

La dynamique des systèmes

Principes, applications et introduction à
la programmation dynamique

Séminaire présenté par Simon Riguelle, Bio-ingénieur.

*Attaché, Service public de Wallonie, Direction du Milieu forestier.
Doctorant, ULg - Gembloux Agro-Bio Tech.*

Dans le cadre du cours :

FORE0015-1 (GF211) ~ Exploitations forestières (Prof. P. Lejeune)

Plan de l'exposé

- Introduction aux systèmes et modèles
- La dynamique des systèmes (DS)
- Applications de la DS au domaine environnemental
- Éléments de programmation dynamique :
 - Visual Basic pour Application
 - STELLA
 - MATLAB SIMULINK
 - (VENSIM)

Introduction aux systèmes et modèles

Systemes et modèles

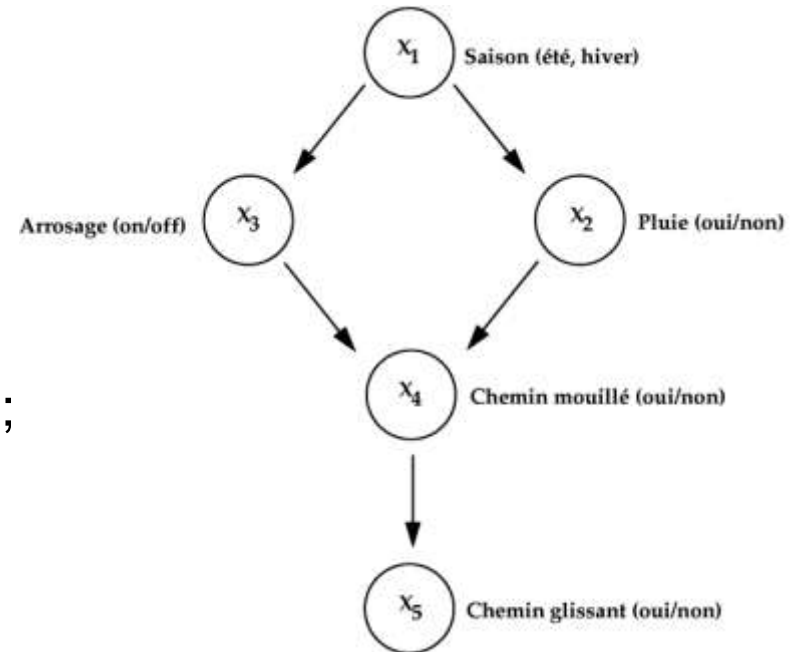
- Un modèle est une représentation **simplifiée** d'un système élaborée en vue de certains objectifs :
 - prévision
 - compréhension
 - manipulation
 - **aide à la décision**
- Il existe de nombreuses formes de représentation :
 - **mathématique**
 - graphique (une carte routière, un diagramme)
 - analogique (une maquette)
 - linguistique

Caractéristiques des modèles

- Statique vs dynamique :
 - Un modèle est **dynamique** si les variables comportent une dimension **temporelle**.
 - Le **temps** (t) est un attribut d'une variable ou une variable à part entière.
- Optimisation vs simulation :
 - L'optimisation recherche la solution à un problème borné sous certaines contraintes ;
 - La simulation est la résolution numérique d'un système d'équations ;
 - La simulation vise à reproduire le comportement du système, à voir de quelle manière il est susceptible de réagir, puis à utiliser le modèle pour tester des **scénarios** (l'approche « what... if ? »).

Différents types de modèles

- Macro-économétriques ;
- Multi-agents ;
- Réseaux bayésiens (probabilistes) ;
- Modèles d'optimisation ;
- Modèles de **dynamique des systèmes (DS)** :
 - approche « inventée » par Jay Forrester, du MIT (USA)
 - Approche transdisciplinaire héritière de la cybernétique



La dynamique des systèmes

Pourquoi la DS ?

- Dès que l'on s'intéresse à l'évolution d'un système complexe ;
- Pas pour objet de déterminer des situations optimales, mais pour répondre à des questions du type « Que se passerait-il si ? » ;
- Tester des hypothèses, mettre en œuvre des scénarios
- Dans le cas des chablis, évaluer des politiques de gestion de crise grâce aux scénarios simulés.

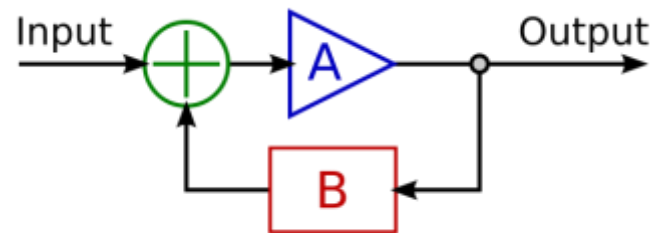
Principes de la DS (1)

- **Deux aspects :**
 - quantitatif (éléments stocks – flux)
 - qualitatif (boucles de rétroaction)
- Aspect **quantitatif :**
 - Les stocks (variables d'état, \neq réserves) sont appelés niveaux.
 - Les variables de flux sont appelés taux.
 - Les variables auxiliaires (constante, coefficient, fonction)
 - Fait l'objet d'une traduction mathématique sous forme d'équations

$$S_x(t) = S_x(t - \delta t) + \frac{\delta F_x}{\delta t} - \frac{\delta F_y}{\delta t}$$

Principes de la DS (2)

- Aspect **qualitatif** :
 - Boucles de rétroaction (*feedback loops*) :
 - Positives : phénomènes de croissance (ou décroissance) cumulative car elles se renforcent mutuellement.
 - Négatives : conduisent à des comportements de stabilisation, d'amortissement car elles se compensent les unes les autres.



- Temporalité :
 - Délai : temps qui s'écoule entre une décision et son implémentation
 - Retard : décalage entre l'état du système réel et l'information que nous en avons
- Relations non-linéaires (non-proportionnalité, effet de seuil...)

Les étapes de la méthode

- 1) Définition de la problématique
- 2) Analyse causale, modèle verbal
- 3) Formalisation du modèle
- 4) Simulation du modèle
- 5) Analyse des résultats

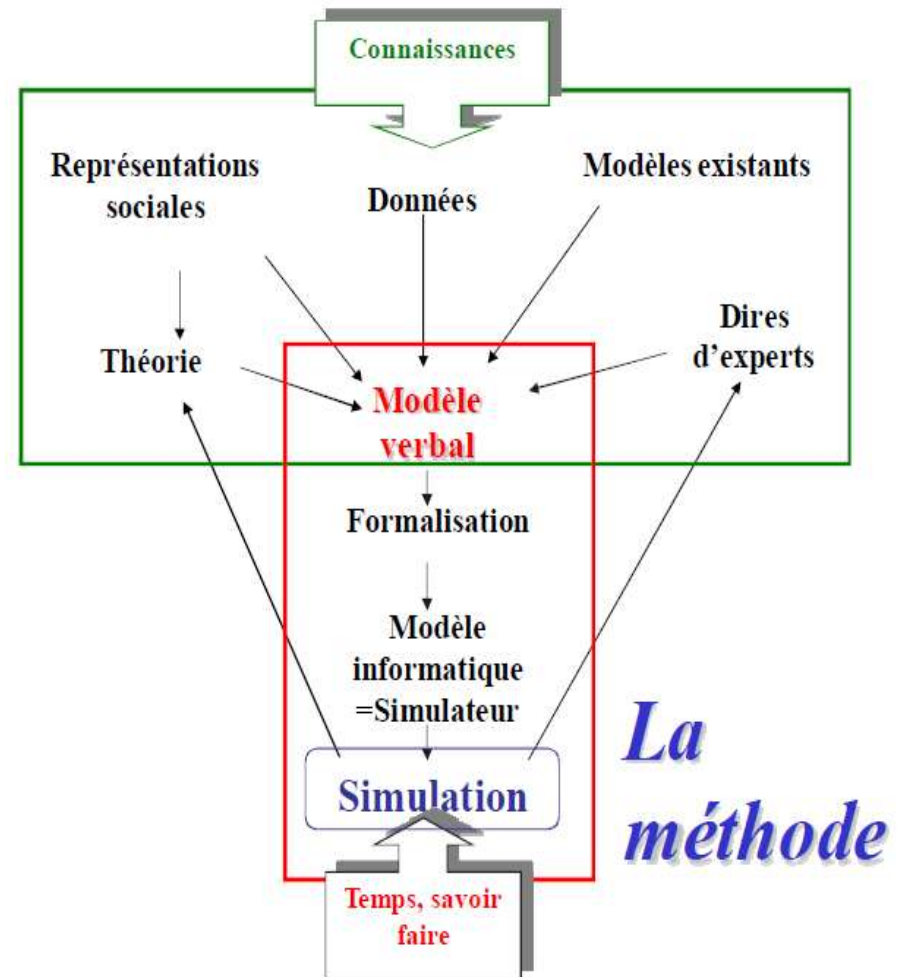
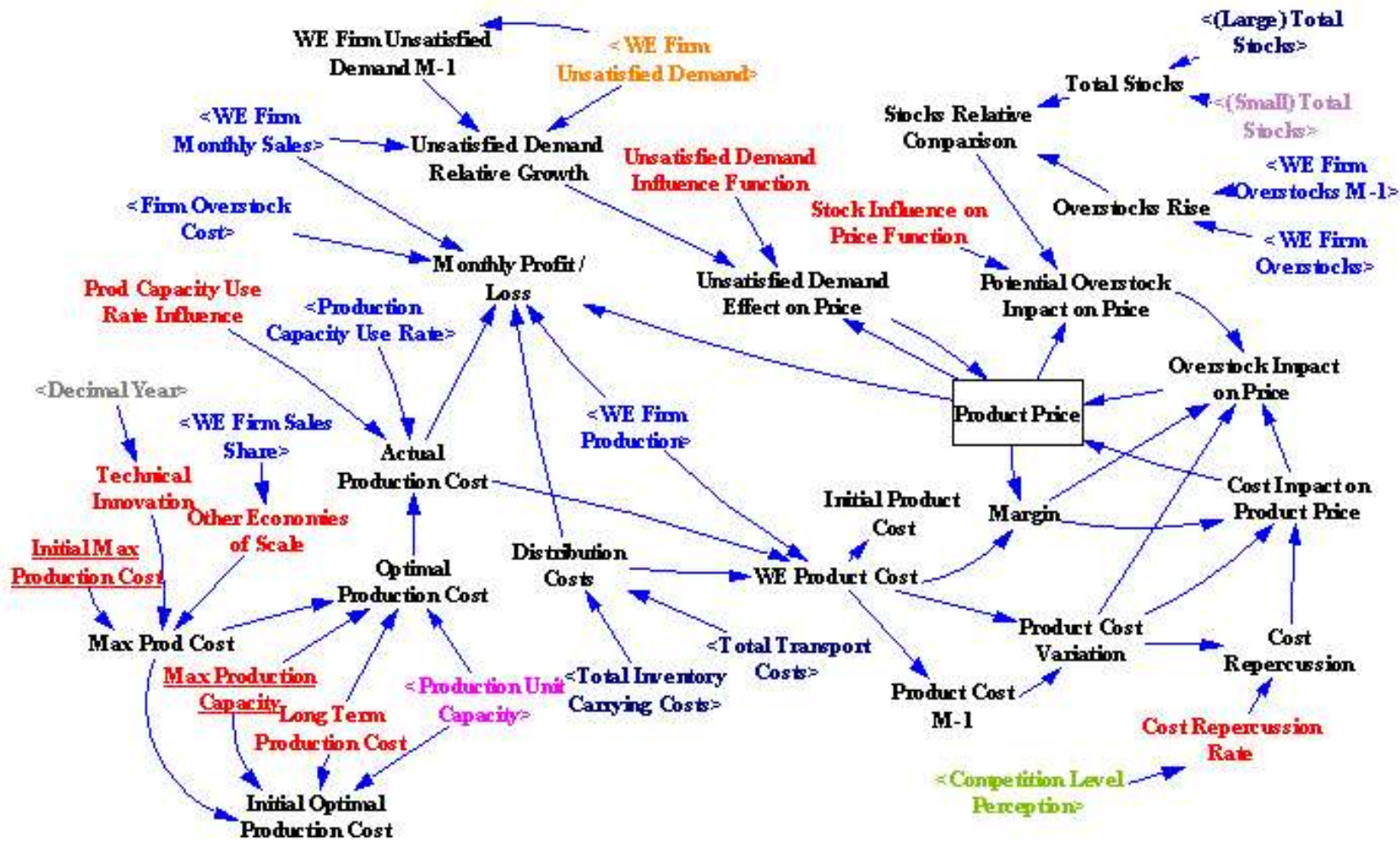


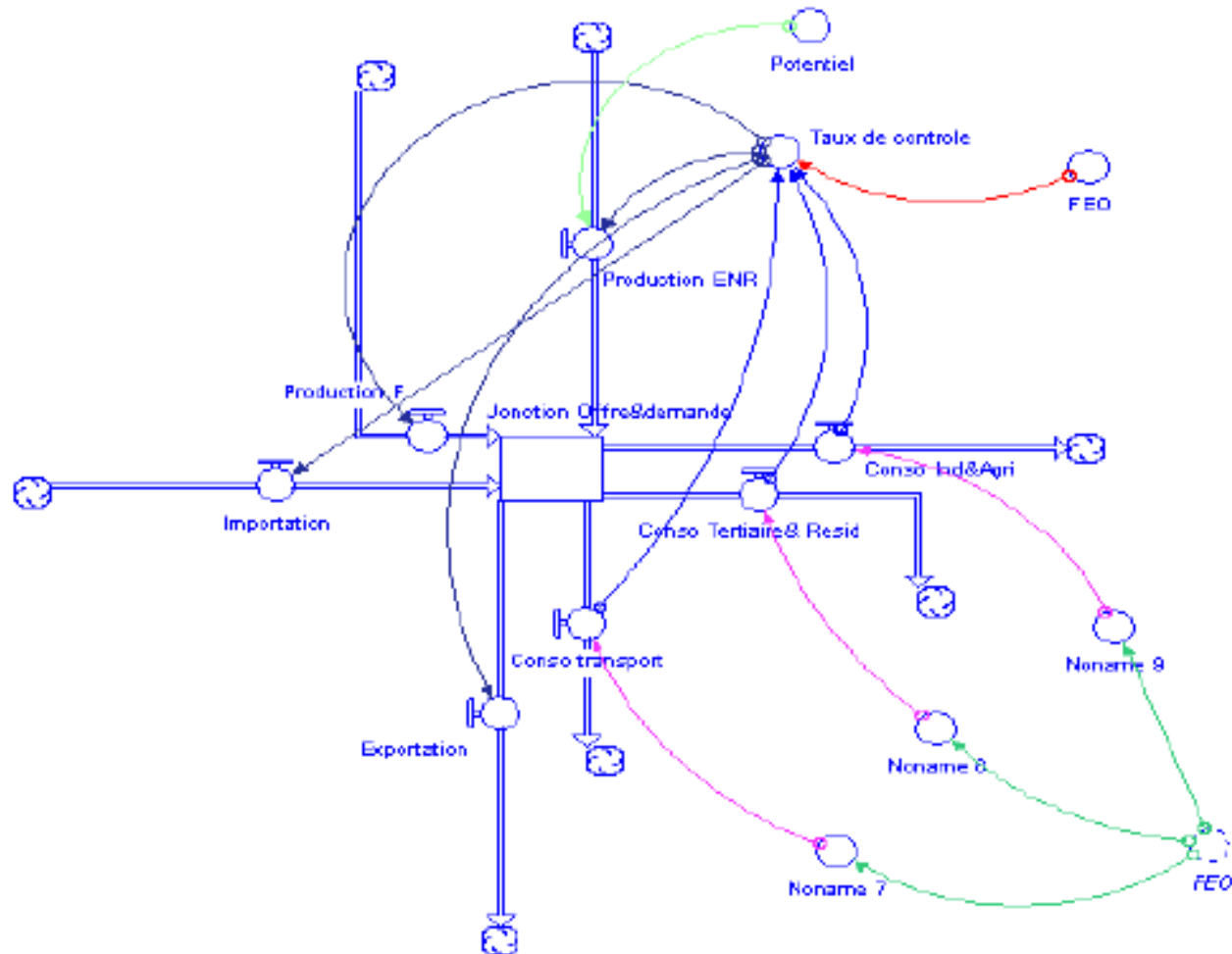
Diagramme d'influence

- Schéma du système, des stocks, des flux et boucles de rétroaction :



Programmation

- Implémentation sous Stella :



Atouts (+)

- **Extrême flexibilité** : tout système dynamique peut être représenté et simulé aisément grâce à la présence sur le marché de logiciels conviviaux et relativement peu chers :
 - Excel
 - Stella, Vensim
 - Matlab
- La formalisation en termes de stock et flux, l'analyse des rétroactions positives et négatives et les concepts de stabilité, cycles limites, oscillations, etc. constituent une **boîte à outils transdisciplinaire** qui permet à des spécialistes de disciplines différentes de collaborer aisément autour d'un même projet. (*Integrated Assessment*).

Faiblesses (-)

- **Extrême flexibilité** : de nombreuses erreurs sont commises dans le champ de la DS, où trop souvent les modélisateurs se lancent dans la construction de modèles complexes sans disposer des connaissances disciplinaires minimales pour construire des représentations scientifiquement acceptables des différents systèmes modélisés.
- **Extrêmement gourmand en données** :
 - il n'existe pas de méthode réellement satisfaisante pour estimer de façon rigoureuse les très nombreux paramètres qui caractérisent la plupart des modèles de DS.
 - La veille des données est primordiale !

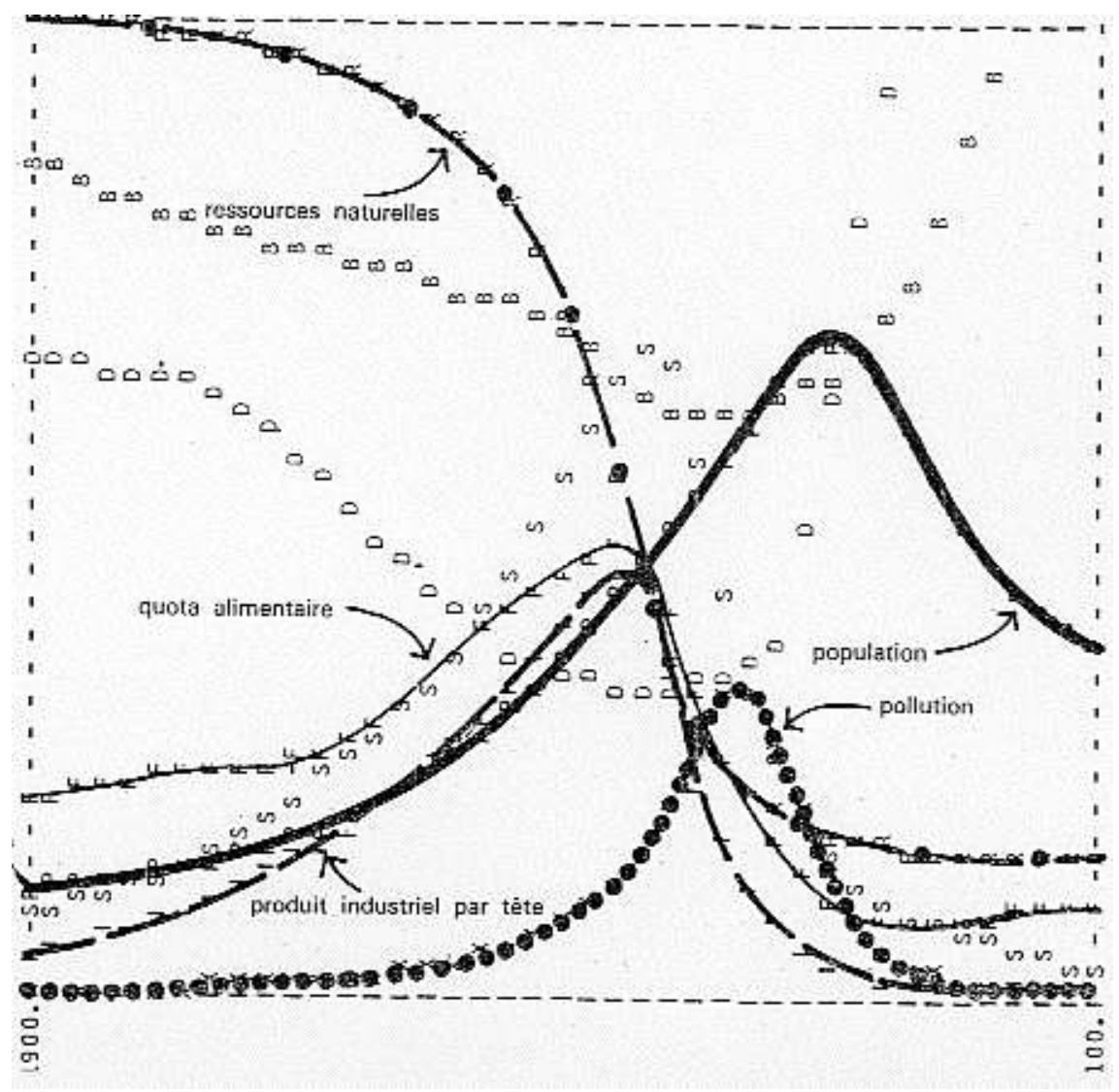
Les modèles de ce type peuvent être extrêmement sensibles aux valeurs des paramètres et des stocks à l'initialisation

Applications au domaine environnemental

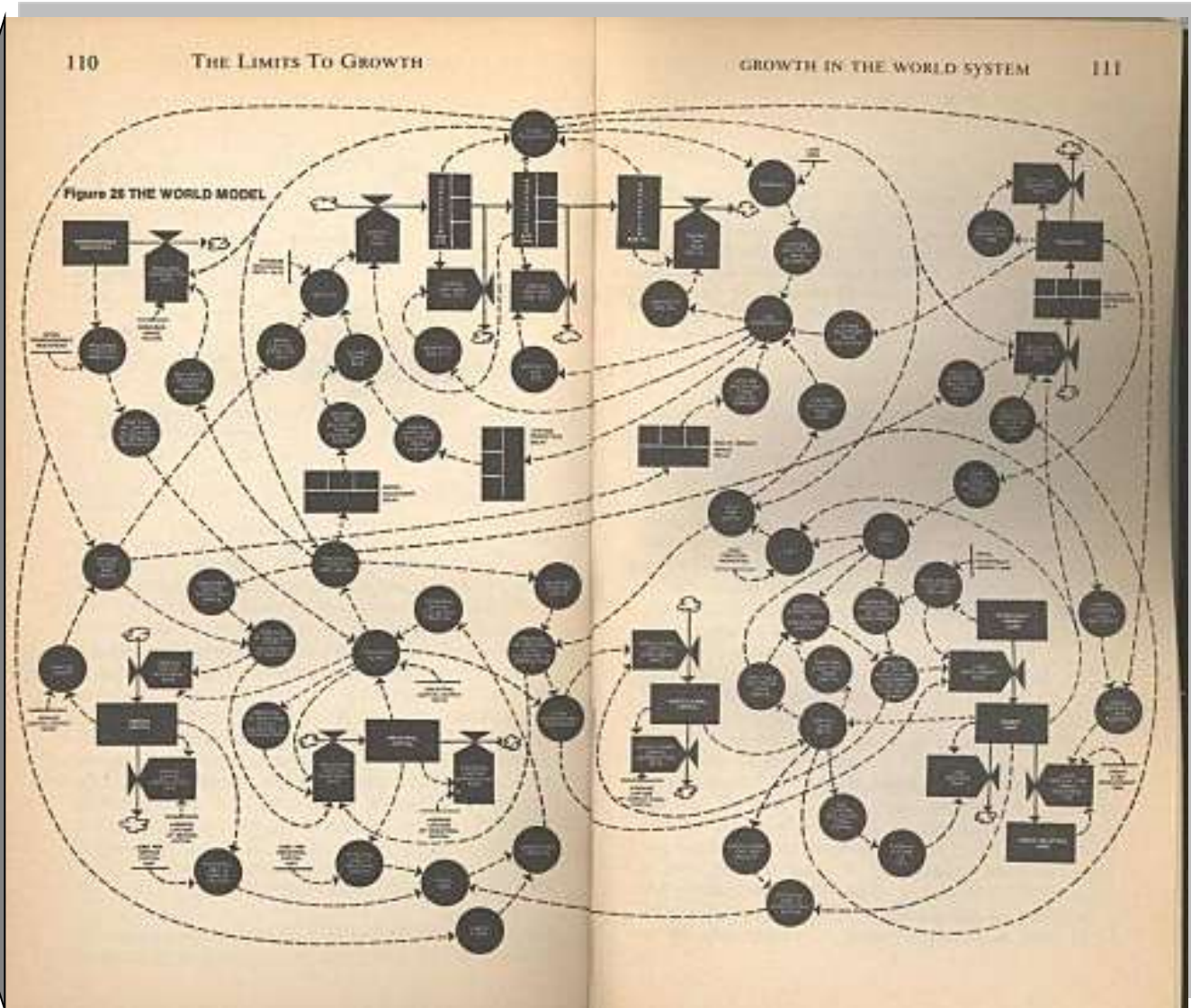
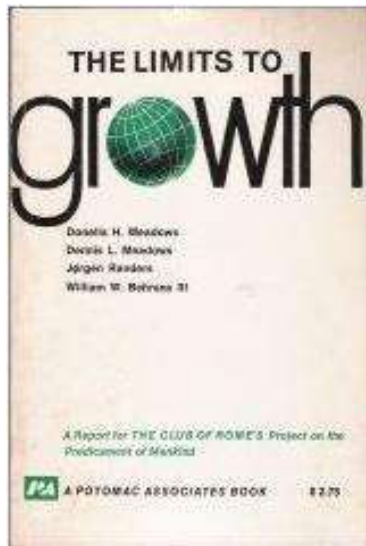
La DS à l'origine du développement durable...

- La méthode a été appliquée à l'exploration des limites environnementales de la croissance démographique et industrielle mondiale (Forrester, 1971).
- C'est ce dernier modèle qui a été amplifié par ses élèves Dennis et Donella Meadows (entre autres) pour le célèbre rapport au Club de Rome «The limits to growth » (Meadows et al., 1972).
- Les modèles de DS ont joué un rôle majeur dans le processus qui a conduit à la notion de **développement durable** et constituent toujours un instrument privilégié pour l'aborder.

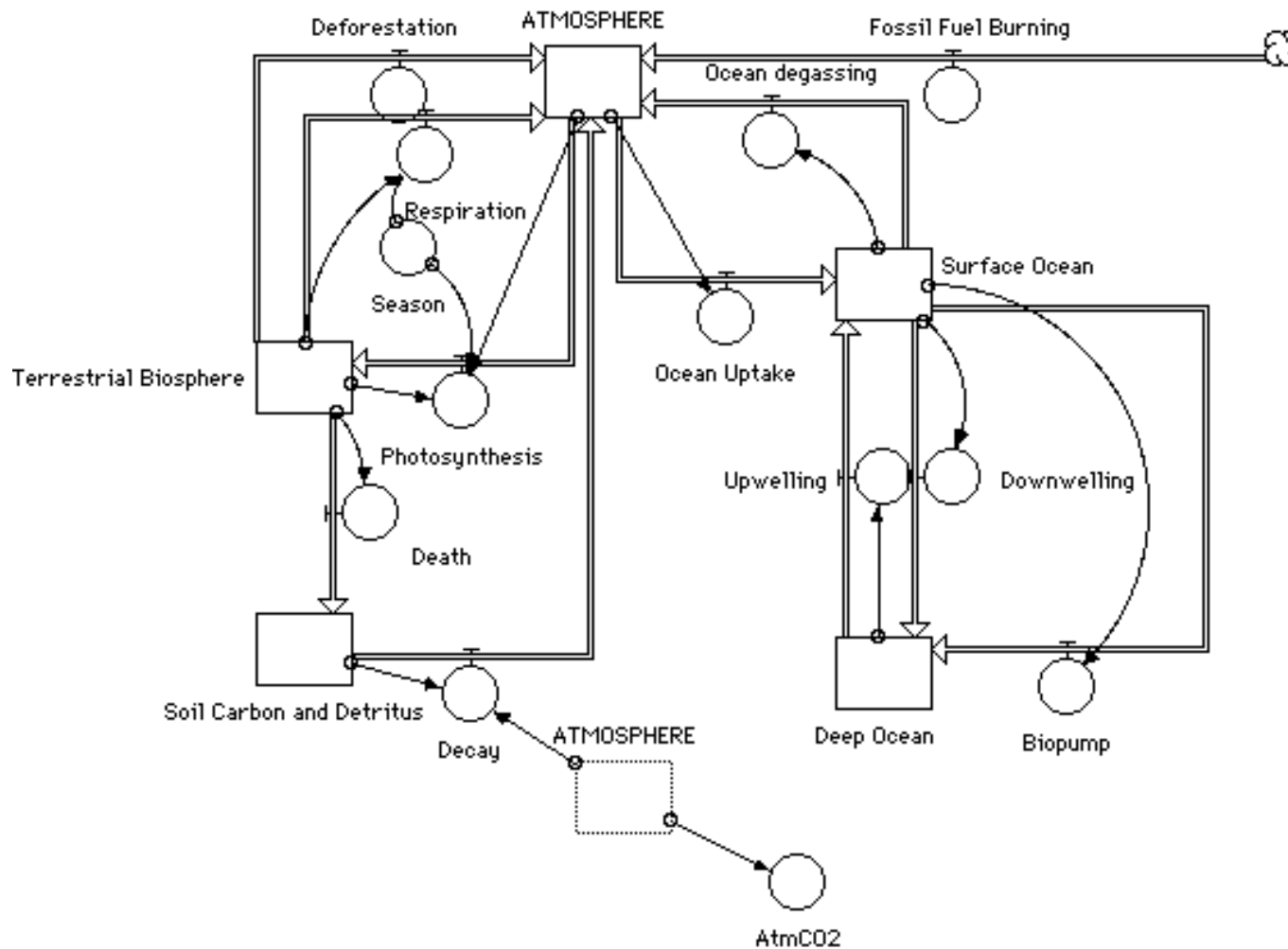
« The limits to growth » (Meadows et al., 1972) (1)



