



Gestion F2 module équations

Maxime Jaunatre | UMR AMURE

[Mail](#) | March 10, 2022

1 Introduction

Le module GestionF2 apparait en 2018 avec les versions nommées **test** et **test2** au début de la thèse de Florence Briton.

2 Déroulé et équation

L'appel de la fonction dépend d'une condition if dans le constructeur de la classe BioE-coPar. Il faut :

- $delay \leq it$
- TACF et TAC non null
- $it \geq 1$
- t_stop egal a 0 ou supérieur à it.

2.1 $Foth_{ei}$ à $t - 1$

Avant de rentrer dans le module GestionF2, une étape préliminaire modifie $Foth_{ei}$ à $t - 1$. Cela dépend de :

- $delay \leq ind_t$
- $gestInd == 1$
- $DELAY > 0$ $DELAY = UpdtadeE$

A ce moment, si $DELAY > delay$, $DELAY = delay$

2.2 GestionF2

Initiatisation effort de base

$nbTrip_f = eff1_f = NBDSF$ (matrice vide)

$nbTrip_{fm} = eff1_{fm} = \frac{eff1_f \cdot eff2_f \cdot nbv_f \cdot mfm_{fm}}{eff1_{fm} \cdot eff2_{fm} \cdot nbv_{fm}}$

copie de $eff1_{fm}$ pour sauvegarder.

appel de **Mortalite**, **DynamicPop**, **CATCHDL**

2.2.1 Initiatisation de $Einterm_{fm}$ avec valeur max

Si NA, max est 300 NBDS.

$$denom = \sum_{m=1}^{nMe} eff1_f \cdot eff2_f \cdot nbv_f \cdot mfm_{fm} \quad (1)$$

$$\alpha_{fst1} = \frac{effSup}{denom} = \frac{effSup}{\sum_{m=1}^{nMe} eff1_f \cdot eff2_f \cdot nbv_f \cdot mfm_{fm}} \quad (2)$$

$$EffsupTMP_{fm} = \alpha_{fst1} \cdot eff1_f \cdot eff2_f \cdot nbv_f \cdot mfm_{fm} \quad (3)$$

$$= \frac{effSup \cdot eff1_f \cdot eff2_f \cdot nbv_f \cdot mfm_{fm}}{\sum_{m=1}^{nMe} eff1_f \cdot eff2_f \cdot nbv_f \cdot mfm_{fm}} \quad (4)$$

si $denom \leq 0$ alors $EffsupTMP_{fm} = \alpha_{fst1} = 0$

Remplissage de $Einterm_{fm} = -1$

appel des parties statiques et spict.

loop sur chaque espece

si $L_{efmit} + LD_{efmi} > 0$ alors $Einterm_{fm} = EffsupTMP_{fm}$

-1 restant sont remplacés par efforts initiaux (si $Einterm_{fm} < -0.5$)

$Einterm_{fm} = eff1_f \cdot eff2_f \cdot nbv_f \cdot mfm_{fm}$

2.2.2 Optimisation

copie de $Einterm_{fm}$ dans $Einterm_copy_{fm}$

mise a jour des efforts et evaluation

$$nbTrip_{fm} = eff1_{fm} = \frac{Einterm_{fm}}{eff2_{fm} \cdot nbv_{fm}} \quad (5)$$

$$nbTrip_f = eff1_f = \frac{\sum_{m=1}^{nMe} eff1_{fm} \cdot eff2_{fm} \cdot nbv_{fm}}{eff2_f \cdot nbv_f} \quad (6)$$

appel **Mortalite**

sauvegarde dans $Zt_i = out_Z_{eit}$

Boucle de $IT < itmax$ et tant que goon

reinit $Einterm_{fm}$: $Einterm_{fm} = Einterm_copy_{fm}$

$$nbTrip_f = eff1_f = eff1_copy_f \quad (7)$$

$$nbTrip_{fm} = eff1_{fm} = \frac{eff1_f \cdot eff2_f \cdot nbv_f \cdot mfm_{fm}}{eff1_{fm} \cdot eff2_{fm} \cdot nbv_{fm}} \quad (8)$$

appel **Mortalite DynamicPop**

$out_Z_{eit} = Zt_i$

appel **CatchDL**

reconciliation

$$alpha_st4_f = \frac{TAC_f}{L_{efmit} + LD_{efmi}} \quad (9)$$

$$valtest = alpha_st4_f \cdot eff1_f \cdot eff2_f \cdot nbv_f \cdot mfm_{fm} \quad (10)$$

si $L_{efmit} + LD_{efmi} > 0$ et $valtest < Einterm_{fm}$ alors $Einterm_{fm} = valtest$
une fois tout les navires de fait, on passe a TAC_{oth}

$$nbTrip_{fm} = eff1_{fm} = \frac{Einterm_{fm}}{eff2_{fm} \cdot nbv_{fm}} \quad (11)$$

$$nbTrip_f = eff1_f = \frac{\sum_{m=1}^{nMe} eff1_{fm} \cdot eff2_{fm} \cdot nbv_{fm}}{eff2_f \cdot nbv_f} \quad (12)$$

appel **Mortalite DynamicPop**

$goon = false$

si $out_Z_{eit} - Ztemp > diffZmax$ alors $goon = true$

$Ztemp = Ztemp + \lambda \cdot (out_Z_{eit} - Ztemp)$

2.2.3 Remise dans Flist

$$nbTrip_{fm} = eff1_{fm} = \frac{Einterm_{fm}}{eff2_{fm} \cdot nbv_{fm}} \quad (13)$$

$$nbTrip_f = eff1_f = \frac{\sum_{m=1}^{nMe} eff1_{fm} \cdot eff2_{fm} \cdot nbv_{fm}}{eff2_f \cdot nbv_f} \quad (14)$$