Mex Files Cequeau

Définition des intrants/extrants

Version 5.0.0

2025-04-22

# Table des matières

[Table des matières 2](#_Toc165465423)

[1 CequeauQuantiteMex 4](#_Toc165465424)

[1.1 Intrants 4](#_Toc165465425)

[1.2 execution 4](#_Toc165465426)

[1.3 parametres 4](#_Toc165465427)

[1.3.1 parametres.option 5](#_Toc165465428)

[1.3.2 parametres.sol 5](#_Toc165465429)

[1.3.3 parametres.solInitial 5](#_Toc165465430)

[1.3.4 parametres.transfert 6](#_Toc165465431)

[1.3.5 parametres.ctp 6](#_Toc165465432)

[1.3.6 parametres.lac 6](#_Toc165465433)

[1.3.7 parametres.surface 6](#_Toc165465434)

[1.3.8 parametres.fonte 6](#_Toc165465435)

[1.3.9 parametres.evapo 7](#_Toc165465436)

[1.3.10 parametres.qualite 7](#_Toc165465437)

[1.3.11 parametres.dli 8](#_Toc165465438)

[1.4 bassinVersant 8](#_Toc165465440)

[1.4.1 structure carreauEntiers 8](#_Toc165465441)

[1.4.2 structure carreauPartiel 8](#_Toc165465442)

[1.4.3 structure barrage 9](#_Toc165465444)

[1.5 meteo 10](#_Toc165465445)

[1.6 etats.quantite 10](#_Toc165465446)

[1.7 etats.qualite 11](#_Toc165465447)

[1.8 assimilations.quantite 11](#_Toc165465448)

[1.8.1 etatsCE 11](#_Toc165465449)

[1.8.2 etatsCP 12](#_Toc165465450)

[1.8.3 etatsFonte 13](#_Toc165465451)

[1.8.4 etatsEvapo 14](#_Toc165465452)

[1.8.5 etatsBarrage 14](#_Toc165465453)

[1.9 assimilations.qualite 15](#_Toc165465454)

[1.9.1 etatsCP 15](#_Toc165465455)

[1.10 PompageEau 16](#_Toc165465456)

[1.11 Extrants 16](#_Toc165465457)

[1.12 etatsCE [1 x nbPasDeTemps struct] 16](#_Toc165465458)

[1.13 etatsCP [1 x nbPasDeTemps struct] 17](#_Toc165465459)

[1.14 etatsFonte [1 x nbPasDeTemps struct] 17](#_Toc165465460)

[1.15 etatsEvapo [1 x nbPasDeTemps struct] 17](#_Toc165465461)

[1.16 etatsBarrage [1 x nbPasDeTemps struct] 17](#_Toc165465462)

[1.17 pasDeTemps 17](#_Toc165465463)

[1.18 etatsQualiteCP [1 x nbPasDeTemps struct] 17](#_Toc165465464)

[1.19 avantAssimilations 18](#_Toc165465465)

[1.20 avantAssimilationsFonte 18](#_Toc165465466)

[1.21 avantAssimilationsEvapo 18](#_Toc165465467)

[1.22 avantAssimilationsQualite 18](#_Toc165465468)

[2 CequeauInterpolationMex 18](#_Toc165465474)

[2.1 Intrants 18](#_Toc165465475)

[2.2 execution 18](#_Toc165465476)

[2.3 parametres 19](#_Toc165465477)

[2.3.1 parametres.interpolation 19](#_Toc165465478)

[2.4 bassinVersant 19](#_Toc165465479)

[2.5 stations [1 x nbStations struct] 20](#_Toc165465480)

[2.6 meteoStations 20](#_Toc165465481)

[2.7 Extrants 20](#_Toc165465482)

# CequeauQuantiteMex

**Attention** : À partir de la version 4.x.x, le mexfile « cequeauQuantiteMex » est utilisé à la fois pour la quantité et la qualité.

Si parametre.option. calculQualite = 0, une simulation quantité est exécuté. Si le paramètre = 1, la simulation de quantité est exécutée suivie de la simulation qualité.