« The limits to growth » (Meadows et al., 1972) (2)

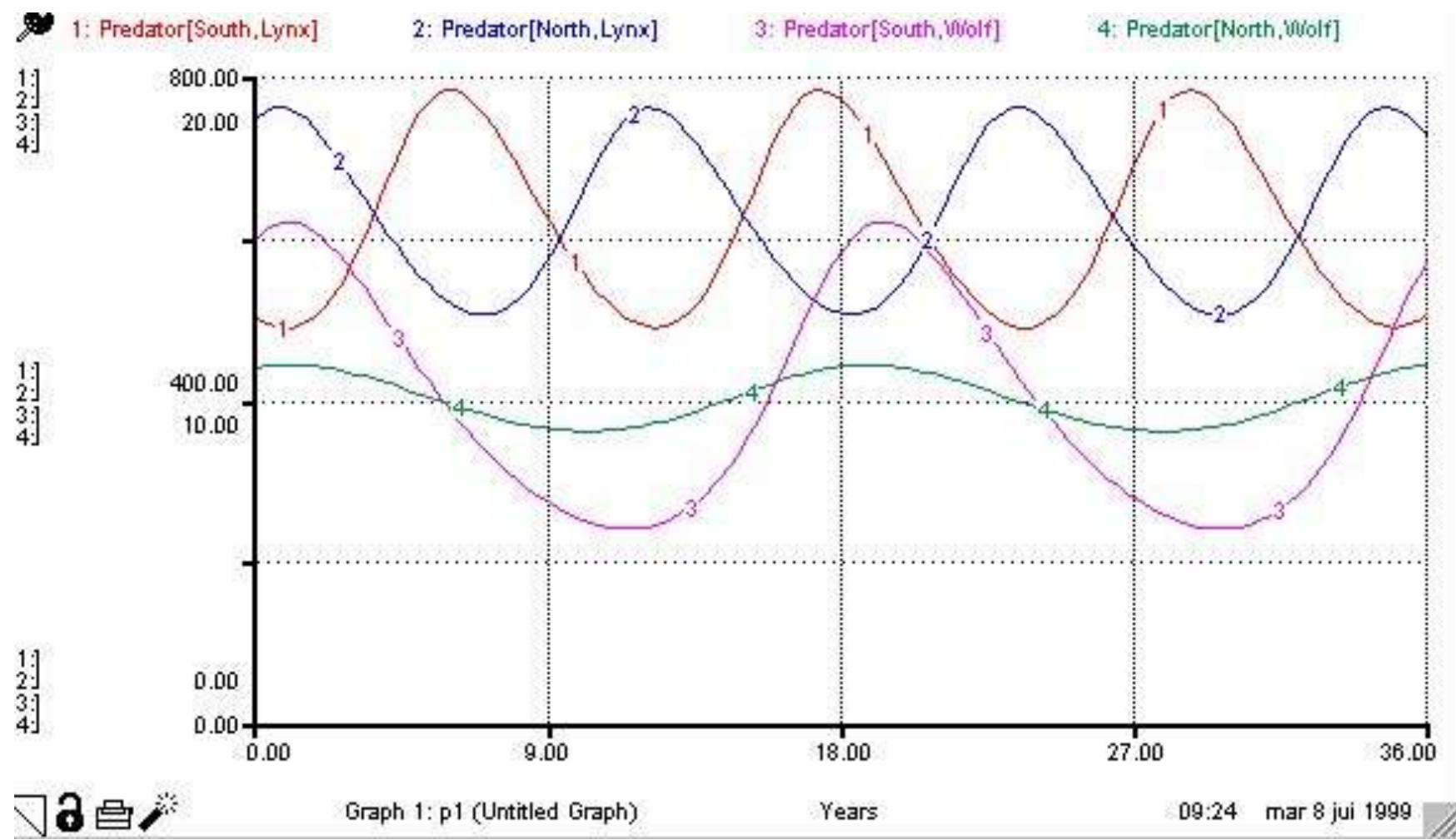


Cycle du carbone



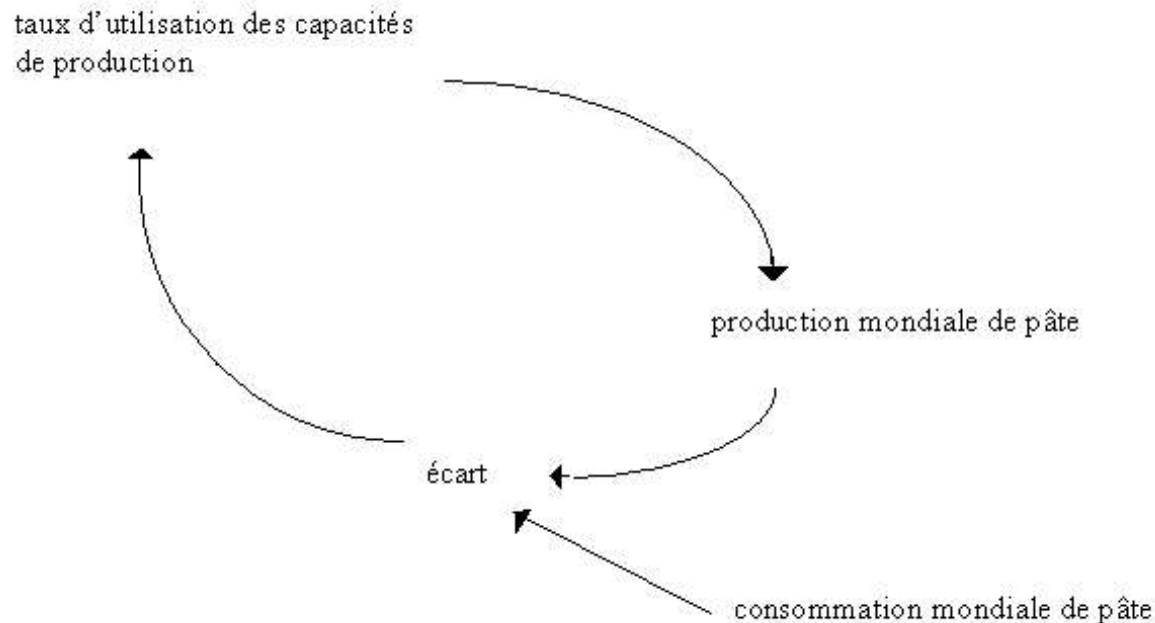
Source: Web page at Cornell University, Department of Geology

Modèle proies-prédateurs



Marché mondial de la pâte à papier (1)

- Exemple de boucle de rétroaction régulatrice :

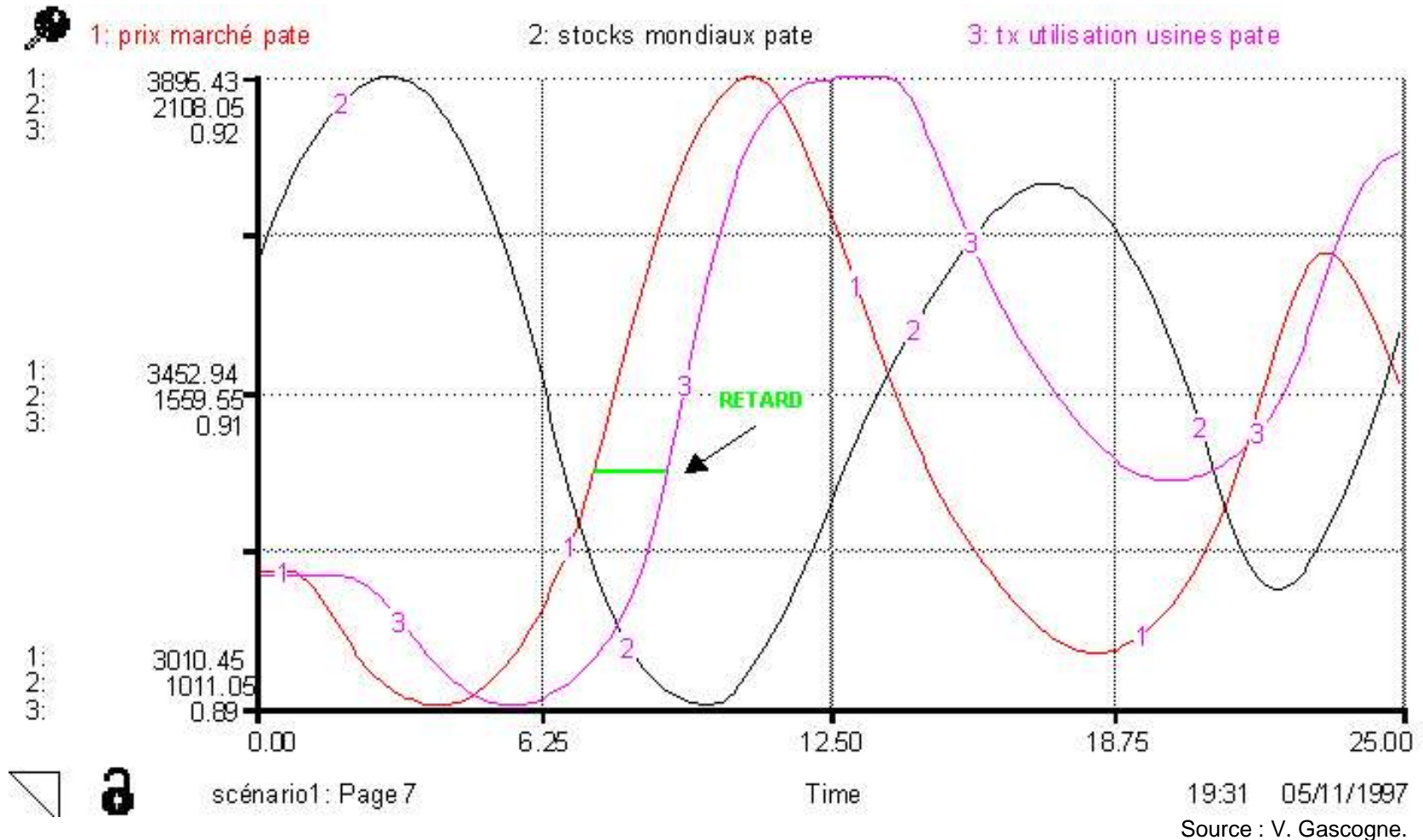


- Cette boucle s'insère dans un modèle comprenant plusieurs centaines de variables !

Source : V. Gascogne.

Marché mondial de la pâte à papier (2)

- Représentation graphique :



Notions de programmation dynamique

Programmation dynamique

- La programmation des systèmes dynamiques se déroule en plusieurs étapes :
 - Description du système à modéliser, des variables d'état, de flux,...
 - Choix d'un outil informatique de modélisation
 - Implémentation du système dans le logiciel et programmation du code
 - Création d'une interface utilisateur
- Quelques outils logiciels et langages de programmation utilisés :
 - Visual Basic pour Application (MS Excel)
 - STELLA[®]
 - MATLAB[®] Simulink[®]
 - (VENSIM[®])

Visual Basic pour Applications (VBA)

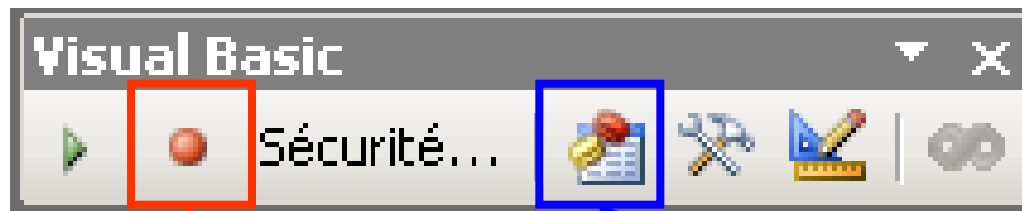
- VBA est le langage qui est utilisé pour programmer en Visual Basic les applications Microsoft Office (Excel, Access, Word).
- Les programmes VBA sont attachés à un document spécifique de l'application hôte : classeur Excel, document Word, DB Access... et ne peuvent fonctionner qu'avec cette application hôte !
- Une bonne connaissance des applications est nécessaire pour tirer profit de la puissance de ces logiciels à l'aide de VBA (simplification des commandes, amélioration des fonctions existantes ou ajout de nouvelles fonctions au logiciel).
- L'écriture de programmes VBA, appelés **procédures** ou **macros**, est en outre la solution la plus économique pour doper le logiciel.

Pourquoi VBA dans Excel ?

- Tirer profit de la puissance du tableur en y rajoutant du code pour automatiser les tâches de calcul nécessaires à la simulation.
- Possibilité de créer et d'éditer des graphiques présentant les résultats des simulations.
- Possibilité de créer des interfaces conviviales et didactiques.
- Logiciel très répandu et *quasi* gratuit.
- Possibilité de communiquer entre deux applications hôtes (Excel – Access).

Programmer en VBA

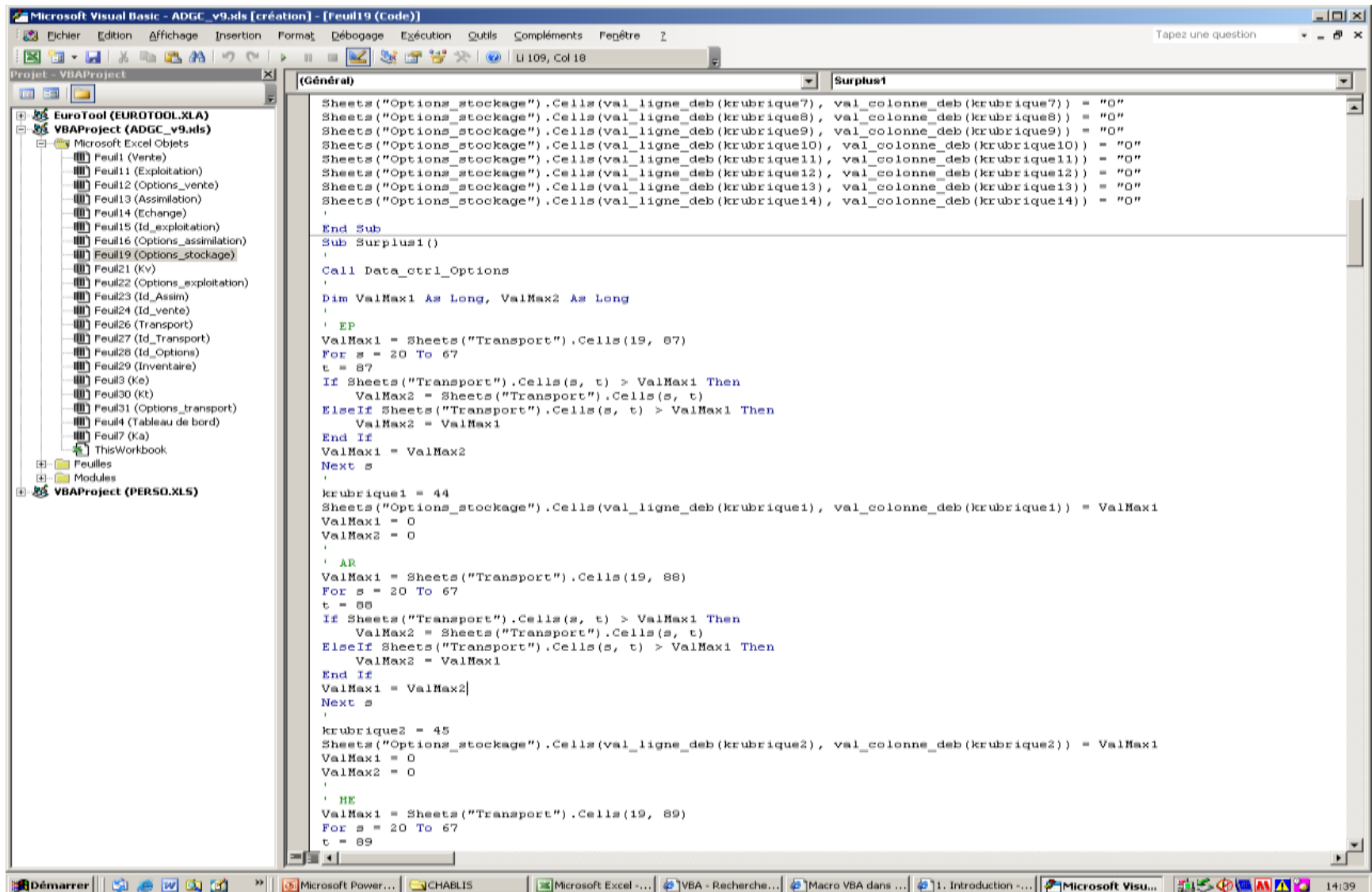
- Deux outils de création de commandes personnalisées :
 - **L'enregistreur de macros** qui mémorise toutes les commandes effectuées dans l'application hôte et les transforme sous forme d'instructions écrites en langage de programmation VBA. Il offre la possibilité de se familiariser avec les procédures de création de commandes personnalisées. On peut ensuite modifier ce code ainsi produit dans Visual Basic Editor.
 - **Visual Basic Editor** qui permet également de créer des commandes personnalisées à la fois souples et performantes contenant des instructions VBA. On a accès à toutes les possibilités de l'application hôte.



Enregistreur de macros

VB Editor

Visual Basic Editor



Programmer VBA dans Excel

```

Sub Surplus1()
'
Call Data_ctrl_Options
'
Dim ValMax1 As Long, ValMax2 As Long
'
' EP
ValMax1 = Sheets("Transport").Cells(19, 87)
For s = 20 To 67
t = 87
If Sheets("Transport").Cells(s, t) > ValMax1 Then
    ValMax2 = Sheets("Transport").Cells(s, t)
ElseIf Sheets("Transport").Cells(s, t) > ValMax1 Then
    ValMax2 = ValMax1
End If
ValMax1 = ValMax2
Next s
'
krubrique1 = 44
Sheets("Options_stockage").Cells(val_ligne_deb(krubrique1), val_colonne_deb(krubrique1)) = ValMax1
ValMax1 = 0
ValMax2 = 0
'
' AR
ValMax1 = Sheets("Transport").Cells(19, 88)
For s = 20 To 67
t = 88
If Sheets("Transport").Cells(s, t) > ValMax1 Then
    ValMax2 = Sheets("Transport").Cells(s, t)
ElseIf Sheets("Transport").Cells(s, t) > ValMax1 Then
    ValMax2 = ValMax1
End If
ValMax1 = ValMax2
Next s
'
krubrique2 = 45
Sheets("Options_stockage").Cells(val_ligne_deb(krubrique2), val_colonne_deb(krubrique2)) = ValMax1
ValMax1 = 0
ValMax2 = 0
'

```

If...Then...Elseif...End If

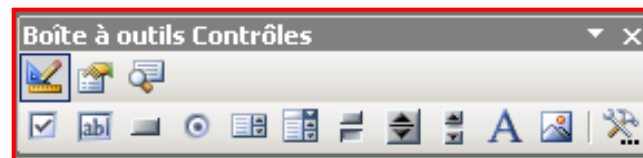
Boucles itératives : For...Next...End

Appel de fonctions

Etc...

Création d'interface

- Boîte à outils de contrôles :



Microsoft Excel - ADGC_v9

Fichier Edition Affichage Insertion Format Outils Données Fenêtre ?

Tapez une question

gembloux faculté universitaire des sciences agronomiques

Outil d'aide à la décision pour la gestion des chablis en Région wallonne

SPW Service public de Wallonie

TABLEAU DE BORD

MODE D'EMPLOI

ESTIMATION DES DEGÂTS

Type d'essence	Forêt soumise (m ²)	Forêt privée (m ²)	Région Wallonne (m ²)
EPICEA	3.100.000	3.100.000	6.200.000
AUTRES RESINEUX	300.000	300.000	600.000
HETRE	300.000	300.000	600.000
AUTRES FEUILLUS	300.000	300.000	600.000
TOTAL (m²)	4.000.000	4.000.000	8.000.000

Mise à jour

PARAMETRES DE SIMULATION

Mois de la tempête :

Délai avant simulation : mois

LANCER LA SIMULATION

VISUALISER LES RESULTATS GRAPHIQUES

SEGMENT VENTE

Options VENTE

* Etat des ventes :

Type d'essence	Vente annuelle (m ²)	Vente avant tempête (m ²)
EPICEA	2.000.000	1.200.000
AUTRES RESINEUX	400.000	240.000
HETRE	600.000	360.000
AUTRES FEUILLUS	1.000.000	600.000

* Echange entre bois chablis/BSP : %

Type d'essence	BOIS SUR PIED ECHANGE	CHABLIS ECHANGE
EPICEA	0	0
AUTRES RESINEUX	0	0
HETRE	0	0
AUTRES FEUILLUS	0	0

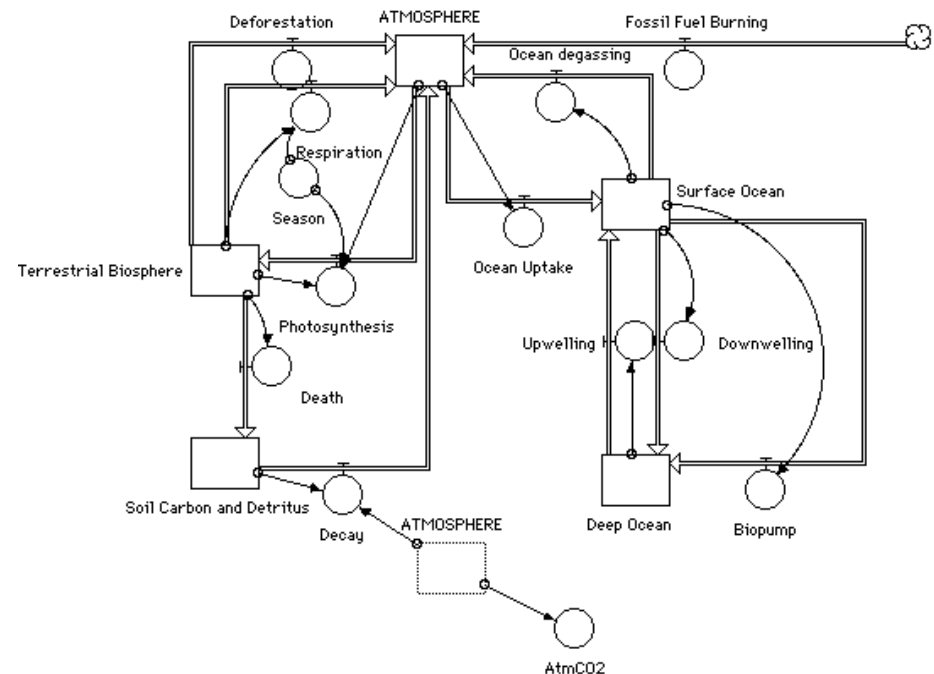
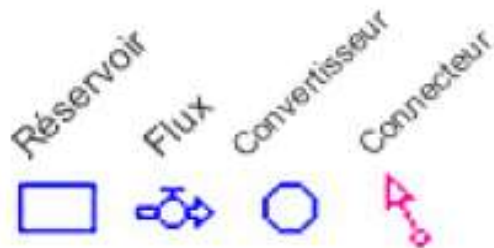
Stella

- Stella[©] est un logiciel qui permet de modéliser d'une façon simple des systèmes complexes, de regarder leur évolution, d'agir sur certains paramètres pour tester différents scénarios.
- <http://www.iseesystems.com/>
- Deux modules :
 - **Graphique** : sert à construire le modèle puis à présenter les résultats sous forme de courbes
 - **Mathématique** : ensemble d'équations différentielles définies à partir du module graphique.



Stella

- **Interface graphique :**
 - glisser-déposer les principaux constituants du modèle
 - organiser leurs interactions
 - définir les conditions initiales
 - composer un écran de visualisation du comportement du modèle.



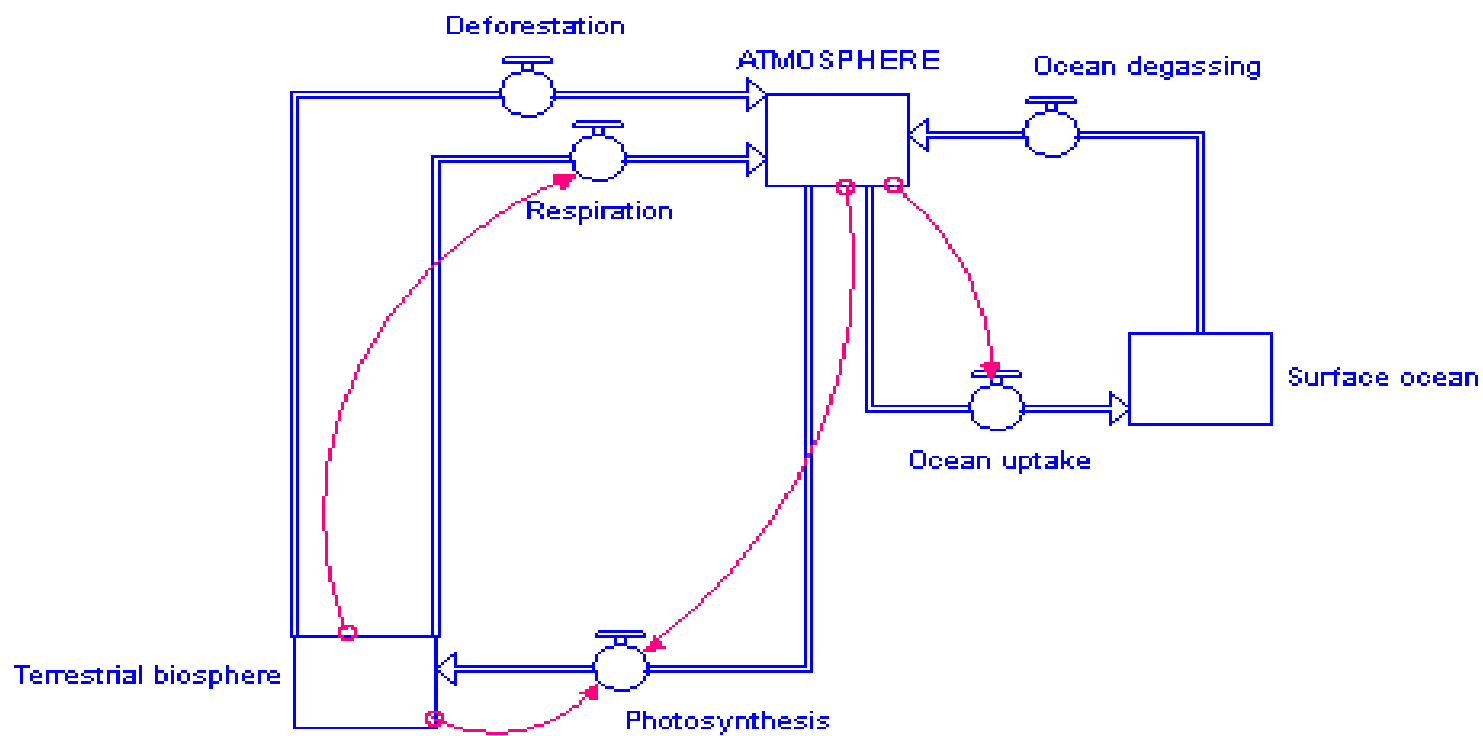


Interface

Map

Model

Equation



Stella

- **Interface mathématique :**
 - Établir les relations entre les stocks et les flux

```

{ INITIALIZATION EQUATIONS }
INIT ATMOSPHERE = 700
INIT Surface_Ocean = 700
Ocean_Uptake = ATMOSPHERE/20
INIT Deep_Ocean = 35000
Upwelling = Deep_Ocean*.002
Downwelling = Surface_Ocean*.002
Biopump = 6.76+Surface_Ocean/700
Ocean_degassing = Surface_Ocean/20
INIT Terrestrial_Biosphere = 550
Season = 1+
(COS(2*PI*(time+0.125)))
Respiration = Terrestrial_Biosphere*.1*Season
INIT Soil_Carbon_and_Detritus = 1200
Photosynthesis = (Terrestrial_Biosphere*.1483+ATMOSPHERE/700)*Season
Death = .05*Terrestrial_Biosphere
Decay = .021*Soil_Carbon_and_Detritus+ATMOSPHERE/750
Fossil_Fuel_Burning = 5
Deforestation = 2

AtmCO2 = ATMOSPHERE/2

```



? ATMOSPHERE(t) = ATMOSPHERE(t- dt) + (Deforestation + Respiration + Ocean_degassing - Photosynthesis - Ocean_uptake) * dt

INIT ATMOSPHERE = { Place initial value here... }

INFLOWS:

Deforestation = { Place right hand side of equation here... }

Respiration = { Place right hand side of equation here... }

Ocean_degassing = { Place right hand side of equation here... }

OUTFLOWS:

Photosynthesis = { Place right hand side of equation here... }

Ocean_uptake = { Place right hand side of equation here... }

? Surface_ocean(t) = Surface_ocean(t- dt) + (Ocean_uptake - Ocean_degassing) * dt

INIT Surface_ocean = { Place initial value here... }

INFLOWS:

Ocean_uptake = { Place right hand side of equation here... }

OUTFLOWS:

Ocean_degassing = { Place right hand side of equation here... }

? Terrestrial_biosphere(t) = Terrestrial_biosphere(t- dt) + (Photosynthesis - Deforestation - Respiration) * dt

INIT Terrestrial_biosphere = { Place initial value here... }

INFLOWS:

Photosynthesis = { Place right hand side of equation here... }

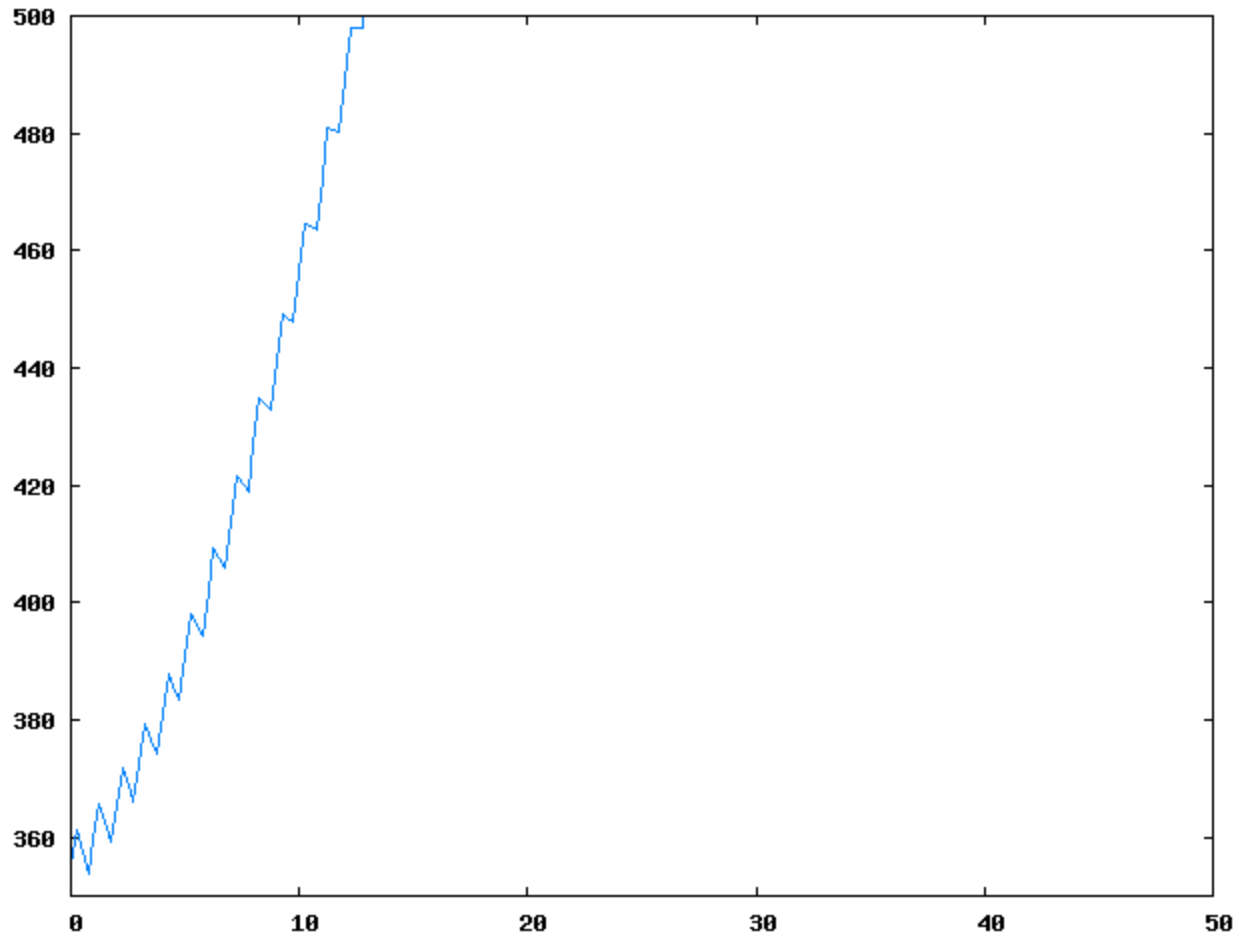
OUTFLOWS:

Deforestation = { Place right hand side of equation here... }

Respiration = { Place right hand side of equation here... }

Stella

- Résultats graphiques :



Value of atmospheric CO2 plotted (on the y axis) as a function of time (on the x axis).

MATLAB®

- MATLAB® est un langage interprété de haut niveau permettant l'exécution de tâches nécessitant une grande puissance de calcul et dont la mise en œuvre sera plus rapide qu'avec des langages de programmation traditionnels tels que le C, le C++ ou le fortran.
- <http://www.mathworks.fr/>
- Possibilités :
 - Développement d'algorithmes ;
 - Analyse et accès aux données ;
 - Visualisation des données (2D/3D) ;
 - Exécution de calculs numériques ;
 - Publication des résultats ;



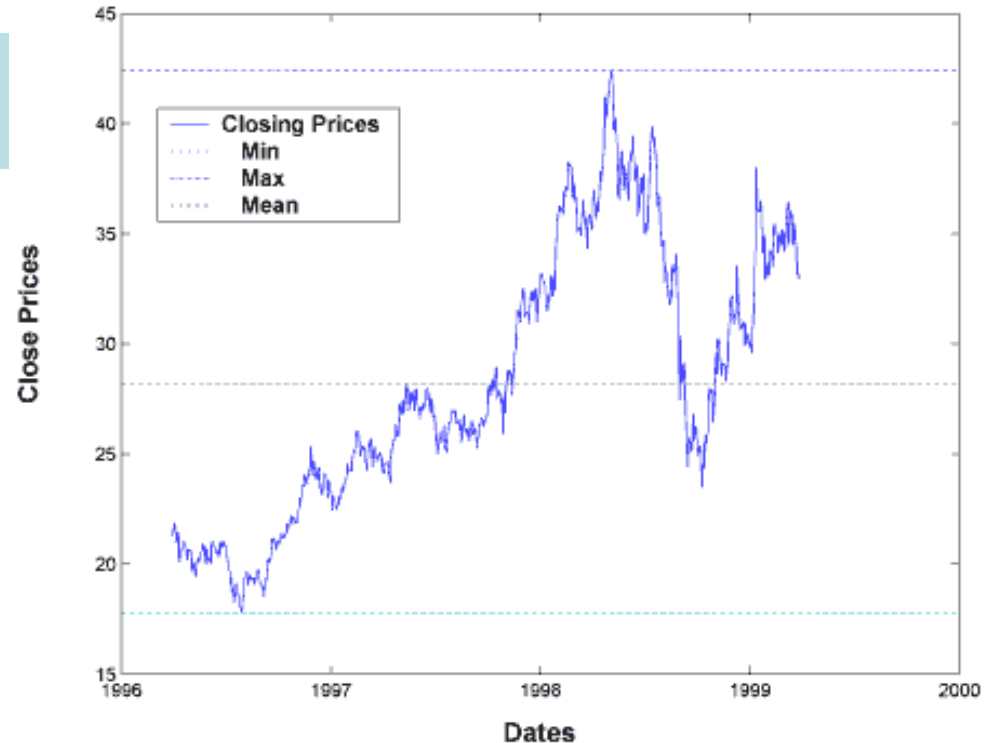
MATLAB®

- Domaines d'applications :

- Calcul scientifique ;
- Automatique ;
- Finance et économie ;
- Etc.

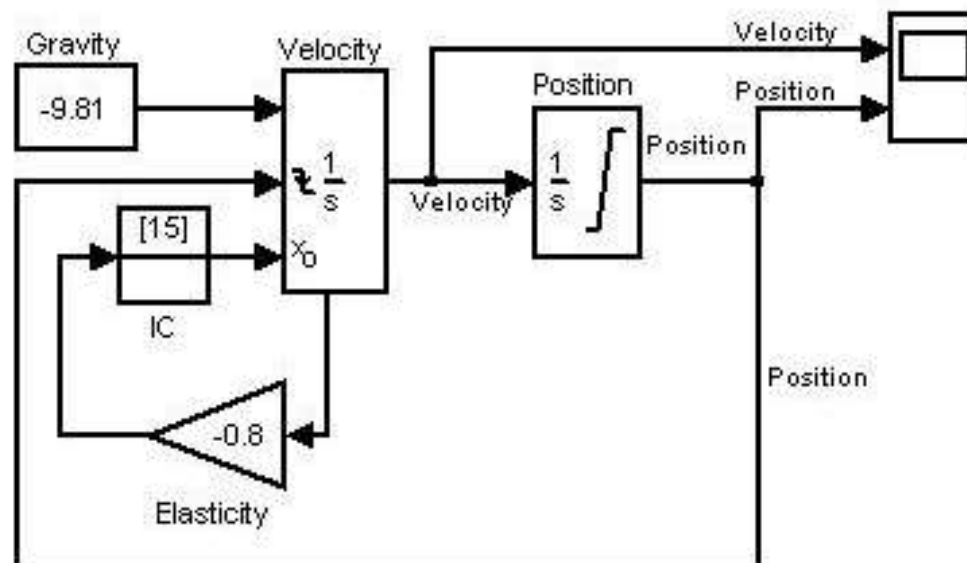
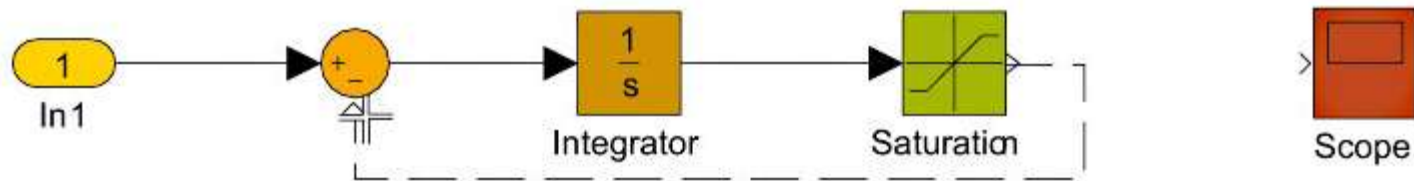
- Modélisation et simulation :

- Noyau de base : MATLAB® contient des centaines de fonctions (opérations matricielles, solveur d'équations différentielles...)
- Boîtes à outils applicatives : codes pré-écrits pour une large gamme de disciplines.
- **Environnement de conception graphique Simulink®** : blocs-diagrammes pouvant appeler toutes les fonctions de MATLAB® et les programmes écrits par l'utilisateur.



Simulink : création de modèles

- Simulink® permet de créer, modéliser et gérer un schéma-bloc détaillé d'un système à partir d'un ensemble complet de bloc prédéfinis.



