1		
v_mm	e_spp2	f_fleet1	m_mBio5		0.6	0.4
v_mm	e_spp2	f_fleet1	m_mBio6		0.3	0.7
v_mm	e_spp2	f_fleet2	m_mBio1		0.5	0.5
v_mm	e_spp2	f_fleet2	m_mBio2			1
v_mm	e_spp2	f_fleet2	m_mBio5		0.1	0.9
v_mm	e_spp2	f_fleet2	m_mBio6		0.1	0.9

FIGURE 2.2 – Exemple illustratif d'une table "mm"

			c_10	c_20	c_30	c_40	c_50
v_icat	e_species	i_0					1
v_icat	e_species	i_1					1
v_icat	e_species	i_2				1	
v_icat	e_species	i_3				1	
v_icat	e_species	i_4				1	
v_icat	e_species	i_5			1		
v_icat	e_species	i_6		1			
v_icat	e_species	i_7		1			
v_icat	e_species	i_8+	1				

FIGURE 2.3 – Exemple illustratif d'une table "icat"

2.7 Feuille "Scénarii"

Dans ce feuillet vont être consignées toutes les informations caractérisant les scénarios basés sur une intervention prédéfinie de l'utilisateur sur une ou plusieurs variables données (cf section 1.2.6). Pour chacun des scénarios envisagés, une série de tables définissant les actions à opérer vont être insérées. Chacune d'entre elles décrit la variable à modifier, les dimensions sur lesquelles porte cette modification (pour quelle espèce, quelle flottille, quel métier, à quel instant,... ?), et bien évidemment le multiplicateur à appliquer.

L'en-tête du feuillet est une partie bloquée en écriture qui présente les variables d'entrée

et les variables internes qui pourront faire l'objet d'un réajustement au cours de la simulation. Le nom de la variable composant chaque tableau devra donc se trouver dans cette liste. La première colonne de la feuille sert à caractériser le scénario : c'est ici que la première colonne de la table devra être inscrite, contenant le nom du scénario considéré. Une même table peut s'appliquer à plusieurs scénarios. Il suffit pour cela de séparer les noms de scénarios par "&". Les colonnes suivantes constituent le tableau de valeurs, lequel suit les mêmes règles de constitution de tables que dans les feuillets "Stock" (cf section 2.1).

Notons pour finir qu'il est possible d'intervenir implicitement sur toutes les modalités d'une dimension donnée si on l'occulte dans la table correspondante. Par exemple, prenons une variable de dimension "flottille-métier-âge". Si la table ne se décline que suivant les dimensions "flottille" et "âge", les multiplicateurs s'appliqueront à tous les métiers pour la flottille et l'âge correspondant.

					t_2015	t_2016
scenario1 & scenario2	v_F_fmi	e_species	f_flottille1	i_0	0.9	0.9
scenario1 & scenario2	v_F_fmi	e_species	f_flottille1	i_1	0.8	1
scenario1 & scenario2	v_F_fmi	e_species	f_flottille1	i_2	0.7	1
scenario1 & scenario2	v_F_fmi	e_species	f_flottille1	i_3	0.8	1
scenario1 & scenario2	v_F_fmi	e_species	f_flottille1	i_4	0.9	1
scenario1 & scenario2	v_F_fmi	e_species	f_flottille1	i_5+	1	1

			t_2016	t_2017
scenario2	v_fc_f	f_flottille1	1.05	1.05
scenario2	v_fc_f	f_flottille2	1.1	1.1
scenario2	v_fc_f	f_flottille3	1.1	1.1
scenario2	v_fc_f	f_flottille4	1.1	1.1
scenario2	v_fc_f	f_flottille5	1.05	1.05
scenario2	v_fc_f	f_flottille6	1.05	1.05

FIGURE 2.4 – Exemple illustratif d'un jeu de tables définissant 2 scénarios distincts

2.8 Feuille "Stochasticité"

Cette feuille regroupe les paramètres régissant l'intégration de l'aspect stochastique dans le modèle, par l'intermédiaire de la donnée de recrutement (dans les futurs développements, la portée de la dimension aléatoire sera élargie à d'autres variables). Deux méthodes de simulation de recrutement sont paramétrables via cette feuille : un recrutement obtenu par tirage aléatoire dans un historique (la donnée doit être renseignée entre les deux balises "Samples" et "END Samples" dans une table préformatée, voir section 2.8.1), ou un recrutement comme étant la réalisation d'une variable aléatoire de loi paramétrable (la donnée doit être renseignée entre les deux balises "Random-variate" et "END Random-variate" dans une autre table préformatée, voir section 2.8.2).

2.8.1 Table "Samples"

Ce tableau comporte 5 colonnes d'en-têtes *Species*, *TimeStep*, *Value*, *Mult*, et *Weighted* :

- **Species** : intitulé (préfixé) de l'espèce à laquelle se rapporte la donnée de recrutement.
- **TimeStep** : année du recrutement spécifié (non préfixé car n'intervient pas dans l'implémentation)
- **Value** : valeur de recrutement
- **Mult** : Multiplicateur de valeur. Si *Value* est en milliers d'individus, *Mult* devra être égal à 1000. La donnée finale doit toujours être en nombre d'individus.
- **Weighted** : Poids affecté à chaque année de recrutement dans le processus de tirage aléatoire. Permet de pondérer les chances d'être tiré au sort. Ces poids doivent être des entiers positifs (potentiellement nuls si on désire écarter certaines valeurs).

L'activation ou la désactivation de ces méthodes se décidera lors de la définition des "arguments" envoyés au modèle .

Species	TimeStep	Value	Mult	Weighted
e_spp1	2000	666544	1000	1
e_spp1	2001	597148	1000	1
e_spp1	2002	560402	1000	1
e_spp1	2003	691143	1000	1
e_spp1	2004	863661	1000	1
e_spp1	2005	1082193	1000	1
e_spp1	2006	764080	1000	1
e_spp2	1996	283968	1000	1
e_spp2	1997	236473	1000	1
e_spp2	1998	232400	1000	1
e_spp2	1999	187920	1000	1
e_spp2	2000	182864	1000	1
e_spp2	2001	216737	1000	1
e_spp2	2002	186543	1000	1
e_spp2	2003	191367	1000	1
e_spp2	2004	257452	1000	2
e_spp2	2005	239724	1000	2
e_spp2	2006	215571	1000	2

FIGURE 2.5 – Exemple d'une table "Samples". Ici, un historique sur deux espèces, avec tirage équiprobable sur la période 2000-2006 pour la première, et tirage sur 1996-2006 pondéré à l'avantage des 3 dernières années pour la seconde.

2.8.2 Table "Random-variate"

Cette table va caractériser les paramètres de la variable aléatoire dont la donnée de recrutement sera une réalisation. Elle comprend 5 champs qui sont la loi de distribution, les paramètres de cette loi (3 paramètres maximum à renseigner selon la loi), et l'espèce à laquelle s'applique cette loi de distribution de recrutement (ne pas oublier le préfixe "e__" pour ce champ). L'en-tête du feuillet retrace les lois de probabilités potentiellement utilisables, ainsi que les noms des paramètres à renseigner pour chacune d'entre elles. De futurs développements permettront d'étendre cette application à d'autres variables que le simple recrutement.

Distribution	Parameter 1	Parameter 2	Parameter 3	Species
lnorm	19.2525924	0.248148268		e_spp

FIGURE 2.6 – Exemple d’une table "Random-variate". Ici, le recrutement de l’espèce considérée est une v.a de loi lognormale, et de paramètres $\logmean = 19,25\dots$ et $\logsd = 0,248\dots$ (cf en-tête du feuillet)

Chapitre 3

Vignettes

Table des matières du chapitre

3.1	Utilisation d'IAM dans R	54
3.1.1	Choix d'implémentation	54
3.1.2	Le package IAM (version 2.0.0)	54
3.1.2.1	Installation et chargement du package	54
3.1.2.2	Importation des paramètres (IAM.input)	55
3.1.2.3	Constitution des arguments (IAM.args)	57
3.1.2.4	Simulation (IAM.model)	60
3.1.2.5	Mise en forme des sorties (IAM.format & IAM.format_quant)	62
3.1.3	Représentations graphiques des sorties avec ggplot2	64
3.1.4	Conseil de réplication	66
3.1.5	Références	67
3.2	Gestion avec TAC	70
3.2.1	Scénario <i>statu quo</i>	71
3.2.2	Gestion TAC générale	72
3.2.2.1	Représentation graphiques	73
3.2.2.2	Pondérer les efforts avec la matrice mfm	76
3.3	Scénarios TAC, quotas et transition F_{msy}	80
3.3.1	Scénario <i>statu quo</i>	81
3.3.2	Gestion TAC générale	82
3.3.2.1	Représentation graphiques	83
3.3.3	TAC jusqu'au F_{msy}	86
3.3.3.1	Représentation graphiques	86
3.3.3.2	Calcul de la probabilité d'atteindre le F_{msy}	86

Les vignettes présentées dans ce chapitre sont associées au package IAM et sont accessibles sur le site¹ hébergeant toute la documentation. Composées en partie de code R réévalué à chaque rendu du document, cette version pdf peut contenir du code ne fonctionnant plus. Dans le cas où une indication ne fonctionnerait pas dans ce document, il est préférable de se référer à la documentation en ligne.

Ces vignettes sont également disponibles après installation du package, en effectuant le code suivant :

1. <https://ifremer-iam.github.io/IAM/>



```
1 # Instalation du package
2 install.packages("IAM", repos = "https://ifremer-iam.r-universe.dev")
3 # Ouverture de la vignette "Utilisation d'IAM dans R"
4 vignette("use_IAM", package = "IAM")
```

Utilisation IAM dans R (fr)

Le modèle bio-économique IAM a été développé dans l'optique d'être utilisable sous un environnement R (R Core Team 2021). L'outil est ainsi accessible par l'intermédiaire d'un package R intégrant un ensemble de méthodes couvrant toutes les étapes de la modélisation, de la phase de paramétrage à la restitution des sorties du modèle. Cette vignette traite des possibilités offertes par les outils du package, et de la manière dont on peut les utiliser pour procéder de manière relativement simple et rapide à des simulations et aux traitements des résultats.

Choix d'implémentation

Les choix d'implémentation du modèle IAM ont découlé d'exigences techniques fixées dans le but de maximiser la pérennité et la fonctionnalité du modèle. Elles portaient à la fois sur l'aspect purement technique du programme (rapidité, qualité des algorithmes,...), sur sa flexibilité (généricité et robustesse face à la multiplicité des configurations d'inputs prises en charge, diversité des scénarios proposés, choix des bases de données d'entrée, diversité des outils satellites au modèle proprement dit,...) ou encore sur son accessibilité (simplicité d'utilisation, logiciels gratuits et facilement téléchargeables, installation simple, méthodes à la fois génériques et paramétrables,...).

Le langage R réunit une grande partie de ces caractéristiques. C'est un logiciel libre et gratuit de traitement statistique, largement répandu à l'heure actuelle au sein de la communauté halieutique et qui offre l'avantage d'être relativement simple d'emploi. Il représente un environnement adéquat pour gérer des objets de paramétrage (notamment multidimensionnels, comme le sont ceux du modèle bio-économique). De surcroît, les sorties exportées en tant qu'objets R peuvent bénéficier des nombreuses fonctionnalités proposées par le logiciel (sorties graphiques, traitements statistiques,...).

Toutefois, il est apparu que le modèle deviendrait très rapidement exigeant en terme de temps calcul (algorithmes d'optimisation, beaucoup de variables, beaucoup de dimensions,...) et que l'utilisateur trouverait là la limite des possibilités offertes par R. La solution la plus évidente était de programmer le cœur du modèle en C++, et de laisser R piloter les routines, se contentant ainsi de gérer le paramétrage et le traitement des sorties. Ainsi, le modèle serait aussi simple d'utilisation que possible, sans toutefois que cela se fasse au détriment de ses performances et de sa flexibilité.

Le package IAM (version 2.0.0)

Ce qui suit va être une description par l'exemple des outils contenus dans la version 2.0.0 du package IAM. Il faut garder à l'esprit que ce package étant encore en cours de développement, certains des descriptifs présentés ici pourront devenir obsolètes dans les prochaines versions. Ajoutons que toute cette partie suppose une connaissance minimale du logiciel R (installation d'un package, utilisation de la fenêtre de commande,...).

Installation et chargement du package

L'installation du package IAM se fait de la même manière que tout autre package du CRAN, à la différence que le dépôt doit être précisé. D'autres méthodes d'installations sont développées dans le README.md du gitlab hébergeant le code source. Ces méthodes peuvent être utiles dans le cas d'installation de versions précédentes ou en cours de développement actif.

```
install.packages("IAM", repos = "https://ifremer-iam.r-universe.dev")
```

Après avoir installé le package IAM, l'instruction suivante va charger le package en session, donnant accès à tous les outils de modélisation :

```
library(IAM)
```

On peut vérifier que le package est bien installé en tapant :

```
help(IAM) # alternative : ?IAM  
help(package = IAM)
```

Ces commandes permettent d'accéder à l'aide du package, avec une aide détaillée pour chaque fonction. S'y ajoute également l'ensemble des méthodes et des classes d'objets dorénavant accessibles.

Importation des paramètres (IAM.input)

La package intègre un fichier exemple de paramétrage qui va permettre d'illustrer le fonctionnement des procédures d'importation. La fonction `IAM_example()` renvoie le chemin du fichier d'exemple choisis dans le package IAM. Il s'agit de données brutes au format `.xlsx` ou `.RData`.

Ces données brutes sont constituées d'un fichier `.xlsx`, d'une fichier `.RData` contenant des informations pour une espèce SS3 ainsi qu'un dossier `fleet/` contenant les informations sur les structures de coûts de plusieurs flotilles types.

```
IAM_example("inputFile.xlsx") # attention ici il s'agit d'un fichier brut.  
# ce fichier est accessible a cette adresse pour être lu et modifié au besoin  
list.files(IAM_example(file = "."), recursive = TRUE)  
# listes des fichiers accessibles avec IAM_example()
```

Le package contient également des objets intermédiaires issus des différentes fonctions. On peut accéder à la liste des fichiers d'exemples et leur documentation avec les commandes suivantes :

```
data(package = "IAM")
```

La méthode `IAM.input()` sert à l'importation des paramètres d'entrée bruts du modèle, et retourne un objet de classe `iamInput` regroupant de manière organisée toute ces données. Elle prend pour argument principal le chemin du fichier `.xlsx` à importer, complété par `t_init` l'instant initial, et `nbStep` le nombre de pas de temps désiré pour la simulation.

Dans le cadre de cet exemple, il faut cependant noter quelques modifications. En effet ici les flotilles ne sont pas comprises dans le fichier `.xlsx` et sont donc à aller chercher dans un dossier flottille avec l'argument `folderFleet`.

De même, une espèce décrite dans le fichier de paramétrage utilise un modèle de dynamique dit *SS3*, et les informations sont donc introduites à ce moment là. Il s'agit de mortalité définies par saisons.

On procède alors à l'importation de la manière suivante (*attention, cela peut prendre quelques secondes selon le nombre de flottilles*):

```
load(IAM_example("IAM_SS3_2009.RData"))  
  
IAM_input_2009 <- IAM::IAM.input(  
  fileIN = IAM_example("inputFile.xlsx"),  
  t_init = 2009, nbStep = 12, folderFleet = IAM_example("fleets"),  
  # elements  
  Fq_i = list(DAR = iniFq_i), iniFq_i = list(DAR = iniFq_i),  
  Fq_fmi = list(DAR = iniFq_fmi), iniFq_fmi = list(DAR = iniFq_fmi),  
  FqLwt_i = list(DAR = iniFqLwt_i), iniFqLwt_i = list(DAR = iniFqLwt_i),  
  FqLwt_fmi = list(DAR = iniFqLwt_fmi), iniFqLwt_fmi = list(DAR = iniFqLwt_fmi),  
  FqDwt_i = list(DAR = iniFqDwt_i), iniFqDwt_i = list(DAR = iniFqDwt_i),  
  FqDwt_fmi = list(DAR = iniFqDwt_fmi), iniFqDwt_fmi = list(DAR = iniFqDwt_fmi),  
  Nt0s1q = list(DAR = Nt0s1q), Ni0q = list(DAR = Ni0q),
```

```
iniNt0q = list(DAR = iniNt0q), matwt = list(DAR = mat_morphage)
)
```

L'objet résultant (ici `IAM_input_2009`) de classe `iamInput` possède une structure fixe spécifique, composée d'éléments appelés "slots" et que l'on peut distinguer de cette façon :

```
slotNames(IAM_input_2009)
#> [1] "desc"      "specific"   "historical" "input"      "scenario"
#> [6] "stochastic" "optimization"
```

Le slot "desc" est un simple descripteur permettant de caractériser l'objet ; c'est un paramètre optionnel de la fonction `IAM.input`. Le slot "specific" est constitué des descriptifs des dimensions du modèle qui vont conditionner la structure des variables. Le slot "historical" regroupe les données antérieures à l'instant initial `t_init`. Les slots "scenario" et "stochastic" intègrent les paramètres définis dans les feuillets correspondants. Enfin, le slot principal "input" rassemble les données initiales injectées dans le modèle. Il est composé d'un élément "Fleet" et d'un élément par espèce modélisée, ces derniers possédant la même structure. Au final, ce slot rassemble toutes les informations renseignées dans les feuillets "Stock," "Flottille" et "Market," ainsi que les matrices "fm," "mm" et "icat." Voici quelques exemples permettant d'accéder aux données de cet objet (l'accès aux slots se fait par `objet@slot`, l'accès aux éléments d'un slot par `slot$élément`) :

```
names(IAM_input_2009@specific)
#> [1] "Species" "StaticSpp" "AllSpp" "Fleet" "Metier" "MetierEco"
#> [7] "Ages" "Cat" "t_init" "NbSteps" "times" "Q"
#> [13] "S"
names(IAM_input_2009@input)
#> [1] "ARC" "COR" "DAR" "GAR" "GLY" "HER" "MAH" "DOR" "PAV"
#> [10] "INO" "TUL" "PAM" "MON" "OED" "HAY" "FET" "RHO" "IPH"
#> [19] "THY" "OCH" "VAU" "ZZZ" "Fleet"
IAM_input_2009@input$COR$wStock_i
#> 2 3 4 5 6 7 +gp
#> 0.200 0.252 0.302 0.373 0.394 0.405 0.555
#> attr(,"DimCst")
#> [1] 0 0 7 0
attributes(IAM_input_2009@input$COR$wStock_i)$DimCst[3]
#> [1] 7
```

On peut ici noter l'attribut `DimCst` associé à la variable `wStock_i`. Chaque variable d'entrée insérée dans le slot `input` est affectée de cet attribut qui définit sa dimension : la première composante est la dimension flottille (0 signifie "pas de dimension"), la seconde est la dimension métier, la troisième est la dimension âge/catégorie, et la quatrième est la dimension temporelle. Dans l'exemple précédent, la variable se décline uniquement suivant la dimension "âge," qui comprend pour l'espèce considérée 7 modalités. Cette attribution systématique permet au modèle de gérer les incompatibilités de dimensions entre les variables et de considérer plusieurs niveaux de définition pour une même variable.

Afin de simplifier l'utilisation de cette classe d'objets, une méthode `summary()` a été implémentée qui permet d'avoir un aperçu rapide :

```
summary(IAM_input_2009)
#> My Input (IAM input) :
#> Simulation of 3 dynamic species, 19 static species and 7 fleet
#> Simulation start in 2009 and end in 2020 (12 steps)
#>
#> -----
#> Dynamic Species | Model | Ages |
#> ARC | XSA | 0 to +gp |
#> COR | XSA | 2 to +gp |
```



```
#>          DAR /   SS3 / 0 to +gp /
#> -----
#>          Fleet /   nbv   /
#>          Alis /    24   /
#>          Antea /    36   /
#>          Atalante /    15  /
#>          Haliotis /     5  /
#>          Marion_Dufresne /    9  /
#>          Pourquoi_pas /    18  /
#>          Thalassa /    60  /
```

Constitution des arguments (IAM.args)

La méthode `IAM.args()` prend en paramètre un objet de classe `iamInput` et permet à l'utilisateur de fixer les caractéristiques de la simulation à opérer. Une interface de paramétrage s'affiche, et la validation renvoie un objet de classe `iamArgs` définissant les choix de pilotage du modèle.

```
IAM_argum_2009 <- IAM.args(IAM_input_2009)
```

L'instruction va provoquer l'affichage d'une interface de paramétrage comportant 5 parties distinctes, chacune activant ou/et paramétrant une partie spécifique du modèle.

- **Recruitment** : ce panneau de commande de type "tabbed notebook" permet de définir l'intégration de la variabilité dans le modèle en caractérisant le recrutement considéré dans la simulation. Il y a un onglet par espèce modélisée, chacun donnant accès à un choix de paramètres similaires. On pourra soit opter pour un recrutement généré par une relation stock-recrutement (activer alors le panneau en cochant "Stock Recruitment Model," soit simuler un recrutement par tirage aléatoire dans un historique ou une loi de probabilité (cocher "StockSim"). Dans le premier cas, 5 paramètres sont à définir : le type de relation stock-recrutement, les paramètres de la relation (3 au maximum), et l'écart-type du bruit gaussien centré additionnel. Les relations sont décrites dans le tableau suivant :

Relation Stock-Recrutement	Formulation (σ est le bruit gaussien)
Mean	$rec \sim a + \sigma$
Hockey Stick	si $ssb \leq b$: $rec \sim a \cdot ssb + \sigma$ sinon : $rec \sim a \cdot b + \sigma$
Beverton-Holt	$rec \sim \frac{a \cdot ssb}{b + ssb} + \sigma$
Ricker	$rec \sim a \cdot ssb \cdot e^{-b \cdot ssb} + \sigma$
Shepherd	$rec \sim \frac{a \cdot ssb}{1 + (\frac{ssb}{b})^c} + \sigma$
Hockey Stick Quadratic	si $ssb \leq b \cdot (1 - c)$: $rec \sim a \cdot ssb + \sigma$ si $ssb \geq b \cdot (1 - c)$: $rec \sim a \cdot b + \sigma$ sinon : $rec \sim a \cdot (ssb - \frac{(ssb - b \cdot (1 - c))^2}{4 \cdot b \cdot c}) + \sigma$
Hockey Stick Smooth	$rec \sim a \cdot \{ssb + \sqrt{b^2 + g} - \sqrt{(ssb - b)^2 + g}\} + \sigma$ avec $g = 0.001$

Il est possible d'annuler l'effet aléatoire en mettant la variable σ à 0 (pas de variance, donc pas de variabilité). Précisons aussi que le recrutement s'exprime ici en nombre d'individus. La deuxième possibilité est de considérer le recrutement comme étant la réalisation d'un tirage aléatoire. Trois options sont alors envisageables :

- l'option 1 permet d'effectuer un tirage aléatoire dans l'historique de recrutement renseigné dans le feuillet "Stochasticité" du fichier de paramétrage, de manière indépendante (avec toutefois la pondération des probabilités spécifiée dans le fichier .xlsx).
- l'option 2 est similaire à l'option 1, à la différence que le tirage ne sera pas indépendant. Pour chaque instant de la simulation, un seul tirage sera effectué pour toutes les espèces d'option 2 de

recrutement pour déterminer un instant commun de l'historique à considérer. Notons que ce type de tirage impliquera seulement l'intersection des historiques pour les espèces concernées.

- enfin, l'option 3 offre la possibilité de simuler le recrutement d'une espèce comme étant la réalisation d'une variable aléatoire de loi prédéterminée. Cette loi devra également être préalablement définie au sein du feuillet "Stochasticité" du fichier de paramètres.
- **Replicates** : cette partie de l'interface permet de piloter le module de réplication. Pour activer/désactiver le module, cocher "activate." Une fois le module activé, on peut paramétrer le nombre d'itérations à opérer, et choisir les variables en sortie. Cette option permet de limiter la taille des résultats de simulations qui peuvent devenir très volumineux si on multiplie inconsidérément le nombre de réplicats. Ce sera surtout le cas avec des variables à haut niveau de précision telles que `L_efmit`.
- **Scenario** : cette partie de l'interface permet de choisir le scénario à simuler, ce dernier étant caractérisé dans le feuillet "Scénario" du fichier de paramétrage. Pour activer/désactiver le module, cocher "activate." Une fois le module activé, un scénario devra être sélectionné dans la liste (par définition, un scénario Statu Quo impliquera la désactivation du module). Afin de simplifier la mise en oeuvre des simulations et la gestion des sorties du modèle, on ne peut simuler qu'un seul scénario par envoi. A noter que ces scénarios sont plus difficiles à régler et demandent une bonne compréhension du modèle par rapport à l'utilisation de scénario par la partie "Management" du modèle et les scénarios de TAC.
- **Management** : le module "Management" peut être piloté intégralement par cette composante de l'interface pour les scénarios de TAC totaux. Un paramétrage de scénarios TAC par flottilles par exemple requiert de paramétrer la fonction `IAM.model()` et fait l'objet d'une autre vignette. Tout comme pour les deux précédents contrôles, on activera le module en cochant "activate," permettant ainsi l'accès aux champs de paramètres à renseigner suivants :
 - *Control* : permet de spécifier sur quelle variable d'effort agir afin d'atteindre les objectifs de gestion (pour le moment, laisse le choix entre le nombre de navires et le nombre de jours de mer).
 - *Target* : variable sur laquelle porte les objectifs définis (3 options : atteinte du TAC, atteinte du F_{bar} , ou atteinte du TAC jusqu'à atteindre le F_{bar} , puis atteinte du F_{bar}).
 - Bouton *TAC/ F_{bar}* : permet de renseigner les valeurs-objectifs de TAC et de F_{bar} pour l'espèce en question, sur l'intervalle de temps spécifié dans le paramétrage (une cellule par pas de temps). Il est conseillé de renseigner toutes les cellules de la table, en particulier pour une cible de type 3 (voir plus haut). La validation des valeurs s'opère à la fermeture de la fenêtre.
 - *Species* : espèce sur laquelle porte la mesure de gestion (à choisir entre toutes les espèces modélisées)
 - *Type* : multiplicatif ou additif.
 - *Delay* : délai d'application de la mesure de gestion (autrement dit, temps d'attente avant la mise en activation du module).
 - *Update* : définit si l'ajustement s'opère à chaque fois sur les valeurs initiales des variables, ou sur les valeurs mises à jour. (NB: cette option n'impacte que sur la manière de quantifier les ajustements, et ne revêt de l'importance que dans des cadres d'utilisation très spécifiques).
 - *Upper bound / Lower bound* : bornes de l'intervalle d'application de la procédure d'optimisation. Un intervalle restreint permet de maximiser le temps de calcul, mais fait prendre le risque de ne pas trouver de solution. Ces valeurs sont fortement dépendantes du type de gestion réglé. A utiliser avec précaution.
 - Bouton *Eff table* : effort maximum par flottille.
 - Bouton *mFm table* : permet de renseigner les pondérations (décimales) par flottille utilisées durant le processus d'ajustement (une cellule par classe de flottille). Cette pondération permet de paramétrer la distribution d'effort concédée par chaque flottille afin d'atteindre l'objectif désiré. Un poids nul signifiera que la flottille en question conservera son effort initial. Un poids p signifiera

que la variation de l'effort consentie par la flottille pour atteindre au final la valeur objectif sera p fois plus importante que celle des flottilles de poids unité (attention : tout ceci s'applique relativement au signe de variation global, qui peut être contrôlé par le biais des bornes supérieures et inférieures préalablement définies). Des poids de signes opposés peuvent être utilisés pour simuler des variations contraires selon les flottilles.

- **Economic** : la dernière phase de paramétrisation concerne le modèle économique. La plupart des paramètres ici décrits peuvent être retrouvés dans le tableau de mise en équation du modèle économique en annexe de ce document.
 - *perscCalc* : Mode de calcul de la variable “coût du personnel”
 - * 0 : la variable est supposée constante
 - * 1 : relation linéaire avec *rtbs* et donnée *cs* disponible
 - * 2 : relation linéaire avec *rtbs* et donnée *cs* indisponible
 - *Discount rate (dr)* (Taux d'actualisation): Attention, le taux est en décimal, pas en pourcentage. Pour 4% d'actualisation, il faut rentrer la valeur 0,04. Ce taux d'actualisation va s'appliquer en sortie du modèle à une sélection de variables économiques principales, qui seront donc disponibles en 2 versions, actualisée et non actualisée. Ces variables sont les suivantes : *rtbs_f*, *cs_f*, *gva_f*, *gcf_f*, *ps_f* et *sts_f*.

