

INTERFACE POM - TÉLÉMAC

Spécifications Fonctionnelles Détaillées

Réf. POM3-CS-2023-DS-PIM-015

	Nom	Société	Fonction	Date	Visa
Rédigé par :	M. Renon	CS GROUP	Equipe CS		
Validé par :	C. Mertz	CS GROUP	Chef de projet		
Pour application :	J. Covès	CS GROUP	Directeur de projet		

CS GROUP
6 rue Brindejonc des Moulinais
Parc de la Grande Plaine
BP 15872
31506 Toulouse Cedex 5

Ed.	Rév.	DATE	MOTIF
01	00	06/10/15	Création du document
01	01	19/10/15	Prise en compte des remarques
01	02	17/11/15	Prise en compte des remarques
01	03	20/11/15	Prise en compte des remarques
01	04	23/11/15	Prise en compte des remarques
02	01	14/06/2018	Spécifications v2.0
02	02	18/07/2018	Prise en compte des remarques, déplacement de chapitres dans les spécifications PI v3
02	03	04/09/2018	Prise en compte des remarques
02	04	07/09/2018	Prise en compte des remarques
02	05	17/09/2018	Méthode d'initialisation « run permanent »
02	06	18/01/2019	Prise en compte des remarques
02	07	10/01/2019	Gestion des dates T0, T1, T2 ; Suppression de l'étape de compilation ; Gestion des fichiers de reprise ;
3.1	00	23/10/2020	Développements du PIM3 sur le PI3 [#194913] extension des données sur les bornes temporelles
3.1	01	11/12/2020	[#198284] ajout de la clef « supplmodelfiles »
3.2	00	02/02/2023	Prise en compte de la nouvelle VM Mascaret Debian10 Gestion séparateur « : » dans les fichiers « .cas » (§3.6.3.5) Mise à jour des options des dossiers modèles (§1.6.2)
03	01	14/02/2023	Prise en compte des retours du SCHAPI
03	02	27/06/2023	Passage à PIT 4.0 et VM Debian 11 Suppression des anciennes clés (modelsubdirectory) (§ 1.7.2)
04	00	18/08/2023	Mise à jour du compute mode (cf §2.1)

Sommaire

Table des matières

1. Généralités.....	5
1.1 Objet du document.....	5
1.2 Glossaire.....	5
1.3 Présentation globale du système.....	5
1.3.1 But.....	5
1.3.2 Contexte du système.....	5
1.4 Architecture générale.....	6
1.4.1 Architecture matérielle.....	6
1.4.2 Architecture logicielle.....	6
1.5 Version des composants.....	6
1.5.1 Version PI.....	6
1.5.2 Version POM.....	7
1.5.3 Version Télémac.....	7
1.5.4 Version Python.....	7
1.6 Interfaces.....	7
1.6.1 Généralités.....	7
1.6.2 Paramétrage du PIT.....	7
1.6.3 Formats de fichiers.....	9
1.6.3.1 LIQ.....	9
1.6.3.2 Selafin (slf).....	9
1.7 Arborescence générale.....	10
1.7.1 Arborescences des fichiers Telemac.....	10
1.7.1.1 Arborescence d'un modèle Telemac.....	10
1.7.2 Arborescence de travail du PIT.....	11
1.7.3 Nomenclatures.....	12
1.8 Installation.....	12
1.9 Lancement de Télémac.....	12
2. Adaptations du PI v3.....	14
2.1 Mode de calcul.....	14
2.2 Nettoyage des fichiers.....	14
2.3 Calculs en parallèle.....	14
2.4 Un « run » par scénario.....	14
2.5 Gestion de l'avancement.....	14
2.6 Gestion des unités.....	15
2.7 Dates des simulations analyse/prévision.....	15
3. Périmètre fonctionnel du PIT.....	17
3.1 Lancement.....	17
3.2 Ordonnanceur de tâches.....	17
3.3 Initialisation.....	17
3.4 Traitement des scénarios.....	18
3.5 Traitement des entrées.....	18
3.6 Runs.....	18
3.6.1 Types de run.....	18
3.6.2 Copie des fichiers du modèle.....	18
3.6.3 Paramétrage de la plateforme.....	19
3.6.3.1 Génération du fichier « .liq ».....	19
3.6.3.2 Génération du fichier des sections de contrôle.....	19

3.6.3.3 Modification du fichier « .cas ».....	20
3.6.3.4 Gestion du séparateur des fichiers « .cas ».....	21
3.6.4 Lancement du run.....	21
3.6.5 Vérification de la bonne exécution du run.....	21
3.7 Traitement des sorties.....	21
3.7.1 Fichiers de reprises.....	22
3.7.2 Sorties de type « fichier ».....	22
3.7.3 Sorties XML Sandre.....	23
3.7.3.1 Lecture du fichier SLF.....	23
3.7.3.2 Lecture du fichier texte des débits.....	23
3.8 Finalisation.....	24

Liste des tableaux

Tableau 1 : Glossaire.....	5
Tableau 2 : Architecture matérielle.....	6

1. Généralités

1.1 Objet du document

Ce document présente la spécification logicielle du programme d'interface entre la plateforme opérationnelle pour la modélisation (POM) et le modèle Télémac. Ce programme est nommé PIT (Programme d'Interface avec Télémac).

1.2 Glossaire

Acronyme	Signification
BDH	Base de Données Hydro.
BDTR	Également appelée BDTR (Base de Données Temps Réel).
PHYC/PHYL	Également appelée PHYC/PHYL (Plateforme Hydro Centrale/Locale) pour inclure ses services web.
CS	Communication System
IHM	Interface Homme Machine
PI	Librairie PomInterface ou Programme d'Interface générique
PIT	Programme d'Interface avec Télémac
POM	Plateforme Opérationnelle pour la Modélisation
SCHAPI	Service Central d'Hydrométéorologie et d'Appui à la Prévision des Inondations
SPC	Service de Prévision des Crues

Tableau 1 : Glossaire

1.3 Présentation globale du système

1.3.1 But

La POM est un outil du SCHAPI permettant de lancer des calculs de prévision des crues sur différentes plate-formes de modèles. Parmi ces plates-formes, certaines sont recommandées nationalement par le SCHAPI : Mascaret, Telemac, GRP, Plathynes.

La POM entre en interaction avec ces plates-formes selon un protocole standardisé. Les modèles doivent donc s'interfacer avec ce protocole pour interagir avec la POM.

Le présent document doit donc définir les méthodes et procédures à implémenter pour piloter Télémac à partir des informations fournies par la POM, d'une manière semblable aux modèles ci-dessus.

1.3.2 Contexte du système

Conceptuellement, le serveur POM et le serveur « Télémac » (machine virtuelle) sont des serveurs distants. Le présent logiciel est lancé à distance par la POM à l'aide du protocole SSH.

L'exécutable doit donc être lancé en ligne de commande avec pour paramètre

- ✓ le nom du fichier de paramétrage que la POM fournit (« parameters.xml »)
- ✓ le nom du fichier de paramétrage du PIT (cf. 1.6.2)

Le chapitre 1.4.1 détaille la chaîne d'exécutables nécessaires au lancement de Télémac par la POM, via PIM (le présent exécutable). Du point de vue de la POM, cette distinction doit être transparente.

Cette spécification présente la version 2.0 du PIT, basée sur la version 1.0 précédente.

1.4 Architecture générale

1.4.1 Architecture matérielle

Le programme d'interface doit être déployé sur le serveur Télémac (machine virtuelle), accessible en SSH depuis le serveur POM. Il y lance alors l'exécutable Télémac.

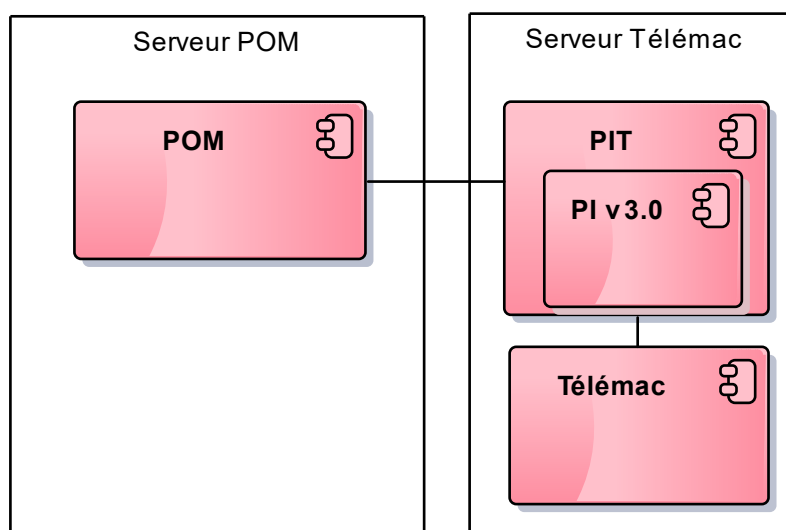


Tableau 2 : Architecture matérielle

Le PIT repose sur la librairie PI (PomInterface) dans sa version 4.0 (au moins).

1.4.2 Architecture logicielle

L'architecture logicielle retenue est celle du Programme d'Interface générique PI également nommé librairie PomInterface. Le PIT consiste en une surcharge de certaines fonctions des processeurs.

1.5 Version des composants

1.5.1 Version PI

La présente spécification logicielle est prévue pour fonctionner sur la version 4 de PI.

1.5.2 Version POM

La présente spécification logicielle est prévue pour s'interfacer avec la version 2.3 du protocole POM.

1.5.3 Version Télémac

La présente spécification logicielle est prévue pour s'interfacer avec Télémac dans sa version v8p4 ou +.

1.5.4 Version Python

Les développements sont réalisés en Python 3.7.

1.6 Interfaces

1.6.1 Généralités

L'exécutable doit s'interfacer avec certains fichiers ou composants comme indiqué ci-dessous.

Le PIT est développé sur la base du Programme d'Interface générique (PI) développé pour réaliser les interfaces entre la POM et les modèles distants.

Le PI gère nativement l'interfaçage avec la POM :

- ✓ Arborescence de fichiers
- ✓ parameters.xml
- ✓ progression.xml
- ✓ ...

Le PI utilise la bibliothèque Libhydro pour la lecture / écriture des fichiers au format XML Sandre.

Note : le PIT n'exploite que des fichiers au format XML Sandre (en entrée et sortie). Les formats XML Image et JSON Lamedo ne sont pas pris en charge et une erreur le signale le cas échéant.

1.6.2 Paramétrage du PIT

Le fichier de paramétrage du **PIT** est au format « .ini » (pit.ini) ou « .toml ». Il reprend les paramètres du PI qu'il complète des paramètres ci-dessous, spécifiques à Télémac, dans la section [pit].

Cette section regroupe les paramètres du PIT.

- ✓ clé « **telemacnodes** » : La liste des correspondances entre les codes station et les numéros de nœuds Télémac. Il s'agit d'une liste de couples « code nœud;code site hydro » séparés par des « | ». Cette clé est obligatoire.
- ✓ clé « **qcomputenodes** » : Les codes des nœuds de calcul Télémac constituant les sections de contrôles pour lesquelles il faut calculer un débit, avec le code site hydro POM attendu. Il s'agit d'une liste de couples « Code nœud debut-Code nœud fin ;code site hydro » séparés par des « | ». Cette clef est facultative (si elle n'est pas renseignée ou vide, les calculs des débits ne sont pas réalisés). Les deux codes nœuds correspondent aux sections de calcul des débits (cf. 3.6.3.2).
- ✓

- ✓ clé « **conditionslimit** » : Le nom du fichier des conditions limites de Télémac, sans le chemin associé car il est défini dans le répertoire du modèle (facultatif, vaut par défaut : 'conditions_limites.liq')
- ✓ clé « **typeconditionslimit** » : Le nom du fichier du type des conditions limites, sans le chemin associé car il est défini dans le répertoire du modèle (facultatif, vaut par défaut : 'type_conditions_limites.cli')
- ✓ clé « **backuptimesteps** » : Le moment de génération des fichiers de reprise (obligatoire, « 0 » par défaut). Il s'agit d'entiers (positifs ou négatifs) séparés par des « | ». Le moment de génération des fichiers de reprise est {TEMPS_BASE} + {PAS_DE_TEMPS}. Cela signifie que les pas de temps positifs se situent après le temps de base (ce qui est autorisé mais à utiliser avec précautions). La valeur « 0 » est obligatoire, si elle n'existe pas, elle est ajoutée.

ATTENTION : cette clef a changé de séparateur et le sens des pas de temps négatifs / positifs.

- ✓ Clé « **liqfilecolumns** » : Les colonnes du fichier « .liq » (pas de valeur par défaut, cette variable doit être renseignée). Il s'agit d'une série de couples « code entité ; code grandeur » séparés par des « | », où le code grandeur vaut H (ou SL pour surface libre) ou Q

ATTENTION : cette clef a changé de séparateurs pour plus de cohérence dans la saisie des paramètres de type « liste ».

- ✓ clé « **supplmodelfiles** » : Les fichiers supplémentaires du modèle. Il s'agit d'une série de noms de fichiers séparés par des « | » (facultatif, vaut par défaut : ""). Ces noms de fichiers sont définis par rapport au dossier du modèle choisi. Cf spécifications PI 3.1 §3.10 pour voir comment définir une clé sur plusieurs lignes.
- ✓ Clé « **commandline_selafin** » : La commande complète 'selafin' (obligatoire, pas de valeur par défaut, cette variable doit être renseignée)

Voici un exemple de valeur pour cette commande :

commandline_selafin = /home/mascaret/telemac-mascaret/v8p2r0/scripts/python3/run_selafin.py

Note : du fait du nouveau mode de lancement de Télémac (mode plus direct, sans le wrapper Python), les paramètres suivants ne sont plus utiles à partir du PIT v2.0 et sont donc supprimés :

- ✓ *Répertoire des entrées : inputdirectory (les entrées sont désormais générées dans le répertoire de calcul Télémac)*
- ✓ *Répertoire résultat : outputdirectory (le fichier résultat porte un nom et une localisation fixe)*
- ✓ *Fichier résultat : resultfile (le fichier résultat porte un nom et une localisation fixe)*
- ✓ *Le nom du fichier de paramétrage Télémac « .cas » : configfile (il est désormais recherché dans le répertoire modèle et doit s'appeler {modele}.cas)*
- ✓
- ✓ *Le fichier de log : logfile (le log est désormais la sortie standard de l'exécution et donc redirigée vers un fichier connu)*
- ✓ *Le fichier des sorties : sortiefile (qui n'est plus généré par Télémac avec le lancement utilisé par le PIT)*
- ✓ *Les noms de fichiers de prise par défaut : backupdefaultfiles (déplacé dans le PI)*
- ✓ *La commande à lancer : commandline (déplacé dans le PI)*

A partir du PIT3.2, les clés suivantes ont été supprimées :

- ✓ Le chemin complet du répertoire de base des fichiers de paramétrage des modèles Télémac : « modelsdirectory » (déplacé dans le PI)
- ✓ Le code du modèle à lancer : « modelsubdirectory » (déplacé dans le PI)

1.6.3 Formats de fichiers

1.6.3.1 LIQ

Le format LIQ (fichiers « .liq ») est un fichier texte, avec séparateur espace, au format suivant :

- ✓ 1 colonne « temps » (nombre de secondes depuis le début de la simulation)
- ✓ N valeurs (H ou Q selon)

Note : l'ordre des valeurs est spécifié dans le fichier de paramétrage du PIT (cf. 1.6.2).

Le fichier commence par trois lignes de commentaire, débutant par un « # », suivies d'une ligne de titre puis d'une ligne d'unités.

Les lignes suivantes comportent les données à chaque date.

Exemple :

```
# Fichier des frontières liquides transitoires
# Généré le 13 nov. 2012 --> à modifier à chaque RUN
# Débit amont Nive (prévisions Mascaret) ; Débit amont Adour
  (prévisions Mascaret) ; Hauteur Lesseps (Adour) (Obs + prévisions
  Mascaret)
T Q(1) Q(2) SL(3)
s m3/s m3/s m
0 0 0 0.4
600 163.97 491.91 -0.008
1200 156.511 548.539 0.087
1800 154.343 727.474 0.227
2400 166.037 849.584 0.336
...
```

1.6.3.2 Selafin (slf)

Le format Sélafin (fichier « .slf ») est un fichier binaire produit par Télémac.

Note : le SCHAPI a fourni des scripts Fortran pour indiquer la manière de décoder ce fichier. Les remarques suivantes sont donc issues de l'analyse de ces fichiers.

Le fichier est constitué comme suit :

- ✓ 80 caractères : titre
- ✓ Deux entiers : nombre de variables, nombre de variables 2
- ✓ Autant de chaînes de 32 caractères que de nombre de variables (nom des variables)
- ✓ Un tableau de 10 entiers (ITAB)
- ✓ Si le dernier item de ITAB vaut 1, la date constituée de 6 entiers
- ✓ 4 entiers

↪ nelem

↪ npoin

↪ ndp

↪ i

- ✓ Un tableau d'entiers, de nelem par ndp éléments (IKLE array)
- ✓ Un tableau d'entiers de npoin éléments (IPOBO array)
- ✓ Un tableau de réels sur 4 octets de npoin éléments (coordonnées X des points)
- ✓ Un second tableau de réels sur 4 octets de npoin éléments (coordonnées Y des points)
- ✓ Les données, avec pour chaque donnée
 - ↪ le temps : réel sur 4 octets
 - ↪ les valeurs : autant de réels sur 4 octets que npoin

Note : l'ordre des valeurs est le même que celui paramétré dans le fichier de paramétrage du PIT, clé « telemacnodes » (cf. 1.6.2).

Le module python « slf » permet de les manipuler.

1.7 Arborescence générale

La machine virtuelle Telemac est organisée comme suit :

- ✓ Répertoires des modèles Telemac (cf. 1.7.1.1) :

```
/home/schapi/pom/modeles/telemac
```

- ✓ Répertoires d'échange POM :

```
/home/schapi/pom/echanges
```

- ✓ Répertoires du PIT (cf. 1.7.2) :

```
/home/schapi/pom/pit
```

1.7.1 Arborescences des fichiers Telemac

Ce chapitre présente l'arborescence des fichiers nécessaires à un lancement de Mascaret.

1.7.1.1 Arborescence d'un modèle Telemac

Le répertoire des modèles contient un sous-répertoire par modèle :

```
/home/schapi
|- pom
  |- modeles
    |- telemac
      |- {CODE_MODELE1}
      |- {MODELE}.cas
      |- conditions_limites.liq
      |- geometrie_ks.slf
      |- type_conditions_limites.cli
      |- sections_controle.txt (éventuellement pour les débits)
```

```

|- {CODE_MODELE2}
  |- {MODELE}.cas
  |- conditions_limites.liq
  |- geometrie_ks.slf
  |- type_conditions_limites.cli
  |- sections_controle.txt (éventuellement pour les débits)
...

```

Chaque répertoire de modèle contient les fichiers et répertoires nécessaires au lancement de Telemac.

Les noms de ces fichiers sont renseignés dans le fichier {MODELE}.cas sous les clefs de paramétrage suivante :

- ✓ FICHIER DES FRONTIERES LIQUIDES : conditions_limites.liq (fichier généré par le PIT et contenant les données d'entrée)
- ✓ FICHIER DES CONDITIONS AUX LIMITES : type_conditions_limites.cli (types de conditions limites, fichier issu du paramétrage du modèle)
- ✓ FICHIER DE GEOMETRIE : geometrie_ks.slf (fichier des géométries, issu du paramétrage du modèle)
- ✓ FICHIER DES SECTIONS DE CONTROLE : sections_controle.txt (fichier des sections de contrôle pour la génération des débits, cf. 3.7.3.2)

1.7.2 Arborescence de travail du PIT

Afin de lancer des calculs indépendants les uns des autres, le PIT s'exécute dans une arborescence qui lui est propre :

```

/home/schapi/pom/pit
|- workspace (répertoire de travail PIT)
  |- archives
    |- {CODE_MODEL_TELEMAC}
    |- {DT_PIVOT}_{DT_ARCHIVE}_{MODE_CALC_POM}_{SCENAR_POM}.pit.slf
  |- {CODE_SESSION_POM}
    |- {DATE_PIVOT}
    |- {MODE_CALCUL_POM}
    |- {CODE_SCENARIO_POM}
      |- {CODE_MODELE_TELEMAC}
        |- analyse
        |- {NOM_SERIE_1}
        |- {NOM_SERIE_2}
        ...

```

Les conventions suivantes sont prises :

- ✓ Les dates sont au format « AAAAMMDD_HHMM » pour effectuer un tri alphabétique chronologique
- ✓ Le mode de calcul POM est lu dans la balise « session → mode » du fichier « parameters.xml ». Il vaut « REAL_TIME » s'il n'est pas renseigné.
- ✓ Le répertoire {CODE_MODELE_TELEMAC} est le nom du répertoire du modèle Télémac à lancer (clef [models]names du fichier « .ini »).

- ✓ Les répertoires « analyse » et « {NOM_SERIE_i} » correspondent aux différents runs à lancer dans Télémac. Ils contiennent les fichiers nécessaires au calcul (cf. 1.9)
- ✓ {DT_ARCHIVE} est la date de référence de l'archivage. Cette date correspond au nom du fichier généré par 'export_date_chop' (cf §3.7.1)

1.7.3 Nomenclatures

Par la suite, les dénominations suivantes sont prises :

- ✓ le répertoire ci-dessous est appelé « **répertoire du modèle Télémac** »

```
/home/schapi/pom/modeles/telemac/{NOM_MODELE}
```

- ✓ Le répertoire ci-dessous est appelé « **répertoire de travail PIT** »

```
/home/schapi/pom/pit/workspace/{CODE_SESSION_POM}/{DATE_PIVOT}/  
{MODE_CALCUL_POM}/{CODE_SCENARIO_POM}/{CODE_MODELE_TELEMAC}
```

- ✓ le répertoire ci-dessous est appelé « **répertoire de travail Télémac** »

```
{répertoire de travail PIT}/analyse/
```

ou

```
{répertoire de travail PIT}/{NOM_SERIE}/
```

1.8 Installation

A partir du PIT 4.0, l'OS d'installation est Linux Debian 11. Il s'agit par défaut d'une VM fournie au format VMDK.

Pour l'installation, le répertoire suivant permet de déposer les fichiers nécessaires au PIT :

```
/home/schapi/uploads
```

Il s'agit des fichiers :

- ✓ d'installation du PI : **pominterface-4.0.X.whl**
- ✓ d'installation du PIT : **pit-4.0.X.whl**

Le PIT est prévu pour être installé de manière standard avec l'outil « pip ».

Exemple d'installation standard :

```
pip install pit-4.0.0-py3-none-any.whl
```

1.9 Lancement de Télémac

Le lancement d'un calcul Télémac a été encapsulé sous différents niveaux d'abstraction pour simplifier l'exploitation de la machine virtuelle par les utilisateurs. Parmi les différentes solutions possibles, nous retenons la solution suivante, qui offre un bon compromis entre souplesse et simplicité d'utilisation.

Le lancement s'effectue en un seul temps avec la commande 'telemac2d.py': c'est elle qui se charge de gérer la compilation (si besoin), puis le calcul (qui génère également les débits si le calcul est paramétré pour) :

```
/home/softs/v8p4r0/scripts/python3/telemac2d.py {MODELE}.cas
```

Les fichiers résultats sont situés à côté du {MODELE}.cas. Le fichier de résultat en hauteurs est au format SLF (cf. 1.6.3.2) nommé « resultats.slf ». Le résultat en débit est un fichier texte (cf. 3.7.3.2) nommé « resultats_debits.txt ».

Le calcul est terminé correctement lorsque la sortie standard de l'exécutable affiche :

```
FIN NORMALE DU PROGRAMME
```

```
DUREE DU CALCUL :
```

```
1 MINUTES
```

```
0 SECONDES
```

```
STOP 0
```

2. Adaptations du PI v3

L'architecture logicielle est basée sur le PI avec surcharge de processeurs ou fonctions du PI.

Les points ci-dessous précisent les fonctionnalités du PI à réutiliser.

2.1 Mode de calcul

Le mode de calcul ne comporte aucun mot clef supplémentaire à ceux gérés dans le PI.

Télémac nécessite une initialisation. Le PIT vérifie donc qu'au moins une méthode de reprise / initialisation est paramétrée.

Note : le mode de calcul ne contient pas le code du modèle Télémac à calculer. Ce code est intégré au fichier « .ini » (cf. 1.6.2). Il suffit de renseigner le chemin du fichier « .ini » dans le scénario POM pour changer le modèle Télémac calculé.

2.2 Nettoyage des fichiers

Le PIT implémente les fonctions de nettoyage des fichiers de Télémac comme suit : nettoyage de tous fichiers du lancement POM en cours après ce lancement (postrun).

```
/home/schapi/pom/pit/workspace/{CODE_SESSION_POM}/{DATE_PIVOT}/
{MODE_CALCUL_POM}/{CODE_SCENARIO_POM}/*
```

2.3 Calculs en parallèle

Le PIT peut s'exécuter plusieurs fois simultanément et est donc paramétré en ce sens.

Le fichier de verrou est créé dans le répertoire courant de travail du PIT afin qu'aucun autre lancement du PIT ne puisse venir le concurrencer.

2.4 Un « run » par scénario

Le PI prévoit de fonctionner différemment selon qu'il doive lancer (au moins) un « run » par scénario POM ou un « run » pour tous les scénarios.

Dans le cas présent, chaque scénario POM donne lieu à au moins un « run ».

2.5 Gestion de l'avancement

L'avancement est géré de manière générique comme suit :

- ✓ 5% : lancement de PIT
- ✓ X% : run
- ✓ 95% : finalisation
- ✓ 100% : fin de PIT

La phase « run » représente donc 90 % : 45 % pour l'analyse, 45 % pour la prévision.

Chaque scénario est représenté à part égale dans ces 45 %. Chaque scénario occupe donc 45/N % (N = nombre de scénarios) et son avancement se décompose comme suit :

- ✓ 20 %: lecture des entrées
- ✓ 60 %: lancements de l'exécutable Télémac
- ✓ 20 % : traitement des sorties

2.6 Gestion des unités

Les valeurs des données dans Télémac sont dans les unités suivantes :

- ✓ en « m » pour les hauteurs
- ✓ « m³/s » pour les débits

Les données étant fournies au format XML sandre, des conversions systématiques sont à prévoir :

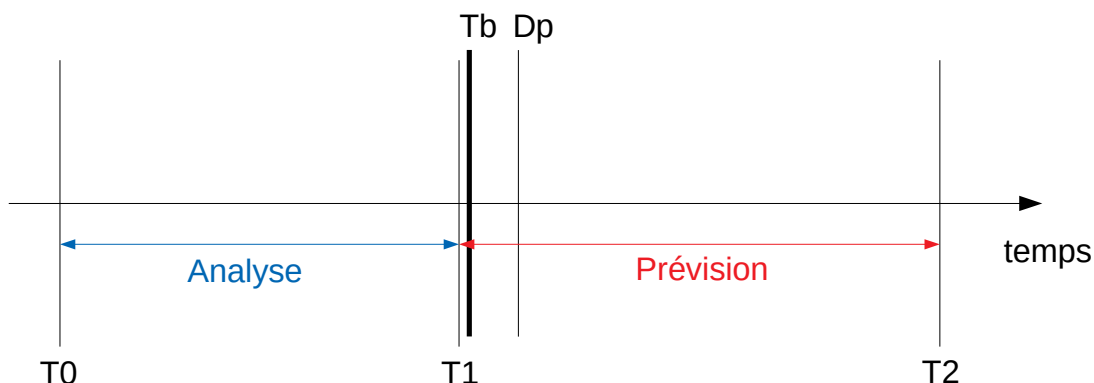
- ✓ en « mm » pour les hauteurs
- ✓ en « l/s » pour les débits

2.7 Dates des simulations analyse/prévision

Le PIT fonctionne désormais en ANALYSE / PREVISION.

Les données sont séparées en 2 paquets :

- Analyse entre T0 et T1
- Prévision entre T1 et T2



Les dates fournies par la POM sont en TU alors que Télémac fonctionne en temps relatif en secondes. Il commence classiquement à 0 en début de simulation (la date de début de la simulation). Ainsi, toutes les dates POM sont reliés au temps Télémac par l'équation ci-dessous :

$$\text{Temps Télémac} = \{\{ \text{DATE_DONNEE} \}\} - \{\{ \text{DATE_DEBUT_SIMULATION} \}\}$$

Les dates absolues sont converties en temps relatif du modèle. Il s'agit d'un nombre de secondes, partant de l'instant {DATE_DEBUT_SIMULATION} (pour lequel le temps relatif

vaut 0). Une fois ce {DATE_DEBUT_SIMULATION} déterminé, toutes les dates absolues peuvent être converties en temps relatif et vice versa.

Pour l'Analyse :

- ✓ {DATE_DEBUT_SIMULATION} = T0
- ✓ {DATE_FIN_SIMULATION} = T1

Pour la Prévision :

- ✓ {DATE_DEBUT_SIMULATION} = T1
- ✓ {DATE_FIN_SIMULATION} = T2

3. Périmètre fonctionnel du PIT

Le présent chapitre décrit les étapes à mener pour le lancement des calculs Télémac. Il détaille les étapes à modifier par rapport au PI.

3.1 Lancement

Le lancement du PIT est identique au lancement du PI (générique) : mêmes arguments de ligne de commande, avec la possibilité facultative de renseigner un fichier « .ini » ou « .toml ».

Note : le nom du fichier « .ini » utilisé par défaut est « pit.ini ».

3.2 Ordonnanceur de tâches

L'ordonnanceur hérite de la classe « ModelBase » et s'appelle « ModelTelemac ».

La méthode principale d'ordonnancement n'est pas modifiée, elle reprend donc les étapes du PI :

- ✓ Initialisation
- ✓ Nettoyage pré run
- ✓ Traitement des scénarios
- ✓ Nettoyage post run

Comme indiqué ci-avant, le PIT est paramétré pour lancer au moins un run par scénario : le lancement éventuel de run après le traitement des scénarios (fonctionnalité utilisée dans le PIG) n'est donc pas effectué.

3.3 Initialisation

L'initialisation du PIT est identique à celle du PI aux points ci-dessous près :

- ✓ la vérification du XML fourni par la POM (parameters.xml) doit être enrichie :
 - ↳ Il ne peut y avoir qu'une ou aucune ressource de type « fichier » en sortie.
 - ↳ Il ne peut y avoir que des grandeurs « H » ou « Q » en entrée (quel que soit le scénario)
- ✓ La lecture du fichier « .ini » doit être adaptée pour lire les paramètres spécifiques au PIT (cf. 1.6.2).
- ✓ Vérification des modes de calcul des scénarios
- ✓ Vérification que le scénario principal donne lieu à un run d'analyse (i.e. contenir des données avant le temps de base)
- ✓ Si la POM attend des débits en sortie, le fichier « .ini » (cf. 1.6.2) doit comporter des nœuds de sections de contrôle pour les entités attendues par la POM.
- ✓ Les méthodes d'initialisation « AUCUNE » et « PERMANENT » ne sont pas autorisées pour Télémac. Une erreur bloquante le signale.

3.4 Traitement des scénarios

Aucun changement par rapport au fonctionnement du PI.

A partir de la version 4.0, le PIT gère plusieurs modèles, comme prévu par le PI (via la section du PI « [models] »).

3.5 Traitement des entrées

La lecture des entrées est réalisée comme dans le PI : toutes les données sont chargées dans une liste commune.

Pour chaque scénario, on lit les données du scénario courant (mais pas du scénario principal).

3.6 Runs

Le lancement des runs de Télémac suit la procédure du PI. Les chapitres ci-dessous précisent les fonctions à surcharger dans les Plx.

Pour rappel, les runs sont liés au scénario POM en cours de traitement.

3.6.1 Types de run

Chaque scénario POM donne lieu à différents types de runs :

- ✓ un run d'analyse sans erreur (plusieurs runs sont envisageables selon les tentatives d'initialisation, mais un seul, le dernier, doit aboutir) si le scénario POM contient des données situées avant le temps de base
- ✓ un ou plusieurs runs de prévision (propagation des séries, ...) si le scénario contient des données après le temps de base.

3.6.2 Copie des fichiers du modèle

On copie les fichiers de paramétrage du modèle décrits en 1.9 : on copie les fichiers en question depuis le répertoire du modèle (cf. 1.7)

```
/home/schapi/pom/modeles/telemac/{NOM_MODELE}
```

vers le répertoire de travail Télémac (cf. 1.7)

```
{répertoire de travail PIT}/analyse/
```

ou

```
{répertoire de travail PIT}/{NOM_SERIE}/
```

selon la nature du run. Il s'agit des fichiers :

- ✓ {MODELE}.cas
- ✓ geometrie_ks.slf
- ✓ type_conditions_limites.cli

Ces 3 fichiers sont obligatoires : si la copie ne peut pas être faite, une erreur est générée.

Le PIT copie aussi des fichiers optionnels, définis par la clé « supplmodelfiles » : si la copie ne peut pas être faite, un warning est généré dans le fichier de log.

Note : le modèle à traiter est lu dans le fichier « .ini » (cf. 1.6.2).

3.6.3 Paramétrage de la plateforme

Le type de lancement de Télémac implémenté dans le PIT nécessite la modification du fichier « .cas » et la génération du fichier « .liq ».

3.6.3.1 Génération du fichier « .liq »

Le fichier doit être généré dans le répertoire de travail du modèle Télémac (cf. 1.7) et doit s'appeler comme indiqué dans le fichier « .ini » (clef conditionslimit, cf. 1.6.2).

Il contient des colonnes dans l'ordre des colonnes paramétrées dans le fichier de paramétrage « .ini » (clef liqfilecolumns, cf. 1.6.2).

Attention : comme indiqué dans les spécifications du PI, les données à faire figurer dans ce fichier dépendent du type de run :

- ✓ classique : ne devrait pas arriver. Ce cas génère une erreur bloquante.
- ✓ analyse : données observées uniquement (i.e. situées avant le temps de base)
- ✓ prévision : cf. 2.2 des spécifications du PI. Ce cas nécessite de filtrer également sur la série en cours de traitement. On tente d'utiliser la valeur de la série à traiter. Si elle n'existe pas ou est vide, on utilise la valeur par défaut (moyenne). Si aucune donnée de la série n'existe, un avertissement est affiché dans le contexte d'exécution.