Simulink : explorateur de modèle

The screenshot shows the Simulink Model Explorer window. The left pane displays the Model Hierarchy, including the Base Workspace and fuel_management sub-workspace. The middle pane shows the Contents of Base Workspace with a table of variables. The right pane shows the configuration for the 'Simulink.Signal: EngineRPM' block, with a dropdown menu open for the Data type, showing options like auto, double, single, int8, uint8, int16, uint16, uint32, and boolean. The 'EngineRPM' signal is currently set to 'uint32' with units of 'rpm' and a maximum value of 8000.

Model Hierarchy

- Simulink Root
 - Base Workspace
 - fuel_management
 - Model Workspace
 - Deployment (Active)
 - Rapid Prototyping
 - HIL
 - Code for fuel_managem...
 - Advice for fuel_managerr...
 - Control Logic

Contents of: Base Workspace

Name	Data Type	Dimensions
input3	single	[1 1]
input4	int16	[1 1]
output	int16	[1 1]
mode	boolean	[1 1]
X	uint16	[1 1]
K1	uint8	[1 1]
K2	auto	[1 1]
T1Break	auto	[1 11]
T1Data	auto	[1 11]
T2Break	auto	[1 3]
T2Data	auto	[3 3]
UPPER	auto	[1 1]
LOWER	auto	[1 1]
EngineRPM	uint32	[1 1]
VehicleSpd	uint32	[1 1]

Simulink.Signal: EngineRPM

Data type: **uint32** (dropdown menu open showing: auto, double, single, int8, uint8, int16, uint16, **uint32**, boolean)

Units: rpm

Dimensions: auto

Complexity: real

Sample time: single

Sample mode: Sample based

Minimum: uint8

Maximum: 8000

Code generation: (checkbox checked)

Storage class: (dropdown menu)

Custom attributes: (checkbox checked)

Struct name: SIGNAL

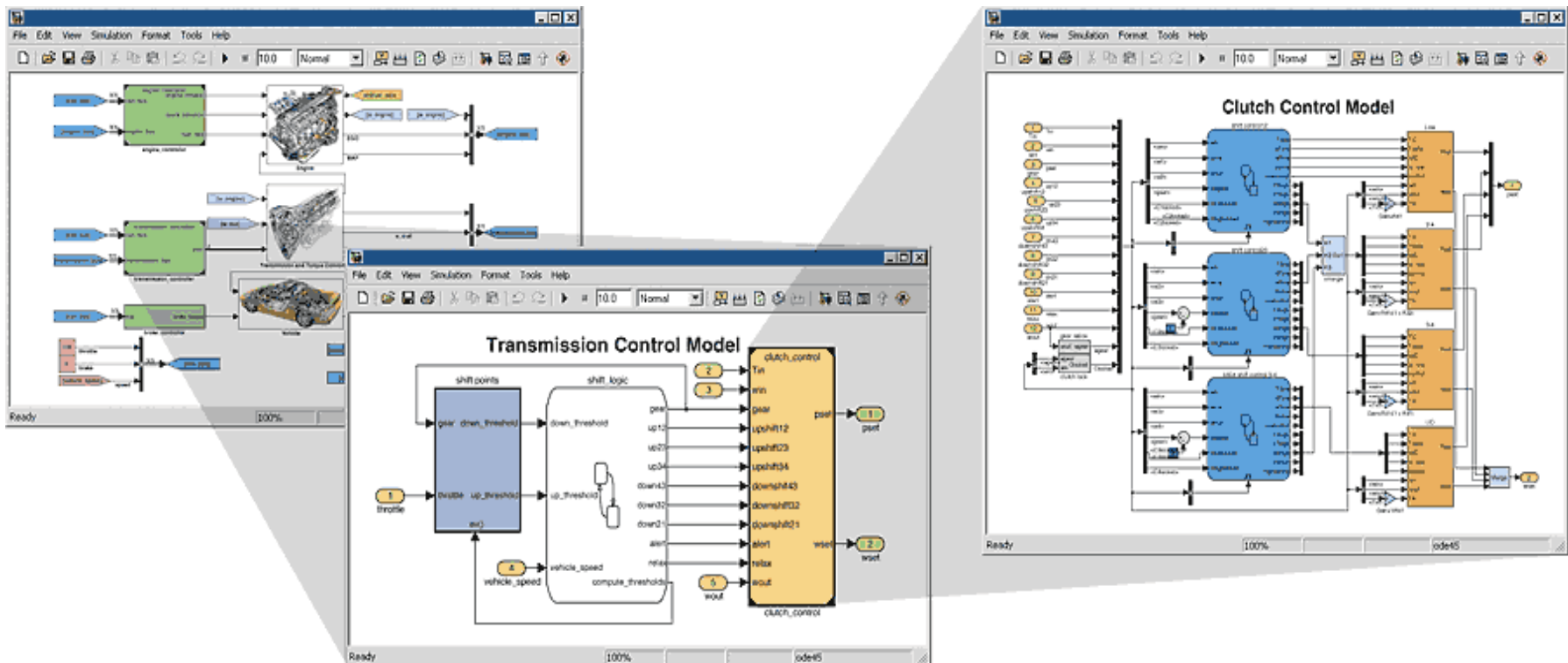
Alias: (text field)

Description: Engine speed of Vehicle

Buttons: Revert, Help, Apply

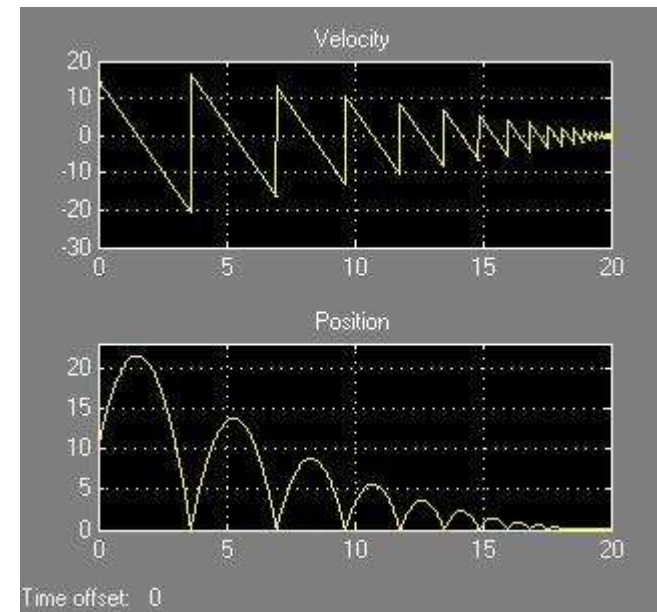
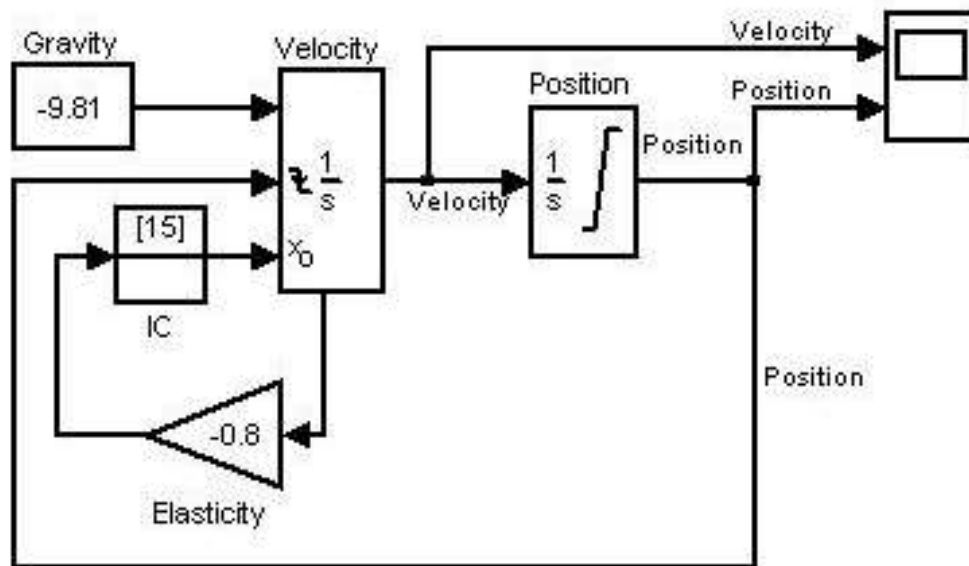
Simulink : sous-systèmes

- Les sous-systèmes permettent d'organiser le modèle en niveaux hiérarchiques clairs et faciles à gérer. Les sous-systèmes renferment un groupe de blocs et de signaux en un seul bloc.



Simulink : exécution d'une simulation

- Les solveurs de Simulink permettent de simuler le comportement dynamique du système et afficher les résultats « en direct ».



- Simulink® offre plusieurs outils permettant d'analyser le système, de visualiser les résultats, et de tester, valider et documenter les modèles.

En conclusion

- **VBA Excel :**

- (+) coût faible voir inexistant du logiciel

- (+) possibilités offertes par VB combinées avec la puissance d'Excel

- (+) utilisable par tous mais attention à la compatibilité (97, XP, Vista)

- (-) risques élevés d'erreurs en programmant (beaucoup de code)

- (-) capacité limitée du classeur en nombre de caractères !

- **Stella :**

- (+) interface de création très didactique pour débiter

- (+) très pratique pour des modèles pas trop complexes

- (-) payant et uniquement en anglais

- **Simulink :**

- (+) combinaison intéressante de VBA et Stella

- (-) payant

Références

Références

- Boulanger P.-M. et Bréchet Th. [2003]. Une analyse comparative des classes de modèles. Action de support PADD I, SSTC, 23p.

<http://users.skynet.be/idd/documents/model/models200103.pdf>

- Gascogne V. [1997]. La modélisation en dynamique des systèmes de la filière bois-papier-carton. Document deu SES – Ministère de l'Équipement, France, 33p.

http://www.complexio.eu/pages/documents_complexio.html

- Salini P. [2007]. La dynamique des systèmes. Présentation PowerPoint.

<http://patrice.salini.free.fr/PS.DSPres.pps>

- <http://www.mathworks.fr/>

- <http://www.iseesystems.com/>

- <http://www.vensim.com/>