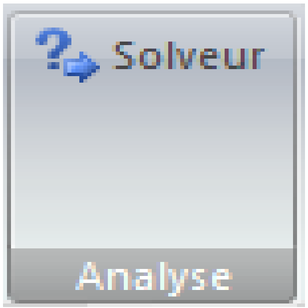
L3 INFO
PROGRAMMATION LINÉAIRE
TRAVAUX PRATIQUES

TP1 : RÉSOLUTION D'UN PROGRAMME LINÉAIRE AVEC LES LOGICIELS EXCEL ET AMPL

1 Logiciel Excel :

1.1 Activer le solveur

Après avoir lancé Excel, il faut activer le solveur, qui ne l'est pas par défaut sur les machines du CRIO. Pour cela, cliquer sur le bouton *microsoft office*, puis sur *options Excel*. Cliquer sur *Complément* puis, dans la zone Gérer, sélectionner *Compléments Exce*l. Cliquez sur le bouton *Atteindre*, dans la fenêtre qui s'ouvre cochez Complément *Solver* et cliquez sur Ok. Patientez le temps de l'installation. Si tout a bien fonctionné, dans l'onglet Données est apparu un groupe tout à droite **Analyse** avec dans ce groupe **Solver**.



1.2 Résoudre un exemple de production

Example 1.1. *Une usine produit deux modèles de machines, l'une que l'on appellera modèle A exige 2 kgs de matière première et de 30 heures de fabrication et donne un bénéfice de 7 ?. L'autre que l'on appellera B exige 4 kgs de matière première et de 15 heures de fabrication et donne un bénéfice de 6 ?. On dispose de 200 kgs de matière première et de 1200 h de travail. Quelle production doit on avoir pour obtenir un bénéfice maximal ?*

On pose x_1 et x_2 respectivement le nombre de produits A et B à fabriquer. Le programme linéaire à résoudre est donc :

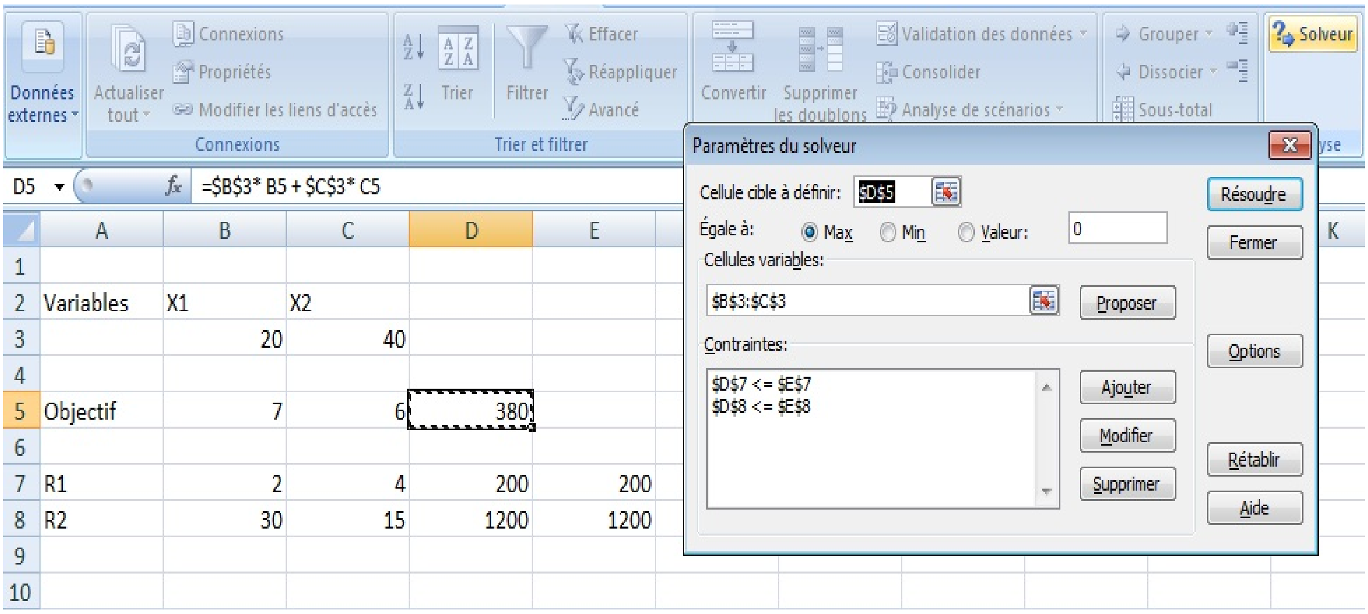
$$(P_1)\left\{\begin{array}{lllll} \max & 7\,x_1 & + & 6\,x_2 & \\ s.c. & & & & \\ & 2\,x_1 & + & 4\,x_2 & \leq & 200 \\ & 30\,x_1 & + & 15\,x_2 & \leq & 1200 \\ & x_1, & & x_2 & \geq & 0 \end{array}\right.$$

Pour le résoudre avec un *solveur Excel*, il faut saisir l'ensemble des coefficients, paramètres et équations du programme linéaire sous une feuille de calcul Excel, sous un format tableau. Pour cela :

— Affecter chaque coefficient ou paramètre à une cellule de la feuille de calcul,

— affecter une cellule à chacune des variables de décision (il n'est pas nécessaire d'attribuer à chacune une valeur initiale),

— puis rentrer les fonctions linéaires associées à la fonction objectif et aux contraintes. Il s'agit d'associer à chaque fonction une cellule dans laquelle se trouvera la formule qui permet de la calculer. Nous rappelons que pour définir une formule dans une cellule, il faut commencer celle-ci par le caractère "=".



Il est maintenant possible d'utiliser le solveur Excel pour la résolution de ce programme linéaire. Pour cela, choisir Solveur dans le menu Outils et remplir les champs suivants :

— