TRUE

TEMPORAL REASONING UNIVERSAL ELABORATION

True System dynamics software

MANUEL Partie 01

Concept

Dernière mise à jour: 2015/03/26

www.true-world.com

Table des matières

I - PRESENTATION

A) Description

TRUE = TEMPORAL REASONING UNIVERSAL ELABORATION

Le logiciel TRUE permet de modéliser, simuler, optimiser, mémoriser et restituer l'évolution de systèmes pluridisciplinaires calculée selon des données, des instructions et des raisonnements par :

- une modélisation mathématique transparente et implicite :
  - équations différentielles couplées d'ordre n^n
- une modélisation de type dynamique des systèmes en temps discret et en temps continu (Euler)
- une optimisation dynamique grâce à des fonctions de rétro-calcul

Il possède une interface graphique wysiwyg qui permet de créer et de restituer dynamiquement des modèles en 2D, 3D et 4D via un modeleur 3D.

B) Principe général

L'évolution d'un système peut être considérée comme une succession d'états restitués à intervalles de temps réguliers (discrétisation), dont les différences, puisque toujours expliquées, peuvent être contrôlées.

La modélisation consiste à définir des règles d'évolution, pour des intervalles de temps définis sur une période donnée, puis à calculer et à restituer cette évolution.
Les règles sont décrites dans des actions planifiées individuellement par sept paramètres temporels.
Ces paramètres définissent la temporalité du modèle :

- **Rate** : le paramètre **Rate** n'agit que si le flux relie deux stocks.
  - Si le paramètre Rate est activé, les stocks seront actualisés par le résultat de l'action (résultat = y/TimeStepI) à la fin de l'unité de temps en cours, avant le passage à l'unité de temps suivante (comme dans Vensim)
  - Si le paramètre **Rate** est désactivé, les stocks seront actualisés par le résultat de l'action (résultat = y) juste après que toutes les actions de même chronologies aient été calculées.

- **chronologie**, **départ**, **répétition**, **intervalle**, **type de cycle** et **cycles filtrés**

Il est possible de définir individuellement pour chaque action un **type de cycle** :

- **cycle par défaut** (cycle de base)

- **cycle flottant**, pouvant être filtré sur le **cycle par défaut**

- **cycle flottant**, pouvant être filtré sur lui-même

Le modèle est calculé pour le nombre de **cycles par defaut** demandés.

A chaque unité de temps, les actions, filtrées et classées selon leurs paramètres temporels, calculent et retournent des valeurs.
Si le flux qui contient les actions relie deux stocks, les valeurs de retour seront transférées entre le stock source et le stock cible, selon le paramètre **Rate**.

La somme des valeurs de tous les stocks du modèle est constante pour toutes les unités de temps.

Les valeurs sont mémorisées dans la base de données du modèle pour chaque unité de temps.

II - PRINCIPE DETAILLE

A) Principe du logiciel Vensim

Le logiciel **Vensim** est un logiciel de dynamique des systèmes réputé.

1. Phases de calcul

Seulement pour t=0

- **Phase 1**
  - Les valeurs initiales des stocks sont calculées à partir de leur équation
  - Les valeurs des stocks sont mémorisées pour les flux(t=0), graphes et les tables
- **Fin de la phase 1**

- **Phase 2**
  - Calcul des flux dans un ordre défini automatiquement par les causalités.
  - Les valeurs des flux sont mémorisées pour les autres flux(t=0), graphes et les tables
- **Fin de la phase 2**

- **Phase 3**
  Les stocks source et cible des flux (**valve**) sont mis à jour par les les valeurs de flux (valve) multipliées par le TIME STEP
  Stock source= stock source- (value * TIME STEP)
  stock cible = stock cible + (value * TIME STEP)
- **Fin de la phase 3**

Pour t>0

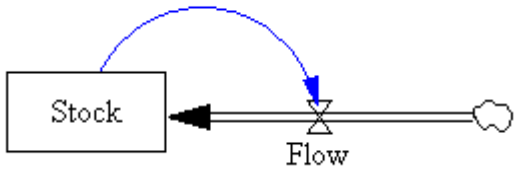
Pour chaque unité de temps

- **Phase 1**
  - Les stocks sont initialisés par leurs valeurs à t-1: **stock(t)=stock(t-1)**
  - Les valeurs des stocks sont mémorisées pour les flux(t), graphes et les tables
- **Fin de la phase 1**

- **Phase 2**
  - Calcul des flux dans un ordre défini automatiquement par les causalités.
  - Les valeurs des flux sont mémorisées pour les autres flux(t), graphes et les tables
- **Fin de la phase 2**

- **Phase 3**
  Les stocks source et cible des flux (**valve**) sont mis à jour par les les valeurs de flux (**valve**) multipliées par le **TIME STEP**
  Stock source= stock source- (value * TIME STEP)
  stock cible = stock cible + (value * TIME STEP)
- **Fin de la phase 3**

Exemple Vensim




```
Valeur initiale du Stock = 10
Equation du Flow : Flow = Stock + 1
Equation du Stock : Stock += Flow
```

Phases pour les unités de temps de 0 à n

	Phase 1	Phase2	Table of results
time	**Stocks calculation**	**Flows calculation**	
0	Stock(t0) = Initial value or calculation	Flow(t0) = Stock(t0) + 1	Flow(t0) & Stock(t0)
1	Stock(t1) = Stock(t0) + Flow(t0)	Flow(t1) = Stock(t1) + 1	Flow(t1) & Stock(t1)
2	Stock(t2) = Stock(t1) + Flow(t1)	Flow(t2) = Stock(t2) + 1	Flow(t2) & Stock(t2)
3	Stock(t3) = Stock(t2) + Flow(t2)	Flow(t3) = Stock(t3) + 1	Flow(t3) & Stock(t3)
n	Stock(tn) = Stock(tn) + Flow(tn)	Flow(tn) = Stock(tn) + 1	Flow(tn) & Stock(tn)

Table des résultats

Time	Stock	Flow
0	10	11
1	21	22
2	43	44
3	87	88
4	175	176
5	351	352
6	703	704
7	1407	1408
8	2815	2816
9	5631	5632
10	11263	11264

2. Fonctionnement en mode continu

Pour obtenir des valeurs intermédiaires et une meilleur précision, le logiciel peut fonctionner en mode continu.
Pour cela, on augmente le nombre d'unités de temps et on diminue les valeurs affectant les stocks.

Le paramètre **TIME STEP** définit l'augmentation du nombre d'unité de temps :

- 1 = 1 (mode discret, le nombre d'unité de temps n'est pas modifié)
- 0,5 = 1/2 (mode continu, nombre d'unité de temps est multiplié par 2)
- 0,25 = 1/4 (mode continu, nombre d'unité de temps est multiplié par 4)
- 0,125 = 1/8 (mode continu, nombre d'unité de temps est multiplié par 8)
- 0,0625 = 1/16 (mode continu, nombre d'unité de temps est multiplié par 16)
- 0,03125 = 1/32 (mode continu, nombre d'unité de temps est multiplié par 32)
- 0,015625= 1/64 (mode continu, nombre d'unité de temps est multiplié par 64)
- 0,0078125 = 1/128 (mode continu, nombre d'unité de temps est multiplié par 128)

Les nouvelles valeurs affectant les stocks sont soit multipliées par le TIMESTEP, c'est la méthode Euler simplifiée, ou calculées par une autre méthode comme la méthode Runge-Kutta

Exemple : 20 unités de temps en temps (années)

- mode discret : **TIME STEP**=1, nombre d'unités de temps = 20 (années)
- mode continu : **TIME STEP**=0,25, nombre d'unités de temps = (20*4)=80 (trimestres)

Phase 1 de l'exemple précédent, en mode continu

	Phase 1
time	**Stocks calculation**
0	Stock(t0) = Initial value or calculation
1	Stock(t1) = Stock(t0) + Flow(t0) * TimeStep
2	Stock(t2) = Stock(t1) + Flow(t1) * TimeStep
3	Stock(t3) = Stock(t2) + Flow(t2) * TimeStep
n	Stock(tn) = Stock(tn) + Flow(tn) * TimeStep

B) Principe du logiciel True

1. Phases de calcul

Dans le logiciel True :

- les actions contenues dans les flux sont planifiées à l'aide de 7 paramètres temporels
- plusieurs actions peuvent êtres contenues dans un flux

voir aussi www.true-world.com/htm/en/algorithm.php

Pour t = 0 uniquement

- **Phase 1**
  Les stocks sont initialisés avec leur valeur initiales
  Stock(0) = Stock(valeur initiale)

  Les stocks miroirs sont initialisés avec la somme de leur stocks source
  Stock miroir(0) = somme(stocks sources(0))

  Les flux sont égales = 0
  Flux(0) = 0

  Les valeurs des stocks, stocks miroirs et des flux sont mémorisées dans la base de données pour les graphes et les tables.

  Ces valeurs représentent l'état du système avant le démarrage du calcul du modèle.

- **Fin de la phase 1**

Fin de pour t = 0 uniquement

Pour l'unité de temps t=1 au nombre maximum d'unité de temps

- **Phase 1**
  Les stocks sont initialisés avec leurs valeurs à t-1: **stocks(t) = stocks(t-1)**
  Les stocks miroir sont initialisés avec leurs valeurs à t-1: **stocks miroirs(t) = stocks miroir(t-1)**
  Les valeurs des flux sont initialisées à 0: **flux(t) = 0**
- **Fin de la phase 1**

- **Phase 2**
  Les actions qui doivent être exécutées sont:
  - filtrées par leurs paramètres temporels: départ, intervalle, répétition, type de cycle et cycle filtré
  - ajoutées à des lots d'actions qui contiendont des actions ayant le même paramètre **Chronologie**.
  - les lots d'actions sont triés par chronologie croissante.
- **Fin de la phase 2**

- **Phase 3, pour chaque lot = 1 au nombre maximum de lot d'actions**

  - **Phase 3.1 pour chaque action**
    Calcul de l'action
    Mémorisation de la **valeur de retour** dans un **buffer**
  - **Fin de la phase 3.1**

  - **Phase 3.2, pour chaque action**
    Si le flux qui contient l'action relie deux stocks:
    - si le paramètre **Rate** est activé:
    -- le stock source est mis à jour: **stock source(t) = stock source(t) - valeur de retour**
    -- le stock cible est mis à jour: **stock cible(t) = stock cible(t) + valeur de retour**
    -- les valeurs des stocks sont mémorisées dans la base de données

    - si le paramètre **Rate** n'est pas activé:
    -- la **valeur de retour** est mémorisé dans un **RateBuffer**

    Les valeurs des flux sont mises à jour: **flux(t) = flux(t) + valeur de retour**
    Llux sont mémorisées dans la base données
  - **Fin de la phase 3.2**

  - **Phase 3.3, pour chaque stock miroir**
    Le stock miroir est mis à jour:
    - **stock miroir(t)= somme(stock sources(t))**
  - **Fin de la phase 3.3**

- **Fin de la phase 3, pour chaque lot**

- **Phase 4, pour chaque RateBuffer**

  Les stocks sources et cibles des actions qui ont un RateBuffer sont mis à jour!
  stock source(t) = stock source(t) - RateBuffer
  stock cible(t) = stock cible(t) + RateBuffer

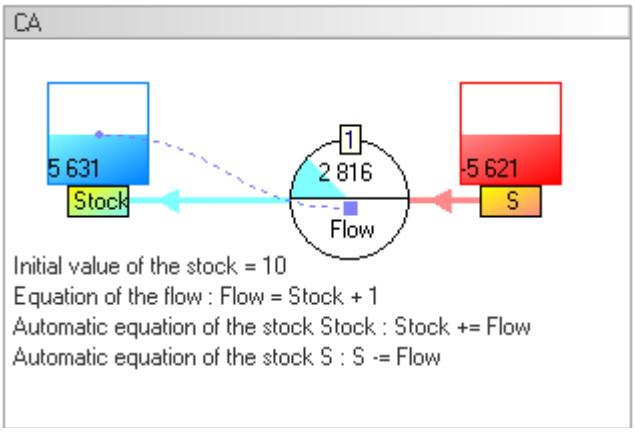
  Note: **TimeStepI = 1 / TIME STEP**

  Les valeurs des stocks sont mémorisées dans la base de données
- **Fin de la phase 4**

- **Phase 5, pour chaque stock mirroir**
  Les valeurs des stock miroirs sont mis à jour avec les valeurs de leurs stocks sources:
  - **stock miroir(t)= somme(stock sources(t))**
  Les valeurs des stocks miroirs sont mémorisées dans la base de données
- **Fin de la phase 5**

Fin de Pour chaque unité de temps

Exemple True en temps discret



Phases pour les unités de temps de 0 à n

	Phase 1	Phase3	Phase4	Table of results
time	**Stocks calculation**	**Flows calculation**	**Stocks calculation**	
0	Stock(t0) = Initial value			0 & Stock(0)
1	Stock(t1) = Stock(t0)	Flow(t1) = Stock(t1) + 1	Stock(t1) += Flow(t1)	Flow(1) & Stock(1)
2	Stock(t2) = Stock(t1)	Flow(t2) = Stock(t2) + 1	Stock(t2) += Flow(t2)	Flow(2) & Stock(2)
3	Stock(t3) = Stock(t2)	Flow(t3) = Stock(t31) + 1	Stock(t3) += Flow(t3)	Flow(3) & Stock(3)
n	Stock(tn) = Stock(tn-1)	Flow(tn) = Stock(tn) + 1	Stock(tn) += Flow(tn)	Flow(n) & Stock(n)

Table des résultats

Time_	Stock	Flow
0	10	0
1	21	11
2	43	22
3	87	44
4	175	88
5	351	176
6	703	352
7	1407	704
8	2815	1408
9	5631	2816
10	11263	5632

2. Fonctionnement en mode continu

Le logiciel TRUE peut reproduire le fonctionnement en mode continu du logiciel Vensim avec la méthode Euler simplifiée.

Note : True utilise le paramètre **TimeStepI** (inverse du paramètre **TIME STEP** dans Vensim)

Une fois le **TimeStepI** initialisé, l'augmentation des unités de temps et la division des valeurs affectant les stocks n'étant pas automatiques, il est nécessaire de :

- Modifier la temporalité du modèle afin d'augmenter le nombre d'unité de temps, en modifiant le paramètre *Répétition* des actions concernées

- Pour modifier la valeur des actions en mode continu, activer le paramètre **Rate** de l'action concernée des flux qui relient des stocks

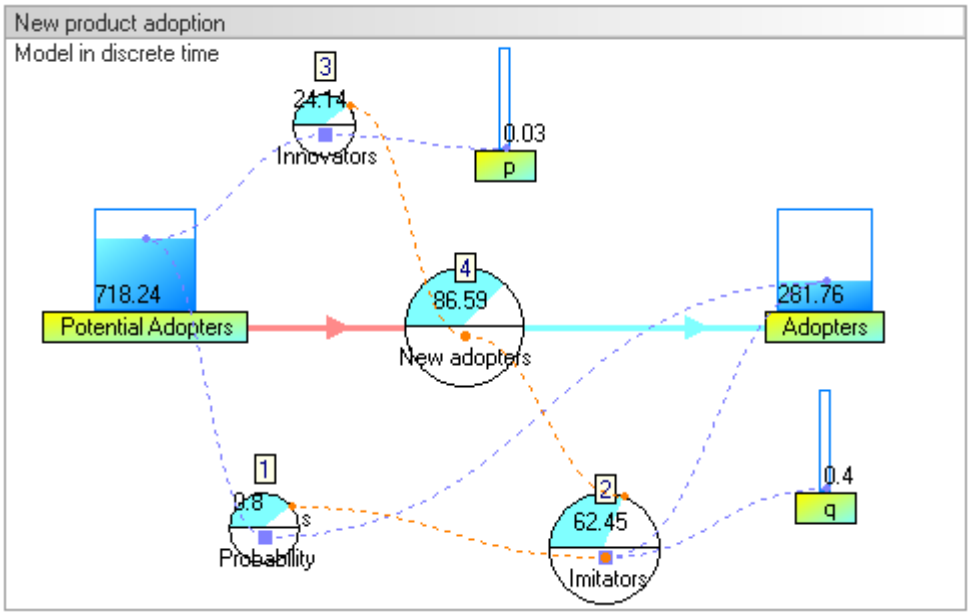
➢ Remarque :
  - Le mode continu n'étant pas automatique, il est possible de construire des modèles qui fonctionnent à la fois en mode continu et en mode discret.