Usage:

[etatsCE, etatsCP, etatsFonte, etatsEvapo, etatsBarrage, pasDeTemps, avantAssimilationsCE, avantAssimilationsFonte, avantAssimilationsEvapo, etatsQualCP, avAssimQual] =

CequeauQuantiteMex(execution, parametres, bassinVersant, meteo, etatsPrecedents, assimilations)

Note:

Si on ne desire pas utiliser etatsPrecedents et/ou assimilations, donner une matrice vide [] .

Pour executer la qualité en plus de la quantité, parametres.option.calculQualite = 1.

## Intrants

Ces variables sont détaillées dans les sections suivantes.

## execution

La variable « execution » contient les champs suivants.

dateDebut : Début de la simulation en format « datenum ».  
  
dateFin : Fin de la simulation en format « datenum ».  
  
resultatsIdCE (**optionnel**): [1 x nombre CE double]

Liste de id de carreaux entiers qu’on désire avoir en sortie dans la variable « etatsCE ».

resultatsIdCP (**optionnel**): [1 x nombre CP double]

Liste de id de carreaux partiels qu’on désire avoir en sortie dans la variable « etatsCP ».

## parametres

La variable « parametres » contient les champs suivants. **Tous les champs sont obligatoires.**

option: [1 x 1 struct]  
 sol: [1 x 1 struct]  
solInitial: [1 x 1 struct]  
 transfert: [1 x 1 struct]  
 ctp: [1 x nbCP double] ou 0  
 lac: [1 x nbCP double] ou 0  
 surface: [1 x nbCP double] ou 0

fonte: [1 x 1 struct]  
 evapo: [1 x 1 struct]

Le détail des champs est inclus ci-dessous. **Voir le guide Cequeau pour la signification des différents paramètres.**

### parametres.option

ipassim: double  
moduleFonte: double (modèle utilisé pour le calcul de la fonte : 1 = cequeau, 2 = cemaNeige, autres…)

moduleEvapo: double (modèle utilisé pour le calcul de l’évapotranspiration : 1 = cequeau, autres…)

calculQualite: double (0 = calcul quantité, 1 = calcul quantité et qualité)

moduleDLI: double (model utilisé pour l’irradiation descentes à ondes longues : 0 à 10)

moduleOmbrage: double (modèle utilisé pour le calcul de l’ombrage: 0 = pas d’ombrage, 1 = ombrage par hauteur d’arbre, 2 = ombrage basé sur l’indice de surface foliare)

### parametres.sol

Les variables ayant le suffixe « \_s » sont spatialisables, c’est-à-dire qu’il est possible de spécifier une valeur unique qui sera appliquée à l’ensemble des carreaux entiers ou bien des valeurs différentes pour chacun des carreaux entiers.

Dans le cas où plusieurs valeurs sont spécifiées et que le nombre de ces valeurs n’est pas identique au nombre de carreaux entier, seul la première valeur sera utilisé et elle sera appliquée à l’ensemble des carreaux entiers.

cin\_s: double ou [1 x nbCE double].  
 cvmar: double ou [1 x nbCE double].  
 cvnb\_s: double ou [1 x nbCE double].  
 cvnh\_s: double ou [1 x nbCE double].  
 cvsb: double.  
 cvsi\_s: double ou [1 x nbCE double].  
 xinfma: double  
 hinf\_s: double ou [1 x nbCE double].  
 hint\_s: double ou [1 x nbCE double].  
 hmar: double  
 hnap\_s: double ou [1 x nbCE double].  
 hpot\_s: double ou [1 x nbCE double].  
 hsol\_s: double ou [1 x nbCE double].  
hrimp\_s: double ou [1 x nbCE double].  
 tri\_s: double ou [1 x nbCE double].  
 xla: double.

### parametres.solInitial

Note : Si des états précédents sont chargés en début de simulation (voir la variable « etats »), ces valeurs ne seront pas utilisées.

hsini: double  
 hnini: double  
 hmini: double  
 q0: double

### parametres.transfert

exxkt: double  
 zn: double

### parametres.ctp

Si on ne désire pas utiliser ce paramètre, il s’agit de mettre la valeur «0 » sinon on doit spécifier les valeurs pour l’ensemble des carreaux partiels (mettre 0 aux carreaux n’ayant pas de CTP).

0 ou [1 x nbCP double].

### parametres.lac

Si on ne désire pas utiliser ce paramètre, il s’agit de mettre la valeur «0 » sinon on doit spécifier les valeurs pour l’ensemble des carreaux partiels.