Pour les types de run ci-dessus :

- ✓ des données sont ajoutées aux extrémités de la plage de calcul afin de combler les lacunes
- ✓ pour la borne inférieure de la plage de calcul, la première valeur existante est dupliquée pour toutes les échéances manquantes avant
- ✓ pour la borne supérieure de la plage de calcul, la dernière valeur existante est dupliquée pour toutes les échéances manquantes après

3.6.3.2 Génération du fichier des sections de contrôle

Le fichier des sections de contrôle est généré si la POM attend des débits en sortie. Ce fichier de paramétrage s'appelle « sections_controle.txt » (cf. 1.9) et est généré dans le répertoire de travail Télémac (cf. 1.7).

Le fichier est organisé comme suit :

```
# Sections de contrôle
{NB_SECTIONS} {INFOS}
{NOM_SECTION_I}
{DEBUT_SECTION_I} {FIN_SECTION_I}
...
```

Où

- ✓ {NB_SECTIONS} est le nombre de sections paramétrées
- ✓ {INFOS} : un entier définissant l'information fournie. Si l'entier est négatif, le paramétrage indique le numéro des nœuds, si l'entier est positif, le paramétrage indique les

coordonnées des extrémités des sections. Dans notre cas, il s'agit de numéros de nœuds (donc 1).

- ✓ {NOM_SECTION_I} est le nom de la ième section (i variant entre 1 et {NB_SECTIONS}). 24 caractères maximum. Dans notre cas, on le prendra égal au couple « nœud-noeud | code POM » du fichier « .ini ».
- ✓ {DEBUT_SECTION_I} (resp. {FIN_SECTION_I}) est le numéro (si {INFOS} est négatif) ou la coordonnée (si {INFOS} est positif) du nœud de début (resp. fin) de la section. Dans notre cas, cela sera les deux numéros de nœuds du fichier « .ini » (clef « telemacnodes », cf. 1.6.2).

Les sections générées dans le fichier sont les sections du fichier « .ini » dont le code site hydro est dans les sorties attendues par la POM.

Par exemple :

```
# Sections de contrôle
5 -1
section_amont
40 1
section_amont_digue
2478 2477
section_encoche_amont
2940 2922
section_parement_aval
3077 3076
section_aval_digue
3568 3567
```

3.6.3.3 Modification du fichier « .cas »

Le fichier « .cas » est généré par copie et modification du fichier « .cas » du répertoire de configuration du modèle Télémac (répertoire « config »). Il est généré dans le répertoire de travail Telemac.

Sur la base de cette copie, certaines valeurs sont mises à jour :

- ✓ Paramétrage de l'initialisation
 - ↳ Cas de la « cote constante »
 - Mettre la variable « SUITE DE CALCUL » à « NON »
 - mettre la variable « CONDITIONS INITIALES » à « 'COTE CONSTANTE' »
 - mettre la variable « COTE INITIALE » à la valeur de la cote associée
 - ↳ Autres cas d'initialisation
 - FICHER DU CALCUL PRECEDENT : pris égal au nom du fichier d'initialisation du issu de la méthode d'initialisation (cf. ci-avant)
 - SUITE DE CALCUL : mis à « OUI »

Note : le paramètre « CONDITIONS INITIALES : 'COTE CONSTANTE' » doit rester dans le fichier de configuration, même s'il n'est pas utilisé.

- ✓ Dates
 - ↳ DATE DE L'ORIGINE DES TEMPS et HEURE DE L'ORIGINE DES TEMPS : pris égaux à {DATE_DEBUT_SIMULATION}
 - ↳ DATE DE L'ORIGINE DES TEMPS : au format « AAAA;MM;JJ »
 - ↳ HEURE DE L'ORIGINE DES TEMPS : au format « HH;MM;SS »

- ↳ DUREE DU CALCUL : pris égal à la durée du calcul exprimé en secondes, calculé par {DATE_FIN_SIMULATION} - {DATE_DEBUT_SIMULATION}
- ✓ Débits : si la POM attend (resp. n'attend pas) des débits en sortie, il faut activer (resp. désactiver) la génération des débits :
 - ↳ la variable « FICHIER DES SECTIONS DE CONTROLE » doit être dé-commentée (resp. commentée) et prise égale à « sections_controle.txt » (cf. 1.9).
 - ↳ si l'on active les débits, le fichier « sections_controle.txt » doit être généré (comme indiqué ci-dessus)

3.6.3.4 Gestion du séparateur des fichiers « .cas »

Le PIT gère le fait que les fichiers « .cas » peuvent avoir 2 caractères différents pour séparer les clés et valeurs, soit « : », soit « = ».

Voici les règles de gestion du séparateur par le PIT :

- ➔ Le séparateur est défini en lisant la première ligne de données (donc la première ligne qui ne commence pas par "/").
- ➔ En lisant cette ligne de gauche à droite, le séparateur est le premier des 2 caractères ":" ou "=" rencontré.
- ➔ La détermination du séparateur doit être indiquée dans les logs ("progression.xml") avec le message : "le séparateur du fichier '<fichier.cas>' est '<separateur>'."
- ➔ Le séparateur est ensuite utilisé pour lire toutes les autres lignes du fichier .cas.

3.6.4 Lancement du run

Seule la commande calcul est à lancer telle qu'indiqué au chapitre 1.9 , dans le dossier « répertoire de travail Télémac ».

3.6.5 Vérification de la bonne exécution du run

Le chapitre 1.9 précise le log attendu pour une exécution correcte.

Si ce log ne présente pas le résultat attendu, les 50 dernières lignes sont ajoutées au fichier de progression sous forme d'un message d'erreur.

