

*** SCENARIO DE TEST scnlnominal : fait partie des tests vérifiant que le logiciel portageMM2CPP écrit en C++ (leSourceCPP) est conforme au modèle sous sa forme ModelMaker (modele2.mod) ***

```
*****
*   Logiciel portageMM2CPP de portage de code ModelMaker en langage C++   *
*   Copyright INRA, février 2006                                         *
*****
```

*** ** Report de test *** **

```
*****
*
*           PARTIE DESCRIPTION DE TEST
*
*****
```

Description du test :

Informations sur la simulation effectuée dans ModelMaker :

- La simulation effectuée : voir lesEntrees/modele2.mod.
- Les données résultats : voir "lesEntrees".

Informations sur la simulation effectuée en C++ :

- La simulation effectuée : voir mainTestPortageMM2CPP.cpp.
- Les données résultats : voir "lesSorties".

Jeu de test JEU_simulationDeroulement :

Définition/objectif :

Jeu de test vérifiant qu'il n'y a pas eu de problème en cours de déroulement de la simulation C++.

Méthodes/opérations de vérification :

Lecture/contrôle du fichier de traces : traceEcranTests.res.

Jeu de test JEU_variablesGraphique :

Définition/objectif : Jeu de test vérifiant les variables.

Méthodes/opérations de vérification :

Comparaison visuelle de graphiques. Les données à vérifier sont :

- Données ModelMaker de "lesEntrees" :
Le graphique Graph4 de modele2.mod, représentant V1 en fonction du temps.
- Données C++ de "lesSorties" :
Graphique avec en abscisses : t.res, ordonnées : V1.res.
(voir "Note 1.").

Note 1. : ce graphique peut être tracé à l'écran à l'aide de l'outil "affgraphs.py". "affgraphs.py" est un logiciel d'affichage de graphiques (il propose et gère une interface permettant de configurer/paramétrer des représentations graphiques, puis affiche à l'écran les tracés demandés). Il en a été fait une copie "copie_affgraphs.py" dans "lesSorties" afin de l'y utiliser (commande : "python affgraphs.py") ; pour plus d'informations (guide d'utilisation, spécification ...), consulter entêtes et commentaires du fichier "affgraphs.py".

Jeu de test JEU_variablesNumerique :

Définition/objectif : Jeu de test vérifiant les variables.

Méthodes/opérations de vérification :

Comparaison "manuelle" de valeurs numériques. Les données à vérifier sont :

- Données ModelMaker de "lesEntrees" :
Fichier graph4.txt contenant
la liste des données : t, V1 pour t=0,2,4,...,198,200.
- Données C++ de "lesSorties" :
Fichier t.res contenant la liste des valeurs du temps t.
Fichier V1.res contenant la liste des valeurs de la variable V1.

Jeu de test JEU_fluxNumerique :

Définition/objectif : Jeu de test vérifiant les flux.

Méthodes/opérations de vérification :

Comparaison "manuelle" de valeurs numériques. Les données à vérifier sont :

- Données ModelMaker de "lesEntrees" :
Fichier table2.txt contenant
la liste des données : t, F1, F2, F3, pour t=0,2,4,...,198,200.
- Données C++ de "lesSorties" :
Fichier t.res contenant la liste des valeurs du temps t
Fichiers F1.res, F2.res, F3.res contenant
les listes des valeurs des flux F1,F2,F3.

Jeu de test JEU_compartimentsGraphique :

Définition/objectif : Jeu de test vérifiant les compartiments.

Méthodes/opérations de vérification :

Comparaison visuelle de graphiques. Les données à vérifier sont :

- Données ModelMaker de "lesEntrees" :
Le graphique Graph6 de modele2.mod représentant
C1_source, C2_reservoir, C3_reservoir, C3max, C4_puits, C5_debordement
en fonction du temps.
- Données C++ de "lesSorties" :
Graphiques avec en abscisses : t.res, et en ordonnées : C1_source.res,
C2_reservoir.res, C3_reservoir.res, C4_puits.res, C5_debordement.res.
(voir "Note 1." plus haut).

Jeu de test JEU_compartimentsNumerique :

Définition/objectif : Jeu de test vérifiant les compartiments.

Méthodes/opérations de vérification :

Comparaison "manuelle" de valeurs numériques. Les données à vérifier sont :

- Données ModelMaker de "lesEntrees" :
Fichier graph6.txt contenant la liste des données :
t, C1_source, C2_reservoir, C3_reservoir, (C3max), C4_puits, C5_debordement
pour t=0,2,4,...,198,200.
- Données C++ de "lesSorties" :

Fichier t.res contenant la liste des valeurs du temps t.
 Fichiers C1_source.res, C2_reservoir.res, C3_reservoir.res, C4_puits.res,
 C5_debordement.res contenant les listes des valeurs des compartiments
 C1_source ... C5_debordement.