0 ou [1 x nbCP double].

### parametres.surface

Si on ne désire pas utiliser ce paramètre, il s’agit de mettre la valeur «0 » sinon on doit spécifier les valeurs pour l’ensemble des carreaux partiels (mettre 0 aux carreaux dont on ne veut pas modifier la valeur).

0 ou [1 x nbCP double].

### parametres.fonte

Contient les paramètres spécifiques au modèle de fonte. Les paramètres de chacun des modèles sont contenus sous le nom du modèle, par exemple « parametres.fonte.cequeau ».

#### parametres.fonte.cequeau

Les variables ayant le suffixe « \_s » sont spatialisables, c’est-à-dire qu’il est possible de spécifier une valeur unique qui sera appliquée à l’ensemble des carreaux entiers ou bien des valeurs différentes pour chacun des carreaux entiers.

Dans le cas où plusieurs valeurs sont spécifiées et que le nombre de ces valeurs n’est pas identique au nombre de carreaux entier, seul la première valeur sera utilisé et elle sera appliquée à l’ensemble des carreaux entiers.

strne\_s: double ou [1 x nbCE double]   
 tfc\_s: double ou [1 x nbCE double]  
 tfd\_s: double ou [1 x nbCE double]  
 tsc\_s: double ou [1 x nbCE double]  
 tsd\_s: double ou [1 x nbCE double]  
 ttd: double  
 tts\_s: double ou [1 x nbCE double]

jonei: double  
 tmur: double  
 tstock: double

### parametres.evapo

Contient les paramètres spécifiques au modèle d’évapotranspiration. Les paramètres de chacun des modèles sont contenus sous le nom du modèle, par exemple « parametres.evapo.cequeau ».

#### parametres.evapo.cequeau

joeva: double  
 evnap: double

xaa: double  
 xit: double.

### parametres.qualite

Paramètres propres à la simulation de qualité.

#### parametres.qualite.cequeau

**Voir le guide Cequeau pour la signification des différents paramètres de qualité.**

paramode: [1 x 1 struct]

coprom: double

colarg: double

temperat: [1 x 1 struct]

##### parametres.qualite.cequeau.paramode

Ce paramètre est utilisé pour déterminer les types de simulation de qualité désirés. Comme la version actuelle simule seulement la température, **ce paramètre n’est pas utilisé**.

##### parametres.qualite.cequeau.temperat

crayso: double

crayin: double

cevapo: double

cconve: double

crigel: double

tnap: double

panap: double

tinit: double

bassol: double

corsol: double

radiasol: [1 x 12 double]

nebulosi: [1 x 12 double]

pressvap: [1 x 12 double]

vitesven: [1 x 12 double]

### parametres.dli

Ces structures contiennent des paramètres pour calculer l'irradiance à longue longueur d'onde.

m1: [1 x 1 struct]

m2: [1 x 1 struct]

m3: [1 x 1 struct]

m4: [1 x 1 struct]

m5: [1 x 1 struct]

m6: [1 x 1 struct]

m7: [1 x 1 struct]

m8: [1 x 1 struct]

m9: [1 x 1 struct]

### parametres.pompage

delai: double

conductiviteHydraulique\_s: double ou[1 x nbCE double]

coeffPompage: double

## bassinVersant

Cette variable contient la définition de bassin versant selon les données physiographiques préparées.

nbCpCheminLong: double  
 superficieCE: double  
 nomBassinVersant: string  
 carreauxEntiers: [1 x nbCE]  
 carreauxPartiels: [1 x nbCP]   
barrage (optionnel): [1 x 1 struct]

puits: [1 x nbPuits]

### structure carreauEntiers

i: double  
 j: double  
pctLacRiviere: double  
 pctForet: double  
 pctMarais: double

altitude: double

Latitude: double

Longitude: double

meanSlope: double

### structure carreauPartiel

i: double  
 j: double  
 code: double (code ascii de A,B,C ou D)  
 pctSurface: double  
 idCPAval: double  
 idCPsAmont: [1 x 5 double]  
 idCE: double  
 pctEau: double  
 pctForet: double  
 pctMarais: double  
 pctSolNu: double  
 altitudeMoy: double  
 profondeurMin: double  
 longueurCoursEauPrincipal: double  
 largeurCoursEauPrincipal: double  
 penteRiviere: double  
 cumulPctSuperficieCPAmont: double  
 cumulPctSuperficieLacsAmont: double  
cumulPctSuperficieMaraisAmont: double  
 cumulPctSuperficieForetAmont: double

lat: double

lon: double

hautMoyenneArbre: double

azimutCoursEau: double

### structure barrage

Cette structure varie selon le type de barrage.