Une fois la validation des arguments effectuée en cliquant sur “OK,” un objet de classe `iamArgs` est retourné, intégrant tous les arguments qui vont conditionner les processus mis en œuvre pendant la phase de simulation. Afin de simplifier l'utilisation de cette classe d'objets, une méthode `summary()` a été implémentée qui permet d'avoir un aperçu rapide

```
summary(IAM_argum_2009)
#> My Input (IAM argument) :
#> Simulation of 3 dynamic species, 19 static species and 7 fleet
#> Simulation start in 2009 and end in 2020 (12 steps)
#>
#> =====
#> SR module | Stock Recruitment | Noise | Proba |
#> -----
#> Species | function : param A ; param B ; param C | Type : sd | Type |
#> ARC (XSA) | Mean 3.641e+07 0.00e+00 0.00e+00 | Norm | 0.00e+00 | . |
#> COR (XSA) | Mean 1.893e+07 0.00e+00 0.00e+00 | Norm | 0.00e+00 | . |
#> DAR (SS3) | not activated 0.000e+00 0.00e+00 0.00e+00 | Norm | 0.00e+00 | . |
#> -----
#>
#> The Gestion module is not active.
#>
#> =====
#> Economic : PerscCalc = 0 ; dr = 0.000 | No replicates |
#> -----
#>
#> The Scenario module is not active.
names(IAM_argum_2009@arguments)
#> [1] "Recruitment" "Replicates" "Scenario" "Gestion" "Eco"
```

On peut modifier l'objet déjà existant utilisant des fonctions spécifiques et les valeurs voulues. Mais ceci suggère une bonne connaissance de R et un risque non négligeable d'entamer la structure de l'objet. Par exemple, pour mettre en œuvre la module Gestion et modifier les variables économiques :

```
# Modify Gestion
IAM_argum_2009 <- IAM.editArgs_Gest(
  IAM_argum_2009, active = TRUE, control = "Nb trips", target = "Fbar",
  espece = "DAR", type = "x")
```

```
# Modify Eco
IAM_argum_2009 <- IAM.editArgs_Eco(IAM_argum_2009, dr = 0.04, perscCalc = 3)
# Check if Gestion is active
summary(IAM_argum_2009)
#> My Input (IAM argument) :
#> Simulation of 3 dynamic species, 19 static species and 7 fleet
#> Simulation start in 2009 and end in 2020 (12 steps)
#>
#> =====
#> SR module | Stock Recruitment | Noise | Proba |
#> -----
#> Species | function : param A ; param B ; param C | Type : sd | Type |
#> ARC (XSA) | Mean 3.641e+07 0.00e+00 0.00e+00 | Norm | 0.00e+00 | . |
#> COR (XSA) | Mean 1.893e+07 0.00e+00 0.00e+00 | Norm | 0.00e+00 | . |
#> DAR (SS3) | not activated 0.00e+00 0.00e+00 0.00e+00 | Norm | 0.00e+00 | . |
#> -----
#>
#> =====
#> Gestion Module |
#> -----
#> Species | control | target | delay | typeG | update | bounds |
#> DAR (SS3) | Nb trips | Fbar | 2 | x | yes | 0 : 0 |
#> -----
#>
#> TAC : NA NA NA NA NA NA NA NA NA NA NA NA
#> FBAR : NA NA NA NA NA NA NA NA NA NA NA NA
#> -----
#>
#> =====
#> Economic : PerscCalc = 3 ; dr = 0.040 | No replicates |
#> -----
#>
#> The Scenario module is not active.
# Deactivate Gestion
IAM_argum_2009 <- IAM.editArgs_Gest(IAM_argum_2009, active = FALSE)
```

Mais la procédure la plus simple pour corriger un objet est de reprendre l'interface au stade de la validation, et d'effectuer directement les corrections. Pour cela, il suffit de rappeler la méthode `IAM.args()`, mais non plus affectée à un objet de paramétrage initial de classe `iamInput`, mais plutôt à l'objet qu'on désire modifier. On tape donc :

```
IAM_argum_2009 <- IAM.args(IAM_argum_2009)
```

Une interface reprenant tous les arguments auparavant validés s'ouvre, permettant la correction, la validation, et ainsi la mise à jour de l'objet `IAM_argum_2009` en sortie.

Simulation (IAM.model)

Une fois les deux principaux composants du paramétrage obtenus, la mise en route de la simulation est très simple : la méthode `IAM.model()` appelle l'objet de classe `iamArgs` et l'objet de classe `iamInput` et restitue les résultats de la simulation dans un objet de classe `iamOutput`, ou de classe `iamOutputRep` si le module de réplication est activé.

```
IAM_out_2009 <- IAM.model(IAM_argum_2009, IAM_input_2009)
class(IAM_out_2009)
#> [1] "iamOutput"
```

```
#> attr("package")
#> [1] "IAM"
```

Toutes les sorties du modèle sont entreposées dans 2 slots de l'objet `IAM_out_2009`. Un slot `outputSp` réunit toutes les variables définies par espèce modélisée (sorties du modèle biologique principalement), et un autre slot `output` rassemble les variables sans dimension espèce (les variables économiques). L'accès aux variables est illustré ci-dessous :

```
slotNames(IAM_out_2009)
#> [1] "desc"      "arguments" "specific"   "outputSp"   "output"
names(IAM_out_2009@outputSp)
#> [1] "F"      "Fr"      "Fothi"    "Fbar"
#> [5] "Z"      "N"      "B"        "SSB"
#> [9] "C"      "Ctot"    "Y"        "Ytot"
#> [13] "D"      "Li"      "Lc"       "Ltot"
#> [17] "L_et"    "L_pt"    "P"        "GVL_f_m_e"
#> [21] "GVLcom_f_m_e" "GVLst_f_m_e" "statY"     "statL"
#> [25] "statD"    "statP"    "statGVL_f_m" "statGVLcom_f_m"
#> [29] "statGVLst_f_m" "PQuot"    "TradedQ"    "F_S1M1"
#> [33] "F_S1M2"    "F_S1M3"    "F_S1M4"    "F_S2M1"
#> [37] "F_S2M2"    "F_S2M3"    "F_S2M4"    "F_S3M1"
#> [41] "F_S3M2"    "F_S3M3"    "F_S3M4"    "F_S4M1"
#> [45] "F_S4M2"    "F_S4M3"    "F_S4M4"    "Fr_S1M1"
#> [49] "Fr_S1M2"    "Fr_S1M3"    "Fr_S1M4"    "Fr_S2M1"
#> [53] "Fr_S2M2"    "Fr_S2M3"    "Fr_S2M4"    "Fr_S3M1"
#> [57] "Fr_S3M2"    "Fr_S3M3"    "Fr_S3M4"    "Fr_S4M1"
#> [61] "Fr_S4M2"    "Fr_S4M3"    "Fr_S4M4"    "Z_S1M1"
#> [65] "Z_S1M2"    "Z_S1M3"    "Z_S1M4"    "Z_S2M1"
#> [69] "Z_S2M2"    "Z_S2M3"    "Z_S2M4"    "Z_S3M1"
#> [73] "Z_S3M2"    "Z_S3M3"    "Z_S3M4"    "Z_S4M1"
#> [77] "Z_S4M2"    "Z_S4M3"    "Z_S4M4"    "N_S1M1"
#> [81] "N_S1M2"    "N_S1M3"    "N_S1M4"    "N_S2M1"
#> [85] "N_S2M2"    "N_S2M3"    "N_S2M4"    "N_S3M1"
#> [89] "N_S3M2"    "N_S3M3"    "N_S3M4"    "N_S4M1"
#> [93] "N_S4M2"    "N_S4M3"    "N_S4M4"    "DD_efmi"
#> [97] "DD_efmc"    "LD_efmi"    "LD_efmc"    "statDD_efm"
#> [101] "statLD_efm" "statLDst_efm" "statLDor_efm" "oqD_ef"
#> [105] "oqD_e"      "oqDstat_ef" "TACTot"     "TACbyF"
#> [109] "PQuot_conv" "diffLQ_conv"
names(IAM_out_2009@output)
#> [1] "nbv_f"      "effort1_f"    "effort2_f"
#> [4] "nbv_f_m"    "effort1_f_m"  "effort2_f_m"
#> [7] "GVLtot_f_m" "GVLav_f_m"    "GVLtot_f"
#> [10] "GVLav_f"    "NGVLav_f_m"   "NGVLav_f"
#> [13] "ET_f_m"     "cnb_f_m"      "cnb_f"
#> [16] "rtbs_f_m"   "rtbs_f"       "rtbsAct_f"
#> [19] "cshrT_f_m"  "cshrT_f"      "ncshr_f"
#> [22] "ocl_f"      "cs_f"         "csAct_f"
#> [25] "csTot_f"    "gua_f"        "guaAct_f"
#> [28] "guamargin_f" "gua_FTE_f"    "ccw_f"
#> [31] "ccwCr_f"    "wageg_f"      "wagen_f"
#> [34] "wageg_FTE_f" "wageg_h_f"    "gp_f"
#> [37] "gpAct_f"    "gpmargin_f"   "ncf_f"
#> [40] "np_f"       "npmargin_f"   "prof_f"
```

```
#> [43] "npmargin_trend_f" "ssTot_f" "ps_f"
#> [46] "psAct_f" "sts_f" "stsAct_f"
#> [49] "BER_f" "CR_BER_f" "fuelEff_f"
#> [52] "ratio_fvol_gva_f" "ratio_gp_gva_f" "ratio_GVL_K_f"
#> [55] "ratio_gp_K_f" "RoFTA_f" "ROI_f"
#> [58] "ratio_np_K_f" "ratio_GVL_cnb_ue_f" "YTOT_fm"
#> [61] "reconcilSPP" "quotaExp_f" "allocEff_f_m"
#> [64] "GoFish"
```

Dans le cas d'un objet de classe `iamOutputRep`, chaque variable est une liste de `n` éléments, avec `n` le nombre de réplicats.

```
# Activate replicates
IAM_argumRep_2009 <- IAM.editArgs_Rep(IAM_argum_2009, n = 2)

IAM_outRep_2009 <- IAM.model(IAM_argumRep_2009, IAM_input_2009)
class(IAM_outRep_2009)
#> [1] "iamOutputRep"
#> attr(,"package")
#> [1] "IAM"
length(IAM_outRep_2009@outputSp$F)
#> [1] 2
names(IAM_outRep_2009@outputSp$F[[1]])
#> [1] "ARC" "COR" "DAR"
names(IAM_outRep_2009@outputSp$F[[2]])
#> [1] "ARC" "COR" "DAR"
```

Mise en forme des sorties (IAM.format & IAM.format_quant)

La méthode `IAM.format` est une fonction de formatage appliquée aux objets `iamOutput`. Elle va permettre la conversion d'une variable multidimensionnelle en une table de type "data.frame" (format facilitant la création de graphiques adaptés à ce type de variables)

Cette fonction prend également en entrée le nom d'une ou plusieurs variables à extraire, un nom de simulation et un numéro qui pourront servir plus tard pour agréger plusieurs simulations ensemble.

On note qu'une variable spéciale nommée "summary" permet d'extraire d'un seul coup quelques variables biologiques et économiques.

```
# Extraction d'une variable
statu_quo_tbl_Fbar <- IAM.format(IAM_out_2009, name = "Fbar",
                                sim_name = "statu_quo", n = 1)

head(statu_quo_tbl_Fbar)
#> # A tibble: 6 x 9
#>   sim_name      n variable species fleet metier age   year value
#>   <chr>      <dbl> <chr>   <chr>   <chr> <chr> <chr> <dbl> <dbl>
#> 1 statu_quo      1 Fbar    ARC     <NA> <NA> <NA> 2009 0.122
#> 2 statu_quo      1 Fbar    ARC     <NA> <NA> <NA> 2010 0.122
#> 3 statu_quo      1 Fbar    ARC     <NA> <NA> <NA> 2011 0.122
#> 4 statu_quo      1 Fbar    ARC     <NA> <NA> <NA> 2012 0.122
#> 5 statu_quo      1 Fbar    ARC     <NA> <NA> <NA> 2013 0.122
#> 6 statu_quo      1 Fbar    ARC     <NA> <NA> <NA> 2014 0.122

# Extraction de plusieurs variables
statu_quo_tbl_summary <- IAM.format(IAM_out_2009, name = "summary",
                                    sim_name = "statu_quo", n = 1)
unique(statu_quo_tbl_summary$variable)
```

```
#> [1] "Fbar"      "SSB"      "L_et"      "N"         "nbv_f"     "effort1_f"
#> [7] "effort2_f" "GVLav_f"  "gva_f"     "gp_f"      "wageg_f"   "wagen_f"
```

Ce format de table long permet ensuite l'utilisation d'outils pour trier et éventuellement produire les tables dans un format large.

```
library(dplyr)
library(tidyr)
#> Warning: le package 'tidyr' a été compilé avec la version R 4.1.2
#>
#> Attachement du package : 'tidyr'
#> L'objet suivant est masqué depuis 'package:magrittr':
#>
#> extract
library(purrr)
#>
#> Attachement du package : 'purrr'
#> L'objet suivant est masqué depuis 'package:magrittr':
#>
#> set_names
library(magrittr)

# Extraction du SSB
statu_quo_tbl_summary %>%
  dplyr::filter( species == "COR", variable == "SSB") %>% # trie des lignes
  discard(~all(is.na(.x))) %>% # retire les colonnes vides (NA)
  # Ligne optionnelle pour enlever les colonnes identiques.
  # discard( ~ n_distinct(.) == 1) %>%
  pivot_wider(names_from = year, values_from = value) # mise en forme horizontale
#> # A tibble: 1 x 16
#>   sim_name      n variable species `2009` `2010` `2011` `2012` `2013` `2014`
#>   <chr>      <dbl> <chr>    <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#> 1 statu_quo      1 SSB      COR      10426. 11347. 12051. 12523. 12820. 13033.
#> # ... with 6 more variables: `2015` <dbl>, `2016` <dbl>, `2017` <dbl>,
#> # `2018` <dbl>, `2019` <dbl>, `2020` <dbl>
# Table de gp_f
statu_quo_tbl_summary %>%
  dplyr::filter(variable == "gp_f") %>% # trie des lignes
  discard(~all(is.na(.x))) %>% # retire les colonnes vides (NA)
  # Ligne optionnelle pour enlever les colonnes identiques.
  discard( ~ n_distinct(.) == 1) %>%
  pivot_wider(names_from = year, values_from = value) # mise en forme horizontale
#> # A tibble: 7 x 13
#>   fleet `2009` `2010` `2011` `2012` `2013` `2014` `2015` `2016` `2017`
#>   <chr>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#> 1 Alis   -3.10e7 -3.10e7 -3.10e7 -3.10e7 -3.10e7 -3.10e7 -3.10e7 -3.10e7 -3.10e7
#> 2 Antea  -4.75e7 -4.75e7 -4.75e7 -4.75e7 -4.75e7 -4.75e7 -4.75e7 -4.75e7 -4.75e7
#> 3 Atala~ -1.18e7 -1.18e7 -1.18e7 -1.18e7 -1.18e7 -1.18e7 -1.18e7 -1.18e7 -1.18e7
#> 4 Halio~ -6.83e5 -6.86e5 -6.92e5 -6.92e5 -6.87e5 -6.72e5 -6.54e5 -6.38e5 -6.24e5
#> 5 Mario~ -6.24e6 -6.26e6 -6.25e6 -6.23e6 -6.23e6 -6.24e6 -6.24e6 -6.25e6 -6.25e6
#> 6 Pourq~ -1.66e7 -1.66e7 -1.66e7 -1.66e7 -1.66e7 -1.66e7 -1.66e7 -1.66e7 -1.66e7
#> 7 Thala~ -3.21e7 -3.21e7 -3.21e7 -3.21e7 -3.21e7 -3.21e7 -3.21e7 -3.21e7 -3.20e7
#> # ... with 3 more variables: `2018` <dbl>, `2019` <dbl>, `2020` <dbl>
```


Dans le cas d'une liste d'objet de classe `iamOutput`, il est possible d'extraire les variables, de concaténer et d'ensuite calculer une table résumée ne présentant que les quantiles d'intérêt. Cela permet d'alléger les sorties du modèle et de les manipuler plus simplement lors de la production de représentations graphiques.

```
res1 <- IAM.format(IAM_out_2009, c("SSB"), n = 1) %>%
  dplyr::filter(species == "ARC", year <= 2011)
# Je modifie légèrement les données pour avoir des quantiles écartés.
res2 <- mutate(res1, n = 2, value = value + rnorm(1, sd = 100))

# On peut associer les données extraites avec rbind.
res <- rbind(res1, res2)

IAM.format_quant(res, probs = c(0.025, 0.975), select_indiv = 2) %>%
  discard(~all(is.na(.x)))
#> # A tibble: 3 x 8
#>   variable species year quant1 quant2 median value nsim
#>   <chr>      <chr>   <dbl> <dbl> <dbl> <dbl> <dbl> <int>
#> 1 SSB      ARC      2009 18728. 18921. 18824. 18926.     2
#> 2 SSB      ARC      2010 17995. 18188. 18092. 18193.     2
#> 3 SSB      ARC      2011 17225. 17418. 17321. 17423.     2
```

A noter que des colonnes seront souvent remplies de valeurs manquantes car peu de variables sont définies sur toutes les dimensions d'IAM. Cependant, il vaut mieux conserver ces colonnes jusqu'au dernier moment afin de pouvoir trier les données avec des fonctions génériques ou encore pouvoir utiliser `rbind`.