 *
 * PARTIE RAPPORT DE TEST *
 *

 *
 * RAPPORT/CONCLUSIONS DU TEST EFFECTUE LE 07/02/06 :
 *
 *
 *

***** Jeu de test JEU_simulationDeroulement : *****

A la lecture du fichier "traceEcranTests.res", il n'est pas trouvé d'anomalie dans les traces :

La simulation est bien effectuée dans les conditions qui ont été demandées (de t = 0 à 200 s, avec sortie de résultats toutes les 2 s).

Les valeurs lues pour C3max, P1, P2, P3, P4, P5 sont correctes.

Les "resultats sortis" pour t, V1, F1, F2, F3, F4, C1_source, C2_reservoir, C3_reservoir, C4_puits, C5_debordement, ne sont pas vérifiés dans ce jeu de test; ils le sont par ailleurs (dans les autres jeux de test).

Conclusion du JEU_simulationDeroulement : TEST_OK.

***** Jeu de test JEU_variablesGraphique : *****

Pour V1 :
 On observe/compare les courbes MM (tracées sur support papier par impression de Graph4) et les courbes C++ (tracées à l'écran avec "copie_affgraphs.py").

Pour V1 :
 Il est vérifié que les courbes MM et C++ ont la même forme, présentent les mêmes variations pour - "à vue d'oeil" - les mêmes valeurs.

Conclusion du JEU_variablesGraphique : TEST_OK.

***** Jeu de test JEU_variablesNumerique : *****

On calcule comme écarts max (absolus) entre valeurs C++ ET MM :

Pour V1 : $\text{MAX}(| \text{ECART} (\text{donnée C++}, \text{donnée MM}) |) = 0,0000111000$

On considère que ces écarts sont liés à la précision des calculs.

Conclusion du JEU_variablesNumerique : TEST_OK.

***** Jeu de test JEU_fluxNumerique : *****

On calcule comme écarts max (absolus) entre valeurs C++ ET MM :

Pour F1 : $\text{MAX}(| \text{ECART} (\text{donnée C++}, \text{donnée MM}) |) = 0,0000300000$
 Pour F2 : $\text{MAX}(| \text{ECART} (\text{donnée C++}, \text{donnée MM}) |) = 0,0001725000$
 Pour F3 : $\text{MAX}(| \text{ECART} (\text{donnée C++}, \text{donnée MM}) |) = 0,0000586300$

On considère que ces écarts sont liés à la précision des calculs.

Conclusion du JEU_fluxNumerique : TEST_OK.

***** Jeu de test JEU_compartimentsGraphique : *****

Pour C1_source, C2_reservoir, C3_reservoir, C4_puits, C5_debordement :
On observe/compare les courbes MM (tracées sur support papier par impression de Graph6) et les courbes C++ (tracées à l'écran avec "copie_affgraphs.py").

Pour C1_source, C2_reservoir, C3_reservoir, C4_puits, C5_debordement :
Il est vérifié que les courbes MM et C++ ont la même forme, présentent les mêmes variations pour - "à vue d'oeil" - les mêmes valeurs.

Conclusion du JEU_compartimentsGraphique : TEST_OK.

***** Jeu de test JEU_compartimentsNumerique : *****

On calcule comme écarts max (absolus) entre valeurs C++ ET MM :

- Pour C1_source :
MAX(| ECART (donnée C++, donnée MM) |) = 0,0012200000
- Pour C2_reservoir :
MAX(| ECART (donnée C++, donnée MM) |) = 0,0016930000
- Pour C3_reservoir :
MAX(| ECART (donnée C++, donnée MM) |) = 0,0014840000
- Pour C4_puits :
MAX(| ECART (donnée C++, donnée MM) |) = 0,0022890000
- Pour C5_debordement :
MAX(| ECART (donnée C++, donnée MM) |) = 0,0028550000

On considère que ces écarts sont liés à la précision des calculs.

Conclusion du JEU_compartimentsNumerique : TEST_OK.

***** CONCLUSION BILAN du test effectué le 07/02/06 : *****

- * TEST_OK,
- * Tous les jeux de test ont été déroulés, aucune anomalie n'a été relevée.

* RAPPORT/CONCLUSIONS DU TEST EFFECTUE LE xx/xx/06 :
*

etc

etc