3.7 Traitement des sorties

Cette étape vise à produire les fichiers de sortie attendus par la POM à partir des résultats générés par Télémac. Pour chaque run (le run d'analyse, les runs des séries de prévision), deux fichiers de résultat sont produits :

- ✓ un fichier des résultats en hauteur, au format SLF, nommé « resultats.slf » (cf. 1.9)
- ✓ éventuellement un fichier de résultats en débits (nommé « resultats_debits.txt ») au format texte, dont la lecture est indiquée ci-dessous.

Les données sont lues dans les fichiers et intégrées à une liste commune. Puis, pour chaque ressource de sortie, on récupère dans la liste de données les données de la ressource sur la plage de dates indiquées.

Les fichiers de résultats slf correspondent à des « films » du calcul, ils contiennent le déroulement du calcul sur une plage de temps.

On en profite également pour générer le(s) fichier(s) « .slf » de reprise/archivage (cf 3.7.1).

3.7.1 Fichiers de reprises

Cela ne concerne que les runs d'analyse.

Il y a autant de fichiers générés que de pas de temps paramétrés (clef backuptimesteps, cf. 1.6.2). On autorise la génération d'archives avant le temps de base sur des pas de temps négatifs et après le temps de base pour les pas de temps positifs.

Chaque fichier de reprise est construit à partir du fichier de résultats SLF et déposé dans le répertoire d'archive. Le module « SLF » dispose d'une fonction idoine « export_dates_chop ».

Cette méthode « export_date_chop » crée les fichiers de reprise uniquement pour les temps de 'backuptimesteps' qui existent dans le fichier résultat SLF. Ces temps correspondent aux périodes de sortie définies dans le fichier '.cas'.

Le fichier généré au temps T1 (valeur '0' obligatoire du 'backuptimesteps') doit être choisi pour être utilisé comme fichier relay pour le run de prévision correspondant (cf principe générique de PI3).

Ces fichiers de reprise correspondent à des « photos » du calcul à un instant 'T'.

Cette méthode « export_date_chop » utilise la commande 'run_telfile.py' installée dans la VM Telemac Debian11. Le chemin complet de cette commande est défini par l'option « commandline_selaFin » du fichier « .ini ».

3.7.2 Sorties de type « fichier »

Le PIT vérifie que le fichier « parameters.xml » contient au plus une sortie de type « fichier ». Dans le cas contraire une erreur est levée et le calcul s'arrête.

Lors de la génération des sorties attendues par la POM (dans le OutputsProcessor), si une sortie de type « fichier » est attendue, le PIT génère une archive ZIP contenant le(s) fichier(s) « .slf » résultat(s) du (des) run(s) lancés lors du scénario.

Cela comprend les fichiers résultats (« films ») du run d'analyse et de tous les runs de prévision.

Cela ne comprend pas les fichiers de résultats de débits.

Les fichiers seront renommés pour éviter des conflits (ils ont le même nom défini dans le fichier cas, par défaut « resultats.slf ») :

'analyse_{resultats.slf}'

'prev_{serie}_{resultats.slf}'

Note : il n'est pas prévu de traitement sur le fichier SLF intégré à l'archive ZIP (pour éventuellement l'alléger). Il peut être très volumineux et poser d'importants problèmes de performance. En cas de besoin, cela pourra faire l'objet d'une évolution. Dans un premier

temps, on peut envisager l'archivage sur serveur de calcul. Pour cela il suffit a priori de désactiver le ménage.

3.7.3 Sorties XML Sandre

Les fichiers « resultats.slf » (pour les prévisions en hauteur) et « resultats_debits.txt » (pour les prévisions en débit) sont lus et chargés en mémoire par les méthodes `pix_get_hydrodata_analyse()` et `pix_get_hydrodata_prevision()`.

3.7.3.1 Lecture du fichier SLF

Le module python « SLF » permet de lire le fichier et consulter les hauteurs prévues dans le fichier de résultat SLF. Les hauteurs sont en m, et doivent donc être multiplié par 1 000 pour être insérés dans le XML Sandre (cf §2.6).

3.7.3.2 Lecture du fichier texte des débits

Les données de débit produites par Télémac à chacune des sections du fichier des sections de contrôle (cf. 3.6.3.2) sont dans un même fichier « resultats_debits.txt » du répertoire de travail Télémac. Les débits sont en m³/s, et doivent donc être multiplié par 1 000 pour être insérés dans le XML Sandre (cf §2.6).

Ce fichier est formaté comme suit :

```
TITLE = "FLUXES FOR "
  VARIABLES = TIME {NOM_SECTION_1} {NOM_SECTION_2} ...
{NOM_SECTION_I}
{ECHEANCE} {VAL_SECTION_1} {VAL_SECTION_2} ... {VAL_SECTION_I}
```

où

- ✓ la première ligne est une constante
- ✓ la seconde ligne liste, après « VARIABLES = TIME », les noms des sections séparés par un espace dans l'ordre des colonnes des valeurs sur les lignes suivantes.
- ✓ Les lignes suivantes présentent les données de chaque section, par colonnes, en commençant par l'échéance associée aux valeurs. Le nombre d'espaces entre les valeurs est variable.

Par exemple :

```
TITLE = "FLUXES FOR "
  VARIABLES = T section_amont section_amont_digue
section_encoche_amont section_parement_aval section_aval_digue
0.0000000000000000 0.0000000000000000 0.0000000000000000
0.0000000000000000 0.0000000000000000 0.0000000000000000
1.0000000000000000 0.0000000000000000 0.0000000000000000
0.0000000000000000 0.0000000000000000 1.63614546116024E-17
2.0000000000000000 0.0000000000000000 -9.72259436519350E-09
0.0000000000000000 0.0000000000000000 1.89742559330921E-12
```

Les valeurs d'échéances sont en secondes par rapport au temps 0 des dates Télémac (cf. 2.7).

3.8 Finalisation

Cette étape est identique au PI. Il s'agit de la mise à jour du fichier de progression de manière à signifier à la POM la fin de l'exécution du modèle : statut à 0, avancement à 100% (avec éventuellement un message de fin d'exécution).