Représentations graphiques des sorties avec ggplot2

Les procédures graphiques ont été retirées du packages et se basent sur l'utilisation du package `ggplot2`. Ainsi, le format de table renvoyé par les fonctions `IAM.format` et `IAM.format_quant` sera requis pour constituer la donnée à représenter.

Pour illustrer différentes représentations graphiques, on va générer plusieurs réplicats. Le premier sera similaire le cas de *statu quo* généré plus haut. On modifie le module de recrutement pour l'espèce "COR" afin d'ajouter un peu d'effets stochastiques. Un second scénario est simulé avec l'application de quotas. Le détails de ce genre de simulations est donné dans la vignette "Gestion_TAC."

```
N = 5
IAM_argum_2009@arguments$Recruitment$COR$wnNOISEmodSR <- 5e6
# Simulation
sim_statu_quo <- replicate(N, {
  IAM::IAM.model(objArgs = IAM_argum_2009, objInput = IAM_input_2009)
})
# Edition des arguments
IAM_argum_2009_TACnbv <- IAM.editArgs_Gest(
  IAM_argum_2009, active = TRUE, control = "Nb vessels", target = "TAC",
  espece = "COR", delay = 2, type = "x", bounds = c(1e7, -1),
  tac = c(NA, NA, rep(3400, 10)))
# Simulation
sim_TACglob_nbv <- replicate(N, {
  IAM::IAM.model(objArgs = IAM_argum_2009_TACnbv, objInput = IAM_input_2009)
})
# Formatage
TACglob_nbv_l <- lapply(1:N, function(x) {
  IAM.format(sim_TACglob_nbv[[x]], name = "summary",
    sim_name = "TAC global nbv", n = x)
```



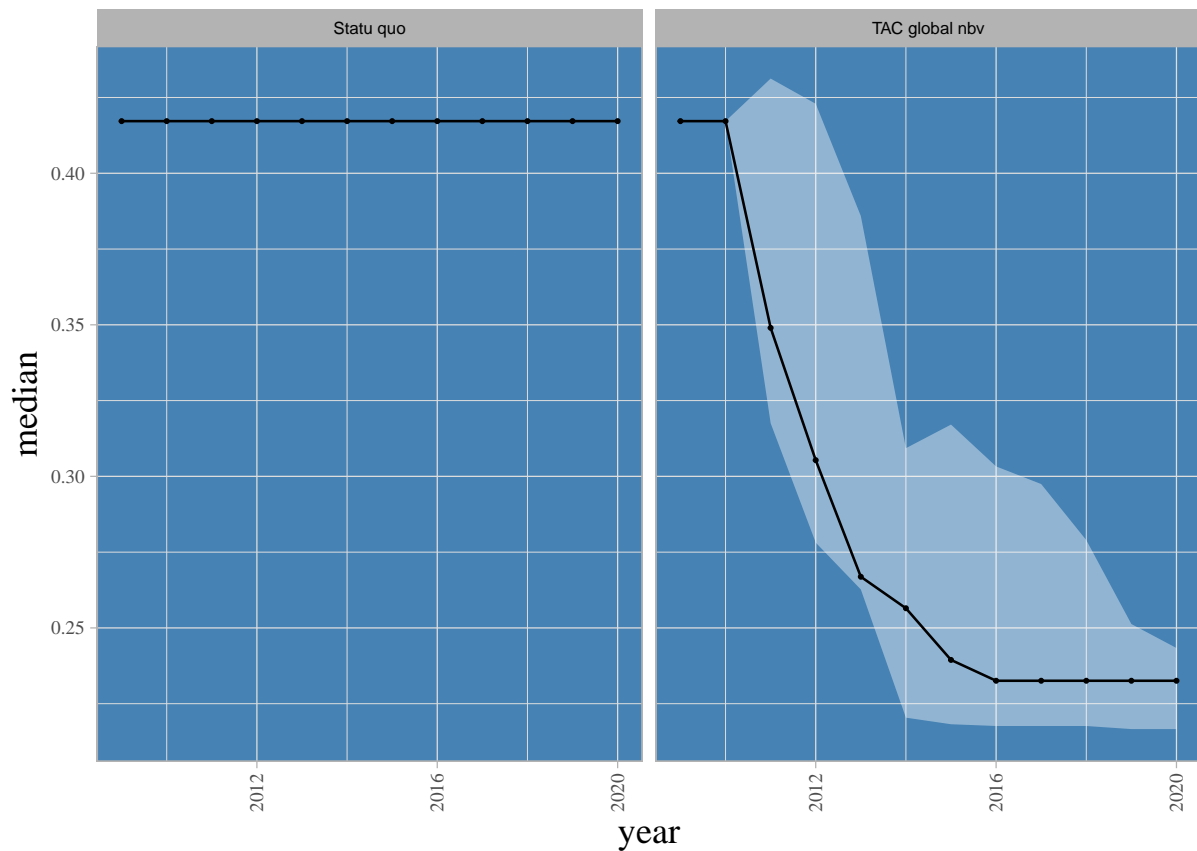
```
}  
Statuquo_l <- lapply(1:N, function(x) {  
  IAM.format(sim_statu_quo[[x]], name = "summary",  
            sim_name = "Statu quo", n = x)  
})  
  
TACvsSQ_5simuls <- do.call(rbind, c(Statuquo_l, TACglobv_nbv_l)) %>%  
  IAM.format_quant(., probs = c(.025, .975))
```

Pour les sorties graphiques on aura besoin du package ggplot2. Cette dépendance est normalement installée en même temps que le package IAM. Ce package est très riche en terme de représentations graphiques, on pourra se référer à sa documentation pour plus de précisions. Un site repertorie également de nombreux exemples : <https://r-graph-gallery.com/>.

```
library(ggplot2)  
library(magrittr) # utilisation du pipe %>%  
library(dplyr) # utilisation de filtres
```

Voici quelques exemples de représentations graphiques disponibles pour les sortie d'IAM :

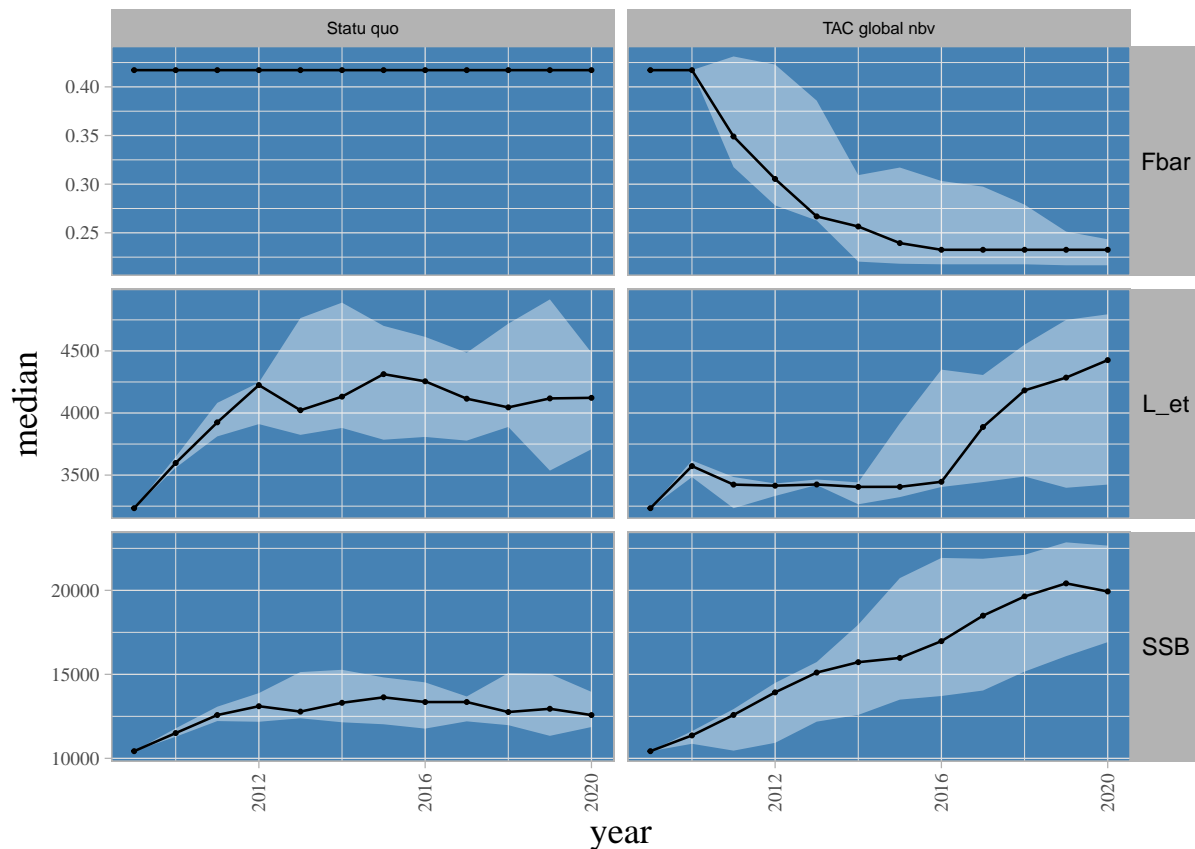
```
TACvsSQ_5simuls %>%  
  # Filtrer de l'espece et de la variable cible  
  dplyr::filter(species == "COR", variable == "Fbar") %>%  
  ggplot(aes(x = year, y = median)) +  
  facet_grid(. ~ sim_name, scales = "free_y") +  
  geom_ribbon(aes(ymin = quant1, ymax = quant2), fill = "white", alpha = .4) +  
  geom_line() + geom_point(size = .5) +  
  guides(x = guide_axis(angle = 90)) + IAM_theme() +  
  NULL
```



L'exemple précédent permet d'illustrer la comparaison du F_{bar} pour deux scénarios et une espèce donnée ("COR"). On peut observer l'enveloppe à droite qui est représentée les deux quartiles entre lesquels on retrouve 95% des simulations.

On peut également représenter plusieurs variables comme ci-dessous les variables biologiques de l'espèce COR :

```
TACvsSQ_5simuls %>%
  dplyr::filter(species == "COR", variable %in% c("Fbar", "SSB", "L_et")) %>%
  ggplot(aes(x = year, y = median)) +
  facet_grid(variable ~ sim_name, scales = "free_y") +
  geom_ribbon(aes(ymin = quant1, ymax = quant2), fill = "white", alpha = .4) +
  geom_line() + geom_point(size = .5) +
  guides(x = guide_axis(angle = 90)) + IAM_theme() +
  NULL
```



Enfin, on peut tout à fait intégrer plus de dimensions aux graphiques avec l'ajout de couleurs pour par exemple ici comparer deux scénarios par Flottes et variables économiques :

```
TACvsSQ_5simuls %>%
  dplyr::filter(variable %in% c("GVLav_f", "nbv_f")) %>%
  ggplot(aes(x = year, y = median, color = sim_name)) +
  facet_grid(variable ~ fleet, scales = "free_y") +
  geom_ribbon(aes(ymin = quant1, ymax = quant2), fill = "white", alpha = .4) +
  geom_line() + geom_point(size = .5) +
  guides(x = guide_axis(angle = 90)) + IAM_theme() +
  scale_color_manual(values=c("darkred", "darkorange")) +
  theme(legend.position = "bottom") +
  NULL
```

Conseil de réplication

L'utilisation du package IAM prend tout son sens lors du lancement d'un même scénario répliqué de nombreuses fois. Cela permet de prendre en compte les problématiques stochastiques de certaines variables comme le recrutement. Ainsi, un même scénario peut être lancé plusieurs centaines de fois. Cette option est implémentée de base dans le package avec la classe `IAMOutputRep`, mais celle-ci est dépréciée car ne permettant plus la réalisation de scénarios complexes.

Une autre alternative réside dans l'utilisation de code R pour forcer de grands nombres de réplicats, au détriment d'une plus grande place en mémoire. Notons tout de même que cela peut prendre un certain temps.

```
# Code non exécuté
library(IAM)
```

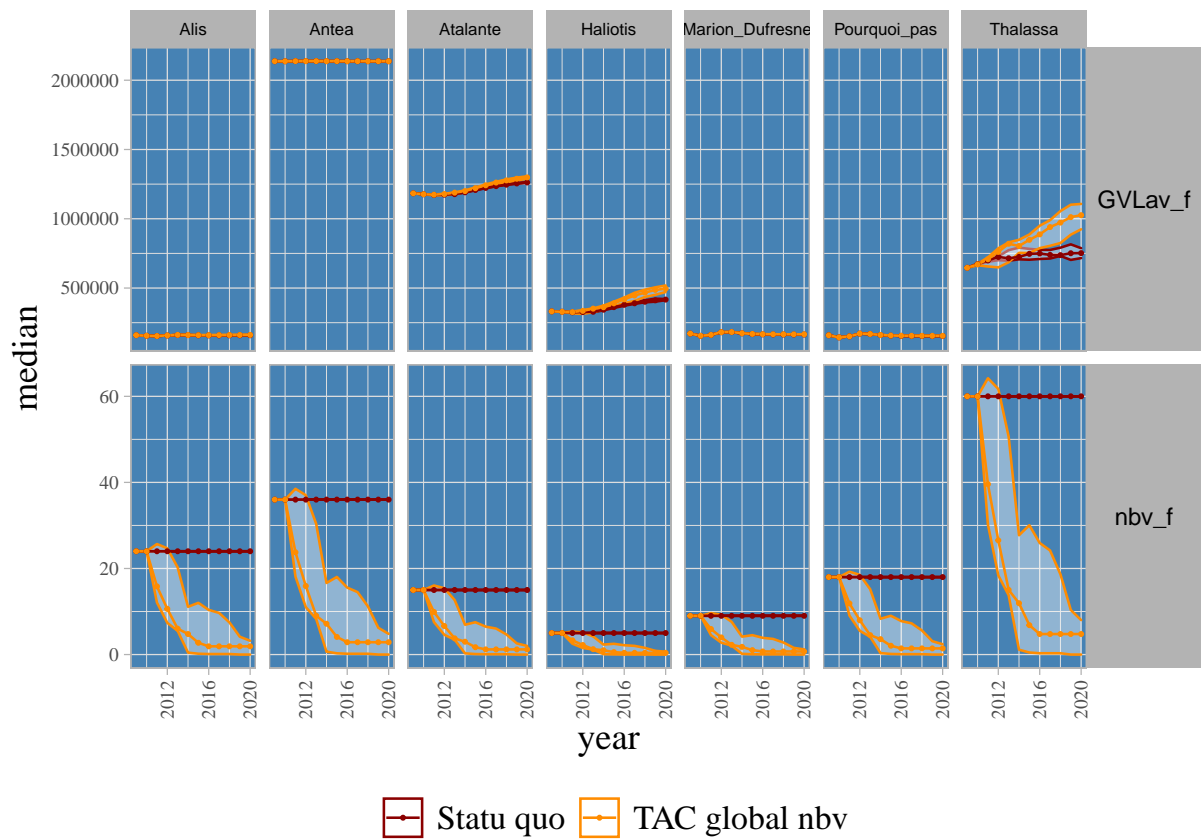


Figure 1: Evolution des variables économiques pour les flottilles modélisées. 5 runs.

```
a <- Sys.time() # mesure du temps
nsim_statu_quo <- replicate(
  500,
  IAM::IAM.model(objArgs = IAM_argum_2009, objInput = IAM_input_2009)
)
b <- Sys.time() # mesure du temps
b - a # 1.17 min
```

Afin de gagner du temps, le package `{parallel}` permet d'effectuer les calculs sur un plus grand nombre de processeurs de l'ordinateur. L'utilisation de ce package permet de réduire le temps de calcul pour un grand nombre de réplicats et est donc conseillé.

```
# Code non exécuté
library(parallel)
cl <- makeCluster(detectCores()-1)

clusterEvalQ(cl, library(IAM)) # charger les librairies dans le cluster.

c <- Sys.time() # mesure du temps
# rendre les objets disponibles pour le code
clusterExport(cl, c("IAM_argum_2009", "IAM_input_2009"))
# S'assurer de "seed" différentes pour chaque noeud.
clusterSetRNGStream(cl)
#... Simuler les réplicats ...
res <- parSapply(cl, 1:50, function(i,...) {
  IAM::IAM.model(objArgs = IAM_argum_2009, objInput = IAM_input_2009)
})
# Arrêter le cluster
stopCluster(cl)
d <- Sys.time() # mesure du temps
d - c # 51 sec !!!
```

References

R Core Team. 2021. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. <https://www.R-project.org/>.

Gestion avec TAC (fr)

```
# Dépendances pour l'analyse des données et les représentations graphiques.
library(magrittr)
library(ggplot2)
library(dplyr)
library(IAM)
library(pbapply) # allow progress bar in replicate and apply.
```

Le modèle bio-économique IAM a été développé notamment pour permettre d'accompagner le développement des plans de gestion des pêcheries et d'explorer les conséquences biologiques et socio-économiques de différents scénarios de TAC et quotas et de transition vers le Rendement Maximum Durable - RMD (ou Maximum Sustainable Yield - MSY).

Ce document présente la mise en œuvre avec le modèle IAM de scénarios de TAC et quotas correspondant à différentes manières de moduler l'effort de pêche. Ce document traite de la gestion de TAC à un niveau interflottes puis une autre partie détaillera la gestion intra flottes.

L'ensemble des simulations présentées dans l'exemple sera réalisé avec le jeu de donnée exemple **Ifremer** composé de 7 flottes et 3 espèces dynamiques dont une espèce à dynamique SS3. Les simulations prennent en compte la variabilité du recrutement et ses conséquences en termes de probabilité d'atteinte du FMSY. Dans ce cadre, une réplication de valeur N sera réalisée à partir du jeu de donnée exemple.

Attention ce document lance de nombreuses simulations $(1 + n_{TAC}) \cdot N$ et prend donc un temps long à simuler (plusieurs dizaines de minutes).

```
data("IAM_input_2009")
summary(IAM_input_2009)
#> My Input (IAM input) :
#> Simulation of 3 dynamic species, 19 static species and 7 fleet
#> Simulation start in 2009 and end in 2020 (12 steps)
#>
#> -----
#> Dynamic Species | Model |      Ages |
#>               ARC |  XSA | 0 to +gp |
#>               COR |  XSA | 2 to +gp |
#>               DAR |  SS3 | 0 to +gp |
#> -----
#>               Fleet |      nbv   |
#>               Alis |      24    |
#>               Antea |      36    |
#>               Atalante |      15    |
#>               Haliotis |      5     |
#>               Marion_Dufresne |      9     |
#>               Pourquoi_pas |      18    |
#>               Thalassa |      60    |
```

L'objet argument est laissé tel quel lors de l'utilisation de l'interface et sera édité à la main plus tard pour chaque scénario.

```
# Cette ligne ouvre une interface via une app shiny.
```

```
IAM_argum_2009 <- IAM.args(IAM_input_2009)
```

Cela revient à initialiser un objet de classe `iamArgs` sans passer par l'interface avec la commande suivante :

```
IAM_argum_2009 <- IAM.input2args(IAM_input_2009)
```

Scénario *statu quo*

Afin de pouvoir comparer l'effet de chaque scénario, il nous faut un scénario de départ dans lequel aucune mesure de Gestion s'imposera.

Dans un premier temps il est utile de définir les dynamiques de recrutements ainsi que des éléments de paramétrage du module économique. On va pour cela éditer l'objet `IAM_argum_2009`.

```
# Module SR
# Add noise to COR recruitment
IAM_argum_2009@arguments$Recruitment$COR$wnNOISEmodSR <- 0.203
IAM_argum_2009@arguments$Recruitment$COR$noiseTypeSR <- 2 # Log-normal
IAM_argum_2009@arguments$Recruitment$COR$typeMODsr <- "Hockey-Stick"
IAM_argum_2009@arguments$Recruitment$COR$parAmodSR <- 2539.5
IAM_argum_2009@arguments$Recruitment$COR$parBmodSR <- 9679

# Module EcoDCF
IAM_argum_2009 <- IAM.editArgs_Eco(IAM_argum_2009, dr = 0.04, perscCalc = 1)

# Module Gestion
mfm <- with(IAM_input_2009@input$Fleet,{
  (effort1_f_m * effort2_f_m * nbv_f_m) / as.vector(effort1_f * effort2_f * nbv_f)
})
mfm[is.na(mfm)] <- 0

IAM_argum_2009 <- IAM.editArgs_Gest(IAM_argum_2009, active = FALSE,
  delay = 1, mfm = mfm)

# Module Scenario
IAM_argum_2009 <- IAM.editArgs_Scenar(IAM_argum_2009) # deactivate scenario

summary(IAM_argum_2009)
#> My Input (IAM argument) :
#> Simulation of 3 dynamic species, 19 static species and 7 fleet
#> Simulation start in 2009 and end in 2020 (12 steps)
#>
#> =====
#> SR module | Stock Recruitment | Noise | Proba |
#> -----
#> Species | function : param A ; param B ; param C | Type : sd | Type |
#> ARC (XSA) | Mean 3.641e+07 0.00e+00 0.00e+00 | Norm | 0.00e+00 | . |
#> COR (XSA) | Hockey-Stick 2.540e+03 9.68e+03 0.00e+00 | LogN | 2.03e-01 | . |
#> DAR (SS3) | not activated 0.000e+00 0.00e+00 0.00e+00 | Norm | 0.00e+00 | . |
#> -----
#>
#> The Gestion module is not active.
#>
#> =====
```

```
#> Economic : PerscCalc = 1 ; dr = 0.040 / No replicates /
#> -----
#>
#> The Scenario module is not active.
```

Les scénarios sont simulés avec N réplicats, correspondant à la variabilité du recrutement.

Les valeurs de N , F_{msy} pour l'espèce COR et TAC sont fixées ici :

```
N <- 5
Fmsy <- 0.26
TAC <- 3600
```

La simulation du scénario Statu Quo est codée comme suit :

```
# Statu quo
SQ <- replicate(N, {
  IAM::IAM.model(objArgs = IAM_argum_2009, objInput = IAM_input_2009)
})
```

Gestion TAC générale

Les différentes options de réglages du module Gestion sont exposées dans la vignette “Utilisation IAM dans R”. Il s’agit ici de les explorer plus en avant et de les utiliser.

On fixe ici un scénario de Gestion qui va agir sur les efforts (nombre de navires ou nombre de voyages) afin d’atteindre un TAC (*Total allowable catches*) donné pour chaque année. Ce quotas vise particulièrement l’espèce dite “COR” et rentre en application 2 ans après le début de la simulation, soit en 2011. Les modification d’efforts seront multiplicatives.

Concernant la recherche de réduction d’effort, une recherche d’optimum se fait par itération afin de trouver le multiplicateur μ tel que la différence entre le TAC requis et atteints est minimale. Le maximum d’itération est de 10, et peut s’arrêter avant si la différence est inférieur à une tolérance donnée.

La recherche de ce μ se fait dans un intervalle donné dans les paramètres (nommé bounds). Cet interval dépend fortement du type de Gestion utilisé. En effet, on essaye d’atteindre un effort var_c à partir d’un effort initial var_i par l’application de μ sur la matrice $m_{f,m}$ comme illustré dans les équations suivantes.

Cas	Place de μ dans la calcul du TAC visé
Additif	$var_c = var_i + \mu \cdot m_{f,m}$
Multiplicatif	$var_c = var_i + var_i \cdot \mu \cdot m_{f,m}$

Cependant, le modèle va venir s’assurer que var_c ne devient pas négatif et assigner la valeur 0 dans ce cas précis. Afin de limiter les temps de calculs, il est donc préférable de chercher μ dans les intervals suivants :

Cas	Valeur de μ possibles
Additif	$\mu \in [\frac{-1}{m_{f,m}}; \infty]$
Multiplicatif	$\mu \in [\frac{-var_i}{m_{f,m}}; \infty]$

Il est également important de noter qu’un interval $[0;0]$ revient à ne pas appliquer le module Gestion.

```
IAM_argum_2009_TACnbv <- IAM.editArgs_Gest(
  IAM_argum_2009, active = TRUE, control = "Nb vessels", target = "TAC",
  espece = "COR", delay = 2,
```



```

type = "x", bounds = c(1e3, -100),
tac = c(NA, NA, rep(TAC, 10)))
# 84, 85, 2011:1995

sim_TACglob_nbv <- pbreplicate(N, {
  IAM::IAM.model(objArgs = IAM_argum_2009_TACnbv,
    objInput = IAM_input_2009)
})

```

Cette simulation permet de diminuer drastiquement la mortalité par pêche de l'espèce dite "COR" avec une réduction immédiate la taille des flottilles. Il s'agit de la capacité de pêche.

Une autre variable d'ajustement de l'effort est le nombre voyage en mer des navires. On parle d'effort de pêche.

```

IAM_argum_2009_TACnbds <- IAM.editArgs_Gest(
  IAM_argum_2009_TACnbv, active = TRUE, control = "Nb trips"
)

sim_TACglob_nbt <- pbreplicate(N, {
  IAM::IAM.model(objArgs = IAM_argum_2009_TACnbds,
    objInput = IAM_input_2009)
})

```

Représentations graphiques

On peut aisément comparer les différents scénarios avec des fonctions graphiques. Pour cela il faut tout d'abord simplifier les sorties sous format de tableau. Cela est réalisable avec la fonction `IAM.format()`. La fonction `IAM.format_quant` va quand à elle permettre de ne récupérer que les quartiles et la médiane de nos N simulations.

```

TACglob_nbv <- lapply(1:N, function(x) {
  IAM.format(sim_TACglob_nbv[[x]], name = "summary",
    sim_name = "TAC capacité de pêche", n = x)
})

TACglob_nbt <- lapply(1:N, function(x) {
  IAM.format(sim_TACglob_nbt[[x]], name = "summary",
    sim_name = "TAC effort de pêche", n = x)
})

Statuquo <- lapply(1:N, function(x) {
  IAM.format(SQ[[x]], name = "summary",
    sim_name = "Statu quo", n = x)
})

TACglob_quant <- do.call(rbind, c(Statuquo, TACglob_nbt, TACglob_nbv)) %>%
  IAM.format_quant(., probs = c(.025, .975))

```

Une fois les données formatées, on peut aisément comparer les différents scénarios de TACS. Ainsi, la mise en place d'un TAC va venir diminuer la mortalité par pêche (F_{bar}) sur le stock visé.

```

TACglob_quant %>%
  filter(species == "COR", variable %in% c("Fbar", "SSB", "L_et")) %>%

```

```
ggplot(aes(x = year, y = median)) +
  facet_grid(variable ~ sim_name, scales = "free_y") +
  geom_ribbon(aes(ymin = quant1, ymax = quant2), fill = "white", alpha = .4) +
  geom_line() + geom_point(size = .5) +
  geom_line(aes(y = value), linetype = "dotted") +
  geom_vline(xintercept=2011, linetype = "dotted") +
  guides(x = guide_axis(angle = 90)) + IAM_theme() +
  NULL
```

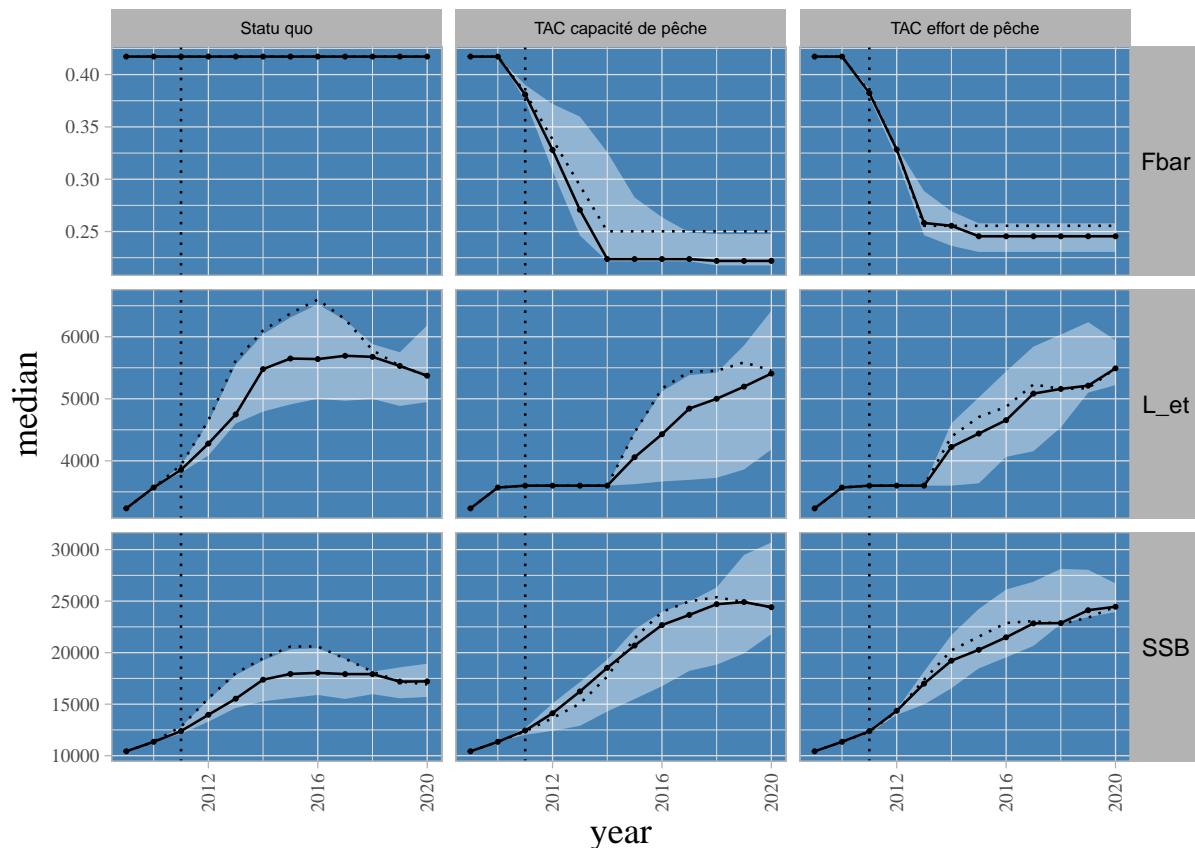


Figure 1: COR variables under global TAC gestion. 5 runs.