**Type2** :

idCP: double  
 idCPAval: double  
 type: 2  
volumeInitial: double  
 niveau: [1 x 7 double]  
 debit: [1 x 7 double]

**Type3** :

Barrage externe au bassin versant dont les évacuations se font sur un carreau partiel du bassin versant; ces débits sont connus.

On utilise la même structure que pour le type 2 mais seulement les champs surlignés sont nécessaires, donc la valeur des autres champs n’a pas d’importance (il est suggéré d’y mettre 0).

idCP: double   
 idCPAval: double  
 type: 3  
volumeInitial: double  
 niveau: [1 x 7 double]  
 debit: [1 x nbPasDeTemps double]

**Type5** :

idCP: double  
 idCPAval: double  
 type: 5  
volumeInitial: double  
 niveau: [1 x 7 double]  
 volume: [1 x 7 double]  
debitsInterne: [nb règles évacuation interne x 7 double]

debitsExterne: [1 x nb règles évacuation externe struct]

où debitsExterne :

dateDebut: double (date format YYYYMMDD)  
 dateFin: double (date format YYYYMMDD)  
parametres: [1 x 7 double]

***Type10****:*

*Nouveau type de barrage (inexistant dans Cequeau original) dont le débit est fonction de l’ouverture des vannes et du niveau.*

*Note : Les calculs effectués avec ce type de barrage sont propres aux ouvrages RTA. Autrement dit ce type de barrage est utilisable seulement dans un contexte RTA.*

*idCP: double   
 idCPAval: double  
 type: 10  
volumeInitial: double  
 ouverture: [1 x nbPasDeTemps double]*

### structure puits

idCE: double  
distanceRiviere: double  
 debitPompage: [nbPasDeTemps x 1 double]

niveauxPuits: [nbPasDeTemps x 1 double]

h0: double

active: double (0 = désactivé, 1 = activé)

## meteo

Données météo pour le période à simuler. Les données par pas de temps doivent suivre le même ordre que les carreaux entiers dans Cequeau.

tMin: [nbPasDeTemps x nbCE double ou single]  
 tMax: [nbPasDeTemps x nbCE double ou single]  
 pTot: [nbPasDeTemps x nbCE double ou single]  
 pluie: [nbPasDeTemps x nbCE double ou single]

neige: [nbPasDeTemps x nbCE double ou single]

rayonnement: [nbPasDeTemps x nbCE double ou single]

nebulosite: [nbPasDeTemps x nbCE double ou single]

pression: [nbPasDeTemps x nbCE double ou single]

vitesseVent: [nbPasDeTemps x nbCE double ou single]

t: [nbPasDeTemps x nbCE double ou single]

lai: [nbPasDeTemps x nbCE double ou single]

lai\_norm: [nbPasDeTemps x nbCE double ou single]

dli: [nbPasDeTemps x nbCE double ou single]

**Note :** Le type de donnée pour la météo peut « double » ou « single » mais pas les deux en même temps.

## etats.quantite

Cette variable est optionnelle et permet d’initialiser une simulation quantité avec les états finaux d’une simulation précédente.

etatsCE : États des carreaux entiers pour un pas de temps.  
etatsCP : États des carreaux partiels pour un pas de temps.  
etatsFonte : États de la fonte sur les carreaux entiers pour un pas de temps.  
etatsEvapo : États de l’évapotranspiration sur les carreaux entiers pour un pas de temps.  
etatsBarrage (optionnel): États des barrages pour un pas de temps.

Plutôt que de donner les détails des de ces trois champs, voici un exemple d’initialisation des états à partir d’une simulation précédente.

etatsPrecedents.quantite.etatsCE = etatsCE(end)  
etatsPrecedents.quantite.etatsCP = etatsCP(end)  
etatsPrecedents.quantite.etatsFonte = etatsFonte (end)  
etatsPrecedents.quantite.etatsEvapo = etatsEvapo (end)  
etatsPrecedents.quantite.etatsBarrage = etatsBarrage(end)

## etats.qualite

Cette variable est optionnelle et permet d’initialiser une simulation qualité avec les états finaux d’une simulation précédente.

etatsCP : États des carreaux partiels pour un pas de temps.