Phases de calculs de l'exemple en mode continu

	Phase 1	Phase3	Phase4
time	**Stocks calculation**	**Flows calculation**	**Stocks calculation**
0	Stock(t0) = Initial value		
1	Stock(t1) = Stock(t0)	Flow(t1) = Stock(t1) + 1	Stock(t1) += Flow(t1) / TimeStepI
2	Stock(t2) = Stock(t1)	Flow(t2) = Stock(t2) + 1	Stock(t2) += Flow(t2) / TimeStepI
3	Stock(t3) = Stock(t2)	Flow(t3) = Stock(t31) + 1	Stock(t3) += Flow(t3) / TimeStepI
n	Stock(tn) = Stock(tn-1)	Flow(tn) = Stock(tn) + 1	Stock(tn) += Flow(tn) / TimeStepI

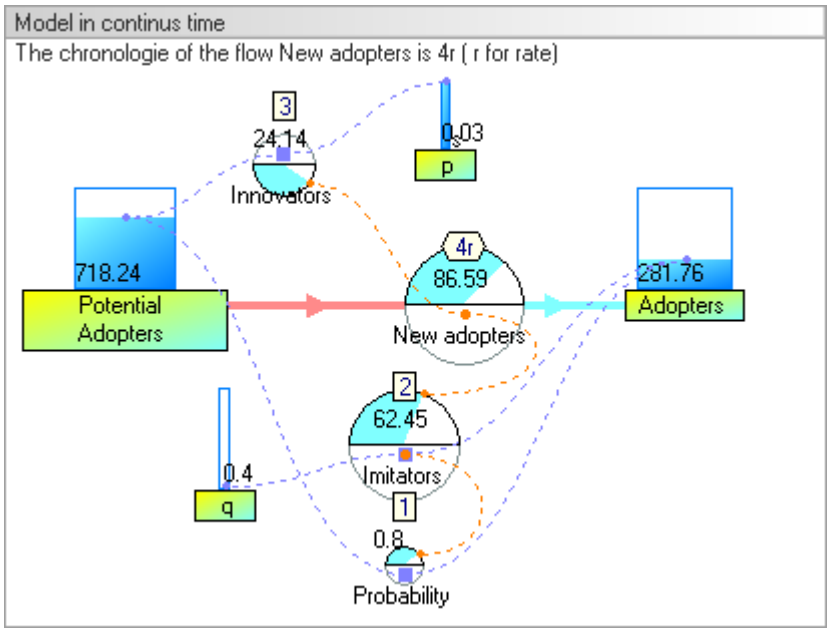
Transformation d'un modèle en mode discret en mode continue

Le modèle en mode discret



- La temporalité de ce modèle est de 15 années, soit un cycle de 15 unités de temps

Le modèle en mode continu



On souhaite transformer ce modèle discret en mode continu avec un **TimeStepI** = 12

Ajustement de la temporalité :

- La temporalité de ce modèle sera de 15 années de 12 mois
- On initialise donc les paramètres *Répétition* des actions à 12
- On compilera le modèle pour 15 cycles

Ajustement de la valeur du flux qui relie les deux stocks source et cible

- On active le paramètre **Rate** de l'action du flux **New adopters**

III – TRUE / VENSIM DIFFERENCES - IMPORTATION DE MODELE

voir http://www.true-world.com/htm/en/import.php

IV – CONSEILS

voir http://www.true-world.com/htm/fr/advice.php

V - WORKING WITH TRUE FUNCTIONS

voir http://www.true-world.com/htm/en/workingwith.php

VI – EXERCICES

voir http://www.true-world.com/htm/en/exercises.php

voir le manuel **Man60-Exercises.pdf**