Cependant, ces scénarios entraînent de fortes réductions d'effort car ils ne ciblent pas les métiers et flottilles impactant le plus le stock cible. Ainsi on observe que les variables ajustées lors des scénarios arrivent vite proche de valeurs nulles.

Pour rappel effort1 et effort2 correspondent respectivement à nbTrip et tripLgth.

```
TACglob_quant %>%
  filter(variable %in% c("nbv_f", "effort1_f")) %>%
  ggplot(aes(x = year, y = value, color = fleet )) +
  facet_grid(variable ~ sim_name, scales = "free_y") +
  geom_line() +
  geom_vline(xintercept=2011, linetype = "dotted") +
  theme(legend.position = "bottom") +
  NULL
```

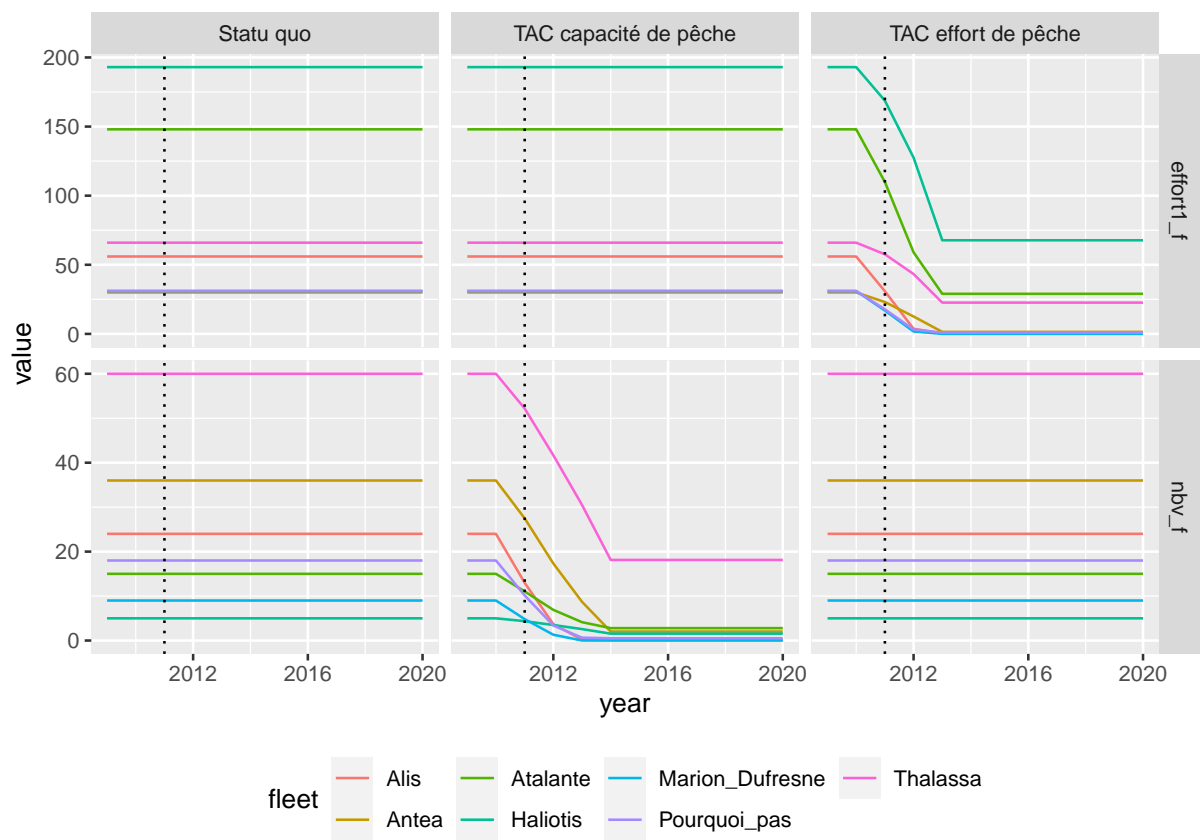


Figure 2: Effort variables under global TAC gestion. 5 runs.

Pondérer les efforts avec la matrice mfm

Ces réductions drastiques peuvent être limitées en ciblant les flottilles et les métiers. Par exemple, l'espèce ciblée est pêchée à 97.0259956 % par la flottille dite "Thalassa", via le métier "Filet_COR". Il serait donc plus intéressant de diminuer l'effort sur cette flottille uniquement.

Pour cela on peut agir sur la pondération des métiers et indiquer que l'on diminue l'effort des métiers "Filet_COR" pour les flottilles qui l'utilisent au profit du métier "Filet_DP". Il s'agit d'un **report d'effort** par métier.

```
trg_f <- c("Thalassa", "Haliotis")

mfm <- with(IAM_input_2009@input$Fleet,{
  (effort1_f_m * effort2_f_m * nbv_f_m) /
  as.vector(effort1_f * effort2_f * nbv_f)
})
mfm[!is.na(mfm)] <- 0 #report d'effort du m?tier 'Filet_COR' vers 'Filet_DP'
mfm[trg_f,"Filet_DP"] <- with(IAM_input_2009@input$Fleet,
  Lref_f_m[trg_f,"Filet_DP"] / sum(Lref_f_m[trg_f,"Filet_DP"])
)
mfm[trg_f,"Filet_COR"] <- with(IAM_input_2009@input$Fleet,
  - Lref_f_m[trg_f,"Filet_DP"] / sum(Lref_f_m[trg_f,"Filet_DP"])
)
print(mfm)
#>           Filet_DAR  Filet_DP Chalut_fond_DP Chalut_fond_GLY
#> Alis                0 0.00000000          NA          NA
#> Antea              NA          NA            0            0
#> Atalante           NA          NA            0            0
#> Haliotis           NA 0.05556576          NA          NA
#> Marion_Dufresne    NA          NA          NA          NA
#> Pourquoi_pas       NA          NA          NA          NA
#> Thalassa           0 0.94443424          NA          NA
#>           Chalut_fond_CEP Chalut_pelagique_DP Chalut_fond_COR Autres
#> Alis                NA          NA          NA          NA
#> Antea              NA          NA          NA          NA
#> Atalante            0            0            0            0
#> Haliotis           NA          NA          NA            0
#> Marion_Dufresne    NA          NA          NA          NA
#> Pourquoi_pas       NA          NA          NA          NA
#> Thalassa           NA          NA          NA            0
#>           Filet_COR Palangre_DP Palangre_DAR
#> Alis                NA          NA          NA
#> Antea              NA          NA          NA
#> Atalante           NA          NA          NA
#> Haliotis          -0.05556576            0          NA
#> Marion_Dufresne    NA          NA            0
#> Pourquoi_pas       NA            0            0
#> Thalassa          -0.94443424          NA          NA
#> attr(,"DimCst")
#> [1]  7 11  0  0
IAM_argum_2009_TACnbvR <- IAM.editArgs_Gest(IAM_argum_2009_TACnbv,
  mfm = mfm, control = "Nb trips")

sim_TACglob_nbvR <- pbreplicate(N, {
  sim_TACreport <- IAM::IAM.model(objArgs = IAM_argum_2009_TACnbvR,
```

```
objInput = IAM_input_2009)
})
```

Il est tout à fait possible de rajouter aux précédents scénarios déjà simulés pour avoir un moyen de comparaison. Il faut cependant éviter de rajouter les objets à la suite avec `rbind` comme cela, car R utilise beaucoup plus de mémoire.

```
# Extraction des variables.
TACglob_nbvR <- lapply(1:N, function(x) {
  IAM.format(sim_TACglob_nbvR[[x]], name = c("summary", "GP"),
    sim_name = "Report TAC global nbds", n = x)
})
# Calcul des quantiles et ajout aux simulations précédentes.
TACglob_quantR <- do.call(rbind, c(TACglob_nbvR)) %>%
  IAM.format_quant(., probs = c(.025, .975)) %>%
  rbind(., TACglob_quant) %>%
  mutate(sim_name = factor(sim_name, levels = c( # Reorder the column in plot
    "Statu quo", "TAC global nbtrip",          # by setting level order
    "TAC global nbv", "Report TAC global nbds"))))
```

La comparaison montre une dimension du Fbar sur l'espèce cible COR.

```
TACglob_quantR %>%
  filter(species == "COR", variable %in% c("Fbar", "SSB", "L_et")) %>%
  ggplot(aes(x = year, y = median)) +
  facet_grid(variable ~ sim_name, scales = "free_y") +
  geom_ribbon(aes(ymin = quant1, ymax = quant2), fill = "white", alpha = .4) +
  geom_line() + geom_point(size = .5) +
  geom_line(aes(y = value), linetype = "dotted") +
  geom_vline(xintercept=2011, linetype = "dotted") +
  guides(x = guide_axis(angle = 90)) + IAM_theme() +
  NULL
```

Sur le graphe des efforts par flottilles, il est facilement notable que les efforts ne sont pas aussi drastiquement impactés.

```
TACglob_quantR %>%
  filter(variable %in% c("nbv_f", "effort1_f")) %>%
  ggplot(aes(x = year, y = value, color = fleet)) +
  facet_grid(variable ~ sim_name, scales = "free_y") +
  geom_line() +
  geom_vline(xintercept=2011, linetype = "dotted") +
  NULL
```

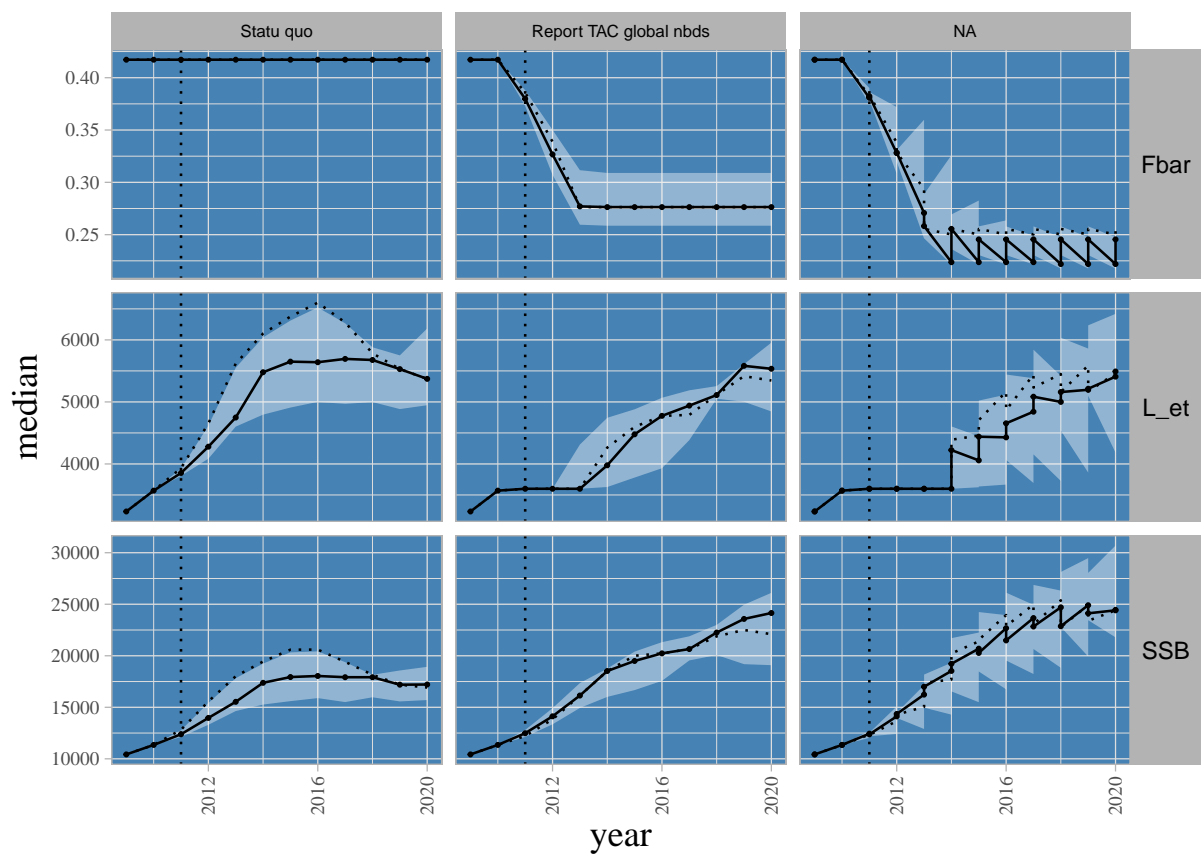
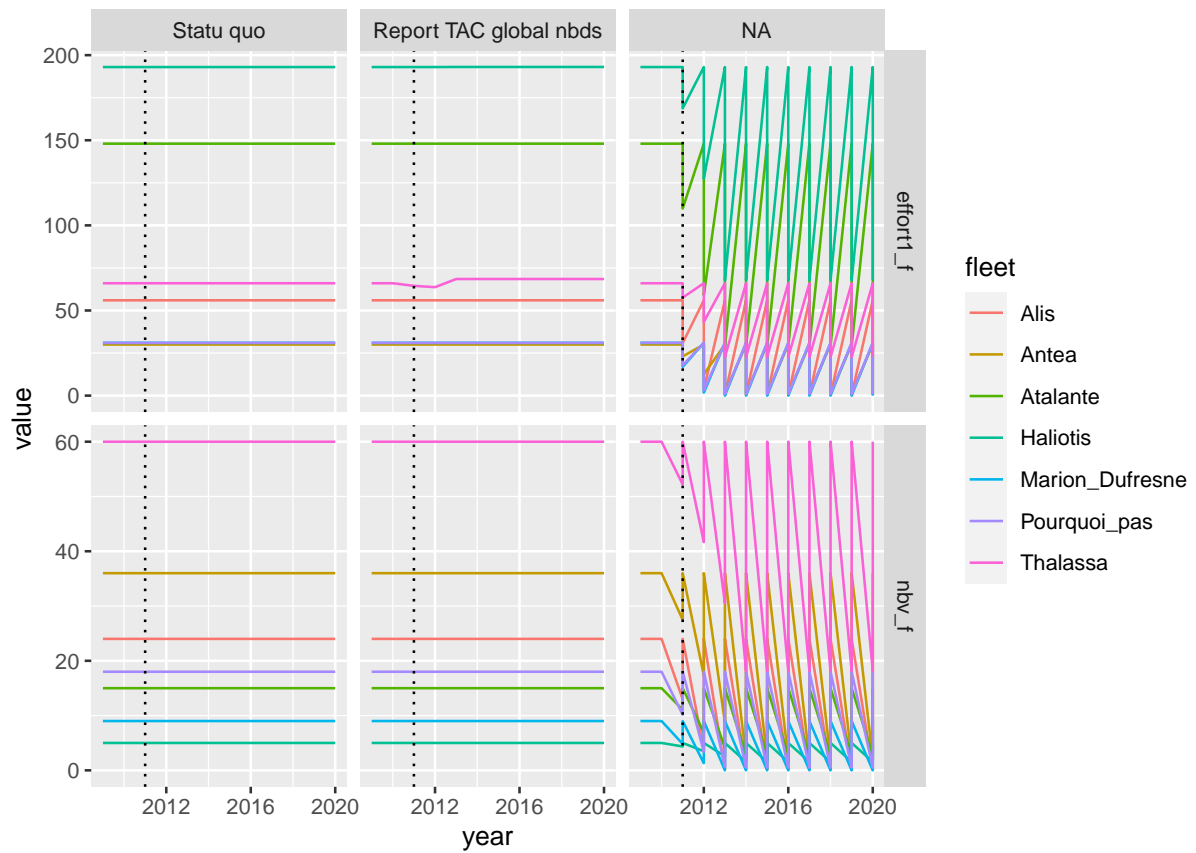