Plutôt que de donner les détails des de ces trois champs, voici un exemple d’initialisation des états à partir d’une simulation précédente.

etatsPrecedents.qualite.etatsCP = etatsQualCP(end)

## assimilations.quantite

Pour chaque pas de temps où l’on désire faire de l’assimilation sur la quantité on doit avoir la structure suivante :

pasDeTemps: double (pas de temps en format datenum)  
 etatsCE: Voir section suivante   
 etatsCP: Voir section suivante  
 etatsFonte: Voir section suivante   
 etatsEvapo: Voir section suivante  
 etatsBarrage: Voir section suivante

Si on ne désire pas utiliser « etatsCE », « etatsCP », « etatsFonte », « etatsEvapo » ou bien « etatsBarrage » pour un pas de temps, donner une matrice vide comme valeur. Par exemple :

pasDeTemps: 734549  
 etatsCE: [1x1 struct]  
 etatsCP: []  
 etatsFonte: [1x1 struct]  
 etatsEvapo: []  
 etatsBarrage: []

### etatsCE

id: [1 x nbAssimilations double]  
 niveauEauSol: [1 x nbAssimilations double]  
 niveauEauNappe: [1 x nbAssimilations double]  
 niveauEauLacsMarais: [1 x nbAssimilations double]  
 evapoPotJour: [1 x nbAssimilations double]  
 production: [1 x nbAssimilations double]

* Si on ne désire pas utiliser une des variables donner une matrice vide comme valeur.
* Si on ne désire pas donner une valeur à tous les carreaux, donner « NaN » comme valeur.

Exemple :

id: [1 2 3 4 5]  
 niveauEauSol: [10 10 NaN NaN NaN] niveauEauNappe: []  
 niveauEauLacsMarais: []  
 evapoPotJour: []  
 production: []

#### Format alternatif

Sauf pour le champ « id », il est possible de donner un facteur multiplicatif et additif plutôt qu’une valeur.

id: [1 x nbAssimilations double]  
 niveauEauSol: [**2** x nbAssimilations double]  
 niveauEauNappe: [**2** x nbAssimilations double]  
 niveauEauLacsMarais: [**2** x nbAssimilations double]  
 evapoPotJour: [**2** x nbAssimilations double]  
 production: [**2** x nbAssimilations double]

* Il est possible d’utiliser les 2 formats en même temps mais seulement pour des champs différents.
* Les deux facteurs doivent être présents sinon ils ne seront pas utilisés.
* Comme précédemment, si on ne désire pas donner une valeur à tous les carreaux, donner « NaN » à la première valeur.

Exemple :

id: [1 2 3 4 5]  
 niveauEauSol: [1.1 1.2 1.3 1.4 1.5 *<= facteurs multiplicatifs*

2.0 3.0 4.0 5.0 6.0] <= *facteurs additifs*  
 niveauEauNappe: [2.1 2.2 2.3 NaN NaN *<= facteurs multiplicatifs*

2.0 3.0 4.0 NaN NaN] <= *facteurs additifs* niveauEauLacsMarais: []  
 evapoPotJour: []  
 production: []

Dans cet exemple, le « niveauEauSol» du carreau entier #1  sera assimilé ainsi:  
valeurAssimilée = valeurCalculée \* 1.1 + 2.0

### etatsCP

id: [1 x nbAssimilations double]  
apport: [1 x nbAssimilations double]  
volume: [1 x nbAssimilations double]  
 debit: [1 x nbAssimilations double]

* Si on ne désire pas utiliser une des variables donner une matrice vide comme valeur.
* Si on ne désire pas donner une valeur à tous les carreaux, donner « NaN » comme valeur.

Exemple :

id: [1 2 3 4 5]  
apport: []  
volume: [55.4991 6.4084 0.0874 0.0041 48.9928]  
 debit: []

#### Format alternatif

Sauf pour le champ « id », il est possible de donner un facteur multiplicatif et additif plutôt qu’une valeur.

id: [1 x nbAssimilations double]  
apport: [**2** x nbAssimilations double]  
volume: [**2** x nbAssimilations double]  
 debit: [**2** x nbAssimilations double]

* Il est possible d’utiliser les 2 formats en même temps mais seulement pour des champs différents.
* Les deux facteurs doivent être présents sinon ils ne seront pas utilisés.
* Comme précédemment, si on ne désire pas donner une valeur à tous les carreaux, donner « NaN » à la première valeur.

Exemple :

id: [1 2 3 4 5]  
apport: []  
volume: [1.1 1.2 NaN 1.4 NaN *<= facteurs multiplicatifs*

2.0 3.0 NaN 5.0 NaN] <= *facteurs additifs*  
 debit: []

Dans cet exemple, le « volume » du carreau partiel #1  sera assimilé ainsi:  
valeurAssimilée = valeurCalculée \* 1.1 + 2.0

### etatsFonte

**Note : Dépend du modèle de fonte. Pour Cequeau :**

id: [1 x nbAssimilations double]  
 stockNeigeForet: [1 x nbAssimilations double]  
 stockNeigeClairiere: [1 x nbAssimilations double]  
indexMurissementNeige: [1 x nbAssimilations double]  
 indexTempNeige: [1 x nbAssimilations double]

* Si on ne désire pas utiliser une des variables donner une matrice vide comme valeur.
* Si on ne désire pas donner une valeur à tous les carreaux, donner « NaN » comme valeur.

Exemple :

id: [1 2 3 4 5]  
 stockNeigeForet: [30 30 30 30 30]  
 stockNeigeClairiere: [30 30 30 30 30]  
indexMurissementNeige: []  
 indexTempNeige: []

#### Format alternatif

Sauf pour le champ « id », il est possible de donner un facteur multiplicatif et additif plutôt qu’une valeur.

id: [1 x nbAssimilations double]  
 stockNeigeForet: [**2** x nbAssimilations double]  
 stockNeigeClairiere: [**2** x nbAssimilations double]  
indexMurissementNeige: [**2** x nbAssimilations double]  
 indexTempNeige: [**2** x nbAssimilations double]

* Il est possible d’utiliser les 2 formats en même temps mais seulement pour des champs différents.
* Les deux facteurs doivent être présents sinon ils ne seront pas utilisés.
* Comme précédemment, si on ne désire pas donner une valeur à tous les carreaux, donner « NaN » à la première valeur.

Exemple :

id: [1 2 3 4 5]  
 stockNeigeForet: [1.1 1.2 1.3 1.4 1.5 *<= facteurs multiplicatifs*

2.0 3.0 4.0 5.0 6.0] <= *facteurs additifs*  
 stockNeigeClairiere: [2.1 2.2 2.3 NaN NaN *<= facteurs multiplicatifs*

2.0 3.0 4.0 NaN NaN] <= *facteurs additifs*  
indexMurissementNeige: []  
 indexTempNeige: []

Dans cet exemple, le « stockNeigeForet » du carreau entier #1  sera assimilé ainsi:  
valeurAssimilée = valeurCalculée \* 1.1 + 2.0

### etatsEvapo

**Note : Dépend du modèle de fonte. Pour Cequeau, pas d’états. Pour un autre modèle, voir la structure de etatsFonte.**

### etatsBarrage

idCarreauPartiel: [1 x nbAssimilations double]  
volume: [1 x nbAssimilations double]  
niveau: [1 x nbAssimilations double]

* Si on ne désire pas utiliser une des variables donner une matrice vide comme valeur.
* Si on ne désire pas donner une valeur à tous les barrages donner « NaN » comme valeur.

Exemple :

idCarreauPartiel: [581]  
volume: [70.25]  
niveau: []

#### Format alternatif

Sauf pour le champ « idCarreauPartiel », il est possible de donner un facteur multiplicatif et additif plutôt qu’une valeur.

idCarreauPartiel: [1 x nbAssimilations double]  
volume: [**2** x nbAssimilations double]  
niveau: [**2** x nbAssimilations double]

* Il est possible d’utiliser les 2 formats en même temps mais seulement pour des champs différents.
* Les deux facteurs doivent être présents sinon ils ne seront pas utilisés.
* Comme précédemment, si on ne désire pas donner une valeur à tous les carreaux, donner « NaN » à la première valeur.

Exemple :

idCarreauPartiel: [581]  
volume: [1.1

2.0]  
niveau: []

Dans cet exemple, le « volume » du barrage sur le carreau partiel #581  sera assimilé ainsi:  
valeurAssimilée = valeurCalculée \* 1.1 + 2.0

## assimilations.qualite

Pour chaque pas de temps où l’on désire faire de l’assimilation sur la qualite on doit avoir la structure suivante :

pasDeTemps: double (pas de temps en format datenum)  
 etatsCP: Voir section suivante

### etatsCP

id: [1 x nbAssimilations double]  
temperature: [1 x nbAssimilations double]

* Si on ne désire pas donner une valeur à tous les carreaux spécifiés, donner « NaN » comme valeur à la température.

Exemple :

id: [1 2 3 4 5]  
temperature: [10 10.5 10 10.75 NaN]

#### Format alternatif

Sauf pour le champ « id », il est possible de donner un facteur multiplicatif et additif plutôt qu’une valeur.

id: [1 x nbAssimilations double]  
temperature: [**2** x nbAssimilations double]

* Les deux facteurs doivent être présents sinon ils ne seront pas utilisés.
* Comme précédemment, si on ne désire pas donner une valeur à tous les carreaux, donner « NaN » à la première valeur.

Exemple :

id: [1 2 3 4 5]  
temperature: [1.1 1.2 NaN 1.4 NaN *<= facteurs multiplicatifs*

2.0 3.0 NaN 5.0 NaN] <= *facteurs additifs*

Dans cet exemple, la « température » du carreau partiel #1  sera assimilé ainsi:  
valeurAssimilée = valeurCalculée \* 1.1 + 2.0

## Extrants

Les variables de sortie sont détaillées ci-dessous. Un journal d’exécution dont le nom est « CequeauQuantite.log » est aussi produit dans le même répertoire que celui d’exécution.

Note : Peut ne pas être présent selon une option de compilation.

## etatsCE [1 x nbPasDeTemps struct]

id: [1 x nbCE double]  
 iCarreauEntier: [1 x nbCE double]  
 jCarreauEntier: [1 x nbCE double]  
 niveauEauSol: [1 x nbCE double]  
 niveauEauNappe: [1 x nbCE double]  
 niveauEauLacsMarais: [1 x nbCE double]  
 evapoPotJour: [1 x nbCE double]  
 production: [1 x nbCE double]

## etatsCP [1 x nbPasDeTemps struct]

id: [1 x nbCP double]  
 apport: [1 x nbCP double]  
 volume: [1 x nbCP double]   
 debit: [1 x nbCP double]

## etatsFonte [1 x nbPasDeTemps struct]

**Note : Dépend du modèle de fonte. Pour Cequeau :**

stockNeigeForet: [1 x nbCE double]  
 stockNeigeClairiere: [1 x nbCE double]  
indexMurissementNeige: [1 x nbCE double]  
 indexTempNeige: [1 x nbCE double]

## etatsEvapo [1 x nbPasDeTemps struct]

**Note : Dépend du modèle d’évapotranspiration. Pour Cequeau, la seule variable d’état est « evapoPotJour » mais elle est conservée dans les états des CE pour les besoins de la simulation Qualité.**

## etatsBarrage [1 x nbPasDeTemps struct]

idCarreauPartiel: double  
 volume: double  
 niveau: double

debitEntrant: double

debitsSortants: [1 x nbReglesEvacuation double]

## pasDeTemps

Contient la date en format « datenum » à la fin de chaque pas de temps.

## etatsQualiteCP [1 x nbPasDeTemps struct]

Si la simulation qualité n’est pas exécutée, contient une structure vide.

id: [1 x nbCP double] : Id du carreau partiel.

temperature: [1 x nbCP double] : Température de l’eau au CP.

qruiss: [1 x nbCP double]

qnappe: [1 x nbCP double]

qhypo: [1 x nbCP double]

qlacma: [1 x nbCP double]

qradso: [1 x nbCP double]

qradin: [1 x nbCP double]

qevap: [1 x nbCP double]

qconv: [1 x nbCP double]

Les champs suivant le champ température sont les valeurs calculées participant au calcul du bilan thermique.

* Énergies associées aux apports advectifs locaux d'eau :
  + qruiss : Ruissellement de surface.
  + qnappe : Eaux provenant de la nappe
  + qhypo : Ruissellement hypodermique
  + qlacma : Debordement des lacs et marais
* Échanges d’énergies à la surface de l’eau
  + qradso : Radiation solaire
  + qradin : Radiation infrarouge
  + qevap : Évaporation
  + qconv : Convection

## avantAssimilations

Sauvegarde des états avant assimilation pour chaque pas de temps d’assimilation. La structure est la même que l’intrant « [Assimilations.quantite](#_Assimilations.quantite) ».

## avantAssimilationsFonte

Sauvegarde des états avant assimilation pour chaque pas de temps d’assimilation. La structure est la même que l’intrant « [assimilations.quantite.etatsFonte](#_etatsFonte) ».

## avantAssimilationsEvapo

Sauvegarde des états avant assimilation pour chaque pas de temps d’assimilation. La structure est la même que l’intrant « « [Assimilations.quantite.etatsEvapo](#_etatsEvapo) ».

## avantAssimilationsQualite

Sauvegarde des états avant assimilation pour chaque pas de temps d’assimilation. La structure est la même que l’intrant « [Assimilations.qualite](#_Assimilations.qualite)  ».

# CequeauInterpolationMex

Usage :

meteoInterpolee = cequeauInterpolationMex(execution, parametres, bassinVersant, stations, meteoStations)

## Intrants

Certains intrants de la simulation quantité ont été réutilisés pour cette version initiale de l’interpolateur. Bien que tous les champs doivent être présents, pour certains intrants seulement quelques champs sont utilisés. Les précisions suivent ci-dessous.

## execution

La variable « execution » contient les champs suivants.

dateDebut : Date de début des données à interpoler en format « datenum ».  
  
dateFin : Date de fin des données à interpoler en format « datenum ».

## parametres

La variable « parametres » contient les champs suivants. **Tous les champs sont obligatoires.**

*neige: [1 x 1 struct]  
 option: [1 x 1 struct]  
 sol: [1 x 1 struct]  
solInitial: [1 x 1 struct]  
 transfert: [1 x 1 struct]  
 ctp: [1 x nbCP double] ou 0  
 lac: [1 x nbCP double] ou 0  
 surface: [1 x nbCP double] ou 0*

**interpolation**: [1 x 1 struct]

**Note :** Les champs avant « interpolation » (*en italique*) sont obligatoires mais ne sont pas utilisés par l’interpolateur. Donc leur contenu n’a pas d’influence sur l’interpolation. Seule les valeurs contenues dans le champ « interpolation » sont importantes.

### parametres.interpolation

type: double

coep: double

coet: double

type : 1: par les polygones de Thiessen.

3: Par pondération des trois stations les plus proches.

coep : Coefficient de correction des précipitations annuelles en fonction de

l'altitude (mm/mètre/an).

coet : Correction des températures en fonction de l'altitude (degC/1 000 m).

## bassinVersant

Voir [bassinVersant](#_bassinVersant) de CequeauQuantiteMex.

Un champ supplémentaire est nécessaire dans la structure carreauEntier pour permettre l’interpolation, il s’agit de l’altitude. Donc si on reprend la définition de cette structure dans la section CequeauQuantiteMex :

i: double  
 j: double  
pctLacRiviere: double  
 pctForet: double  
 pctMarais: double

**altitude**: double

## stations [1 x nbStations struct]

id: char

nom: char

i: double

j: double

tp: double

altitude: double

où

id : Identifiant de la station.

nom : Nom de la station. Ce champ n’est pas utilisé par l’interpolateur.

i : abscisse de la station dans le système de coordonnées "I et J" du bassin.

J : ordonnée de la station dans le système de coordonnées "I et J" du bassin.

tp : précipitation moyenne interannuelle de la station (mm).

altitude : Altitude de la station.

## meteoStations

La structure est la même que [meteo](#_meteo) de cequeauQuantiteMax, par exemple tMin :

tMin: [nbPasDeTemps x nbCE double ou single]

sauf qu’on donne la météo uniquement aux stations. Donc par exemple pour tMin :

tMin: [nbPasDeTemps x nbStations double ou single]

Important : La météo doit être donnée dans le même ordre que les stations dans la variable stations de la section précédente.

## Extrants

Un seul extrant, la météo interpolée. Cet extrant à la même structure que l’intrant [météo](#_meteo) de cequeauQuantiteMex.