Figure 3: COR variables under global TAC gestion with effort report from Filet_COR to Filet_DP. 5 runs.



Il est bien entendu évident que les impacts de ce TAC seront perceptibles sur d'autres variables économiques.

Scénarios TAC, quotas et transition Fmsy

```
# Dépendances pour l'analyse des données et les représentations graphiques.
library(magrittr)
library(ggplot2)
library(dplyr)
library(IAM)
library(parallel)
library(beepr)
```

Le modèle bio-économique IAM a été développé notamment pour permettre d'accompagner le développement des plans de gestion des pêcheries et d'explorer les conséquences biologiques et socio-économiques de différents scénarios de TAC et quotas et de transition vers le Rendement Maximum Durable - RMD (ou Maximum Sustainable Yield - MSY).

Ce document présente la mise en œuvre avec le modèle IAM de scénarios de TAC et quotas correspondant à différents chemins de transition vers le F_{msy} (différentes valeurs de TAC constants permettant l'atteinte du FMSY plus ou moins rapidement).

L'ensemble des simulations présentées dans l'exemple sera réalisé avec le jeu de donnée exemple **Ifremer** composé de 7 flottilles et 3 espèces dynamiques dont une espèce à dynamique SS3. Les simulations prennent en compte la variabilité du recrutement et ses conséquences en termes de probabilité d'atteinte du FMSY. Dans ce cadre, une réplication de valeur N sera réalisée à partir du jeu de donnée exemple.

Attention ce document lance de nombreuses simulations $(1 + n_{TAC}) \cdot N$ et prend donc un temps long à simuler (plusieurs dizaines de minutes).

```
data("IAM_input_2009")
summary(IAM_input_2009)
#> My Input (IAM input) :
#> Simulation of 3 dynamic species, 19 static species and 7 fleet
#> Simulation start in 2009 and end in 2020 (12 steps)
#>
#> -----
#> Dynamic Species | Model |    Ages |
#>                ARC |  XSA | 0 to +gp |
#>                CDR |  XSA | 2 to +gp |
#>                DAR |  SS3 | 0 to +gp |
#> -----
#>                Fleet |    nbv |
#>                Alis |    24 |
#>                Antea |    36 |
#>                Atalante |    15 |
#>                Haliotis |     5 |
#>                Marion_Dufresne |     9 |
#>                Pourquoi_pas |    18 |
#>                Thalassa |    60 |
```

L'objet argument est laissé tel quel lors de l'utilisation de l'interface et sera édité à la main plus tard pour chaque scénario.


```
# Cette ligne ouvre une interface via une app shiny.
IAM_argum_2009 <- IAM.args(IAM_input_2009)
```

Cela revient à initialiser un objet de classe `iamArgs` sans passer par l'interface avec la commande suivante :

```
IAM_argum_2009 <- IAM.input2args(IAM_input_2009)
```

Scénario *statu quo*

Afin de pouvoir comparer l'effet de chaque scénario, il nous faut un scénario de départ dans lequel les mesures de Gestion ne s'imposent pas.

Dans un premier temps, les dynamiques de recrutements ainsi que des éléments de paramétrage du module économique sont définies. On va pour cela éditer l'objet `IAM_argum_2009`.

```
# Module SR
# Add noise to COR recruitment
IAM_argum_2009@arguments$Recruitment$COR$wnNOISEmodSR <- 0.203
IAM_argum_2009@arguments$Recruitment$COR$noiseTypeSR <- 2 # Log-normal
IAM_argum_2009@arguments$Recruitment$COR$typeMODsr <- "Hockey-Stick"
IAM_argum_2009@arguments$Recruitment$COR$parAmodSR <- 2539.5
IAM_argum_2009@arguments$Recruitment$COR$parBmodSR <- 9679

# Module EcoDCF
IAM_argum_2009 <- IAM.editArgs_Eco(IAM_argum_2009, dr = 0.04, perscCalc = 1)

# Module Gestion
mfm <- with(IAM_input_2009@input$Fleet,{
  (effort1_f_m * effort2_f_m * nbv_f_m) / as.vector(effort1_f * effort2_f * nbv_f)
})
mfm[is.na(mfm)] <- 0

IAM_argum_2009 <- IAM.editArgs_Gest(IAM_argum_2009, active = FALSE,
  delay = 1, mfm = mfm)

# Module Scenario
IAM_argum_2009 <- IAM.editArgs_Scenar(IAM_argum_2009) # deactivate scenario

summary(IAM_argum_2009)
#> My Input (IAM argument) :
#> Simulation of 3 dynamic species, 19 static species and 7 fleet
#> Simulation start in 2009 and end in 2020 (12 steps)
#>
#> =====
#> SR module | Stock Recruitment | Noise | Proba |
#> -----
#> Species | function : param A ; param B ; param C | Type : sd | Type |
#> ARC (XSA) | Mean 3.641e+07 0.00e+00 0.00e+00 | Norm | 0.00e+00 | . |
#> COR (XSA) | Hockey-Stick 2.540e+03 9.68e+03 0.00e+00 | LogN | 2.03e-01 | . |
#> DAR (SS3) | not activated 0.000e+00 0.00e+00 0.00e+00 | Norm | 0.00e+00 | . |
#> -----
#>
#> The Gestion module is not active.
#>
#> =====
```

```
#> Economic : PerscCalc = 1 ; dr = 0.040 / No replicates      /  
#> -----  
#>  
#> The Scenario module is not active.
```

Selection des différents TACs.

La recherche du TAC pour atteindre le F_{msy} se fait en simulant plusieurs scénarios correspondant à différentes valeurs de TACs simulées.

Les scénarios sont simulés avec N réplicats, correspondant à la variabilité du recrutement.

Les valeurs de N , F_{msy} et TAC sont fixées ici :

```
N <- 500  
Fmsy <- 0.26  
TACS <- seq(3200, 5600, by = 400)
```

3500 simulations seront donc effectuées. La simulation du scénario Statu Quo est codée comme suit :

```
# Statu quo  
SQ <- replicate(N, {  
  IAM::IAM.model(objArgs = IAM_argum_2009, objInput = IAM_input_2009)  
})  
  
SQ1 <- lapply(1:N, function(y) {  
  IAM.format(SQ[[y]], name = c("Fbar", "SSB", "L_et"),  
    sim_name = "SQ", n = y)  
})  
SQ <- do.call(rbind, SQ1)
```

Afin de simplifier le code, seules les variables biologiques seront extraites et représentées. Pour cela, la fonction suivante est utilisée :

```
#' Function for simulation under TAC  
#' @param x TAC value  
#' @param argum iamArgs object  
#' @param input iamInput object  
#' @param N Number of replicates  
#'  
#' @return formatted table of class iam_formtbl that regroup values  
#' for N simulation under a scenario. Only Fbar, SSB and L_et  
#' variables are extracted.  
simultac <- function(x, argum, input, N){  
  
  argum_int <- IAM.editArgs_Gest(argum, type = "x", active = TRUE,  
    tac = c(NA, NA, rep(x, 10)) )  
  
  sim <- replicate(N, {  
    IAM::IAM.model(objArgs = argum_int, objInput = input)  
  })  
  
  sim1 <- lapply(1:N, function(y) {  
    IAM.format(sim[[y]], name = c("Fbar", "SSB", "L_et"),  
      sim_name = as.character(x), n = y)  
  })  
  sim <- do.call(rbind, sim1)
```

```

    return(sim)
}

```

De même, des modifications du paramétrage sur le module Gestion ont lieu en commun pour tout les scénarios suivants. Ces modifications sont regroupées ici avec la sélection de la variables d'ajustement (`Nb trips`), la cible (`TAC`), l'espèce concernées (`COR`), un délais avant l'application de la mesure de gestion (2 ans) et comment s'applique la modulation de l'effort (multiplicatif borné entre 1e6 et -100)

```

IAM_argum_2009_TAC <- IAM.editArgs_Gest(
  IAM_argum_2009, active = FALSE, control = "Nb trips", target = "TAC",
  espece = "COR", delay = 2,
  type = "x", bounds = c(1e6, -100),
  tac = c(NA, NA, rep(3600, 10)))
  # 09, 10, 2011:2020

TACS # TACS values to remember
#> [1] 3200 3600 4000 4400 4800 5200 5600
# Warning ! Code run for long time.
x <- Sys.time()
cl <- makeCluster(detectCores()-1)
invisible(clusterEvalQ(cl, library(IAM)))

res <- parLapply(cl, as.list(TACS), simultac,
  IAM_argum_2009_TAC, IAM_input_2009, N)
stopCluster(cl)

res <- do.call(rbind, c(res, list(SQ)))
print(Sys.time() - x)
#> Time difference of 35.0032 mins

```

Représentations graphiques

On peut aisement comparer les différents scénarios en utilisant les fonctions graphiques.

```

COR <- res %>%
  filter(species == "COR") %>%
  IAM.format_quant(., probs = c(.025, .975))

COR %>%
  ggplot(aes(x = year, y = median)) +
  facet_grid(variable ~ sim_name, scales = "free_y") +
  geom_ribbon(aes(ymin = quant1, ymax = quant2), fill = "lightblue") +
  geom_line() +
  geom_line(aes(y = value), linetype = "dotted") +
  geom_vline(xintercept=2011, linetype = "dotted") +
  IAM_theme() +
  NULL

line_data <- data.frame(xintercept = c(2011, 2015),
  Lines = c("Scenarii start", "Target Fmsy"),
  linetype = c("dotted", "dashed"),
  stringsAsFactors = FALSE)

Proba <- res %>%
  filter(species == "COR") %>%

```

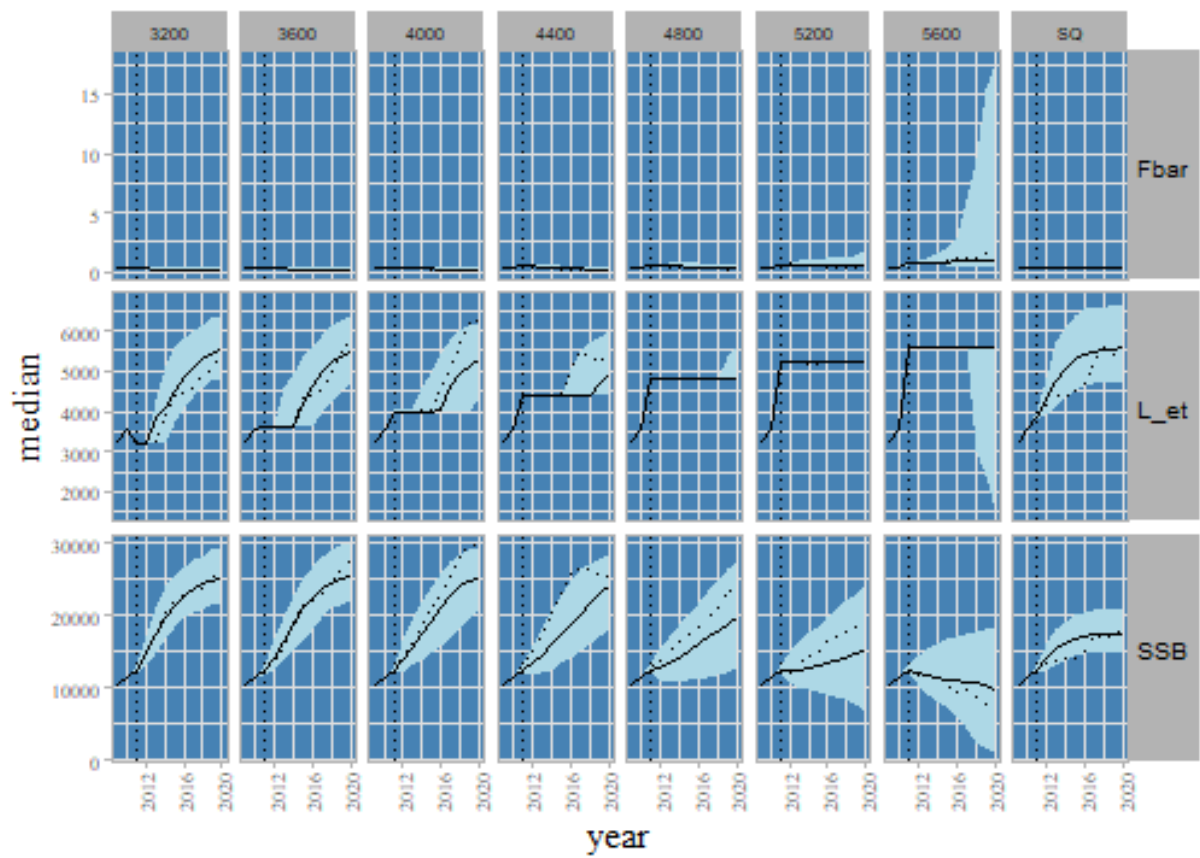


Figure 1: COR variables for different TAC scenarii. 500 runs.

```
group_by(.data$sim_name, .data$variable, .data$year) %>%
  summarize(Pfmsy = sum(.data$value <= Fmsy) / length(unique(n)) * 100,
    .groups = "keep") %>% ungroup()

Proba %>%
  filter(variable == "Fbar") %>%
  ggplot(aes(x = year, y = Pfmsy, color = as.factor(sim_name)))+
  geom_line() +
  geom_vline(xintercept= line_data$xintercept, linetype = line_data$linetype) +
  annotate("text", line_data$xintercept, 100, hjust = -.25,
    label = line_data$Lines) +
  scale_colour_discrete(name = "TAC constant") +
  NULL
```

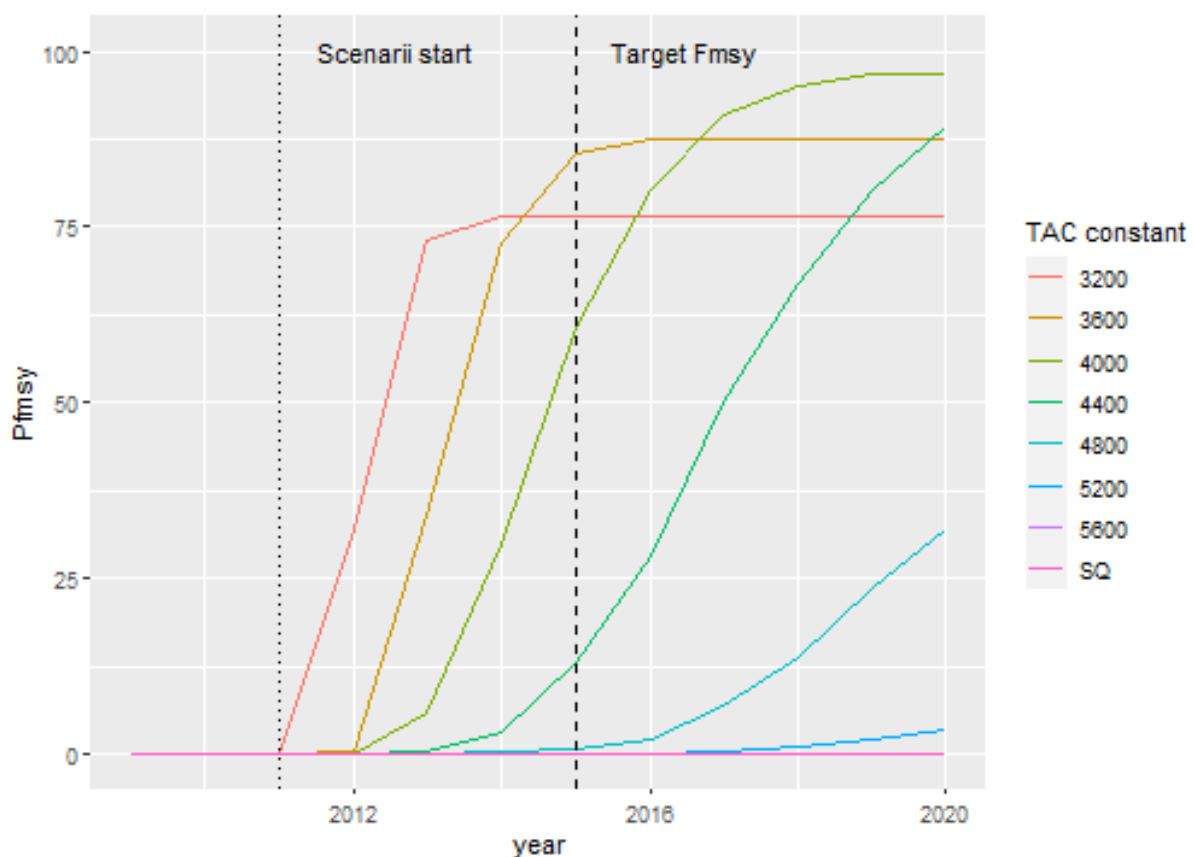


Figure 2: Fmsy Probability over 500 runs. Target Fmsy for COR species is : 0.26

```
filter(Proba, year == 2015, variable == "Fbar")
#> # A tibble: 8 x 4
#>   sim_name variable year Pfmsy
#>   <chr>    <chr>   <dbl> <dbl>
#> 1 3200     Fbar     2015  76.4
#> 2 3600     Fbar     2015  85.6
#> 3 4000     Fbar     2015  60.6
#> 4 4400     Fbar     2015   13
#> 5 4800     Fbar     2015   0.6
```

```
#> 6 5200    Fbar    2015    0
#> 7 5600    Fbar    2015    0
#> 8 SQ      Fbar    2015    0
beep::beep(3)
```

TAC jusqu'au Fmsy.

On reproduit l'analyse précédente mais en changeant la cible des scénarios. Ainsi, on va ici viser un TAC jusqu'à atteinte d'un F_{bar} . On précise pour cela un F_{bar} cible et on réduit les bornes de recherche du multiplicateur μ .

```
IAM_argum_2009_TAC_fmsy <- IAM.editArgs_Gest(
  IAM_argum_2009_TAC, active = FALSE, target = "TAC->Fbar",
  type = "x", bounds = c(100, -100),
  tac = c(NA, NA, rep(3600, 10)),
  fbar = c(NA, NA, rep(Fmsy, 10)))
  # 09, 10, 2011:2020

TACS # TACS values to remember
#> [1] 3200 3600 4000 4400 4800 5200 5600
# Warning ! Code run for long time.
x <- Sys.time()
cl <- makeCluster(detectCores()-1)
invisible(clusterEvalQ(cl, library(IAM)))

res <- parLapply(cl, as.list(TACS), simultac,
  IAM_argum_2009_TAC_fmsy, IAM_input_2009, N)
stopCluster(cl)

res <- do.call(rbind, c(res, list(SQ)))
print(Sys.time() - x) # 50 -> 3 min
#> Time difference of 40.88487 mins
```

Représentations graphiques

Les représentations graphiques reprennent le même code que précédemment.

```
COR <- res %>%
  filter(species == "COR") %>%
  IAM.format_quant(., probs = c(.025, .975))

COR %>%
  ggplot(aes(x = year, y = median)) +
  facet_grid(variable ~ sim_name, scales = "free_y") +
  geom_ribbon(aes(ymin = quant1, ymax = quant2), fill = "lightblue") +
  geom_line() + geom_point(size = .5) +
  geom_line(aes(y = value), linetype = "dotted") +
  geom_vline(xintercept=2011, linetype = "dotted") +
  IAM_theme() +
  NULL
```

Calcul de la probabilité d'atteindre le F_{msy}

On peut calculer la probabilité d'atteindre le F_{msy} comme le pourcentage de simulation ayant un $F_{bar} < F_{msy}$.

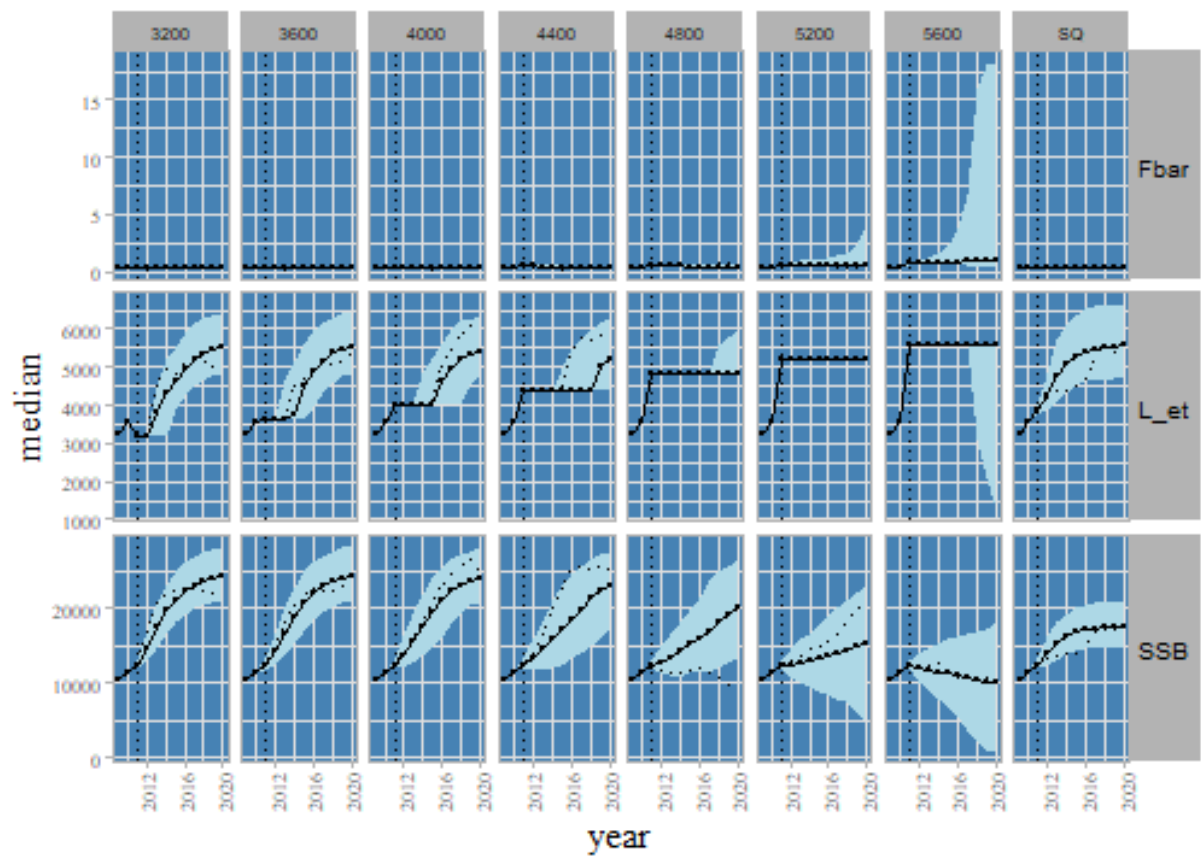


Figure 3: COR variables for different TAC scenarii. 500 runs.

```
line_data <- data.frame(xintercept = c(2011, 2015),
  Lines = c("Scenarii start", "Target Fmsy"),
  linetype = c("dotted", "dashed"),
  stringsAsFactors = FALSE)

Proba <- res %>%
  filter(species == "COR") %>%
  group_by(.data$sim_name, .data$variable, .data$year) %>%
  summarize(Pfmsy = sum(.data$value <= Fmsy) / length(unique(n)) * 100,
    .groups = "keep") %>% ungroup()

Proba %>%
  filter(variable == "Fbar") %>%
  ggplot(aes(x = year, y = Pfmsy, color = as.factor(sim_name))) +
  geom_line() +
  geom_vline(xintercept = line_data$xintercept, linetype = line_data$linetype) +
  annotate("text", line_data$xintercept, 100, hjust = -.25,
    label = line_data$Lines) +
  scale_colour_discrete(name = "TAC constant") +
  NULL
```

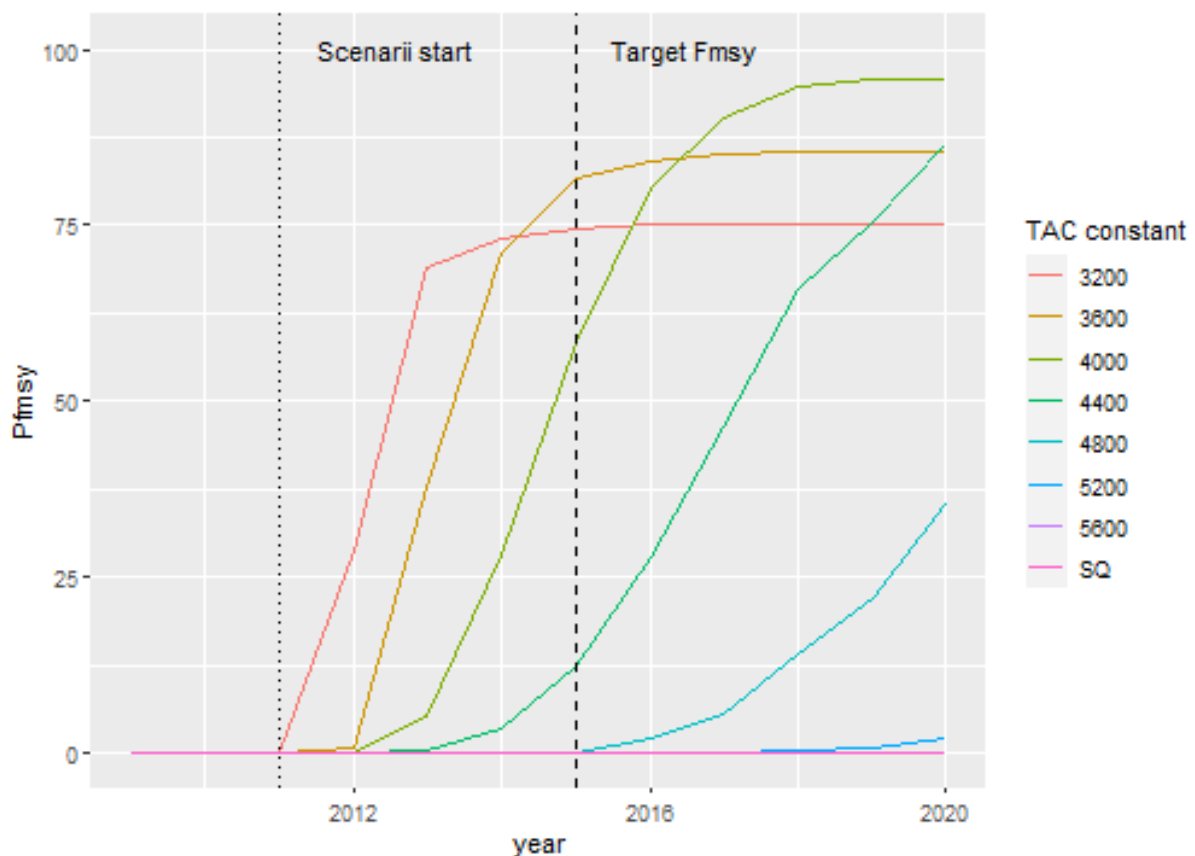


Figure 4: Fmsy Probability over 500 runs. Target Fmsy for COR species is : 0.26


```
filter(Proba, year == 2015, variable == "Fbar")
#> # A tibble: 8 x 4
#>   sim_name variable  year Pfmsy
#>   <chr>      <chr>    <dbl> <dbl>
#> 1 3200      Fbar      2015  74.6
#> 2 3600      Fbar      2015  81.6
#> 3 4000      Fbar      2015  58.2
#> 4 4400      Fbar      2015  12.4
#> 5 4800      Fbar      2015    0
#> 6 5200      Fbar      2015    0
#> 7 5600      Fbar      2015    0
#> 8 SQ       Fbar      2015    0
beep::beep(3)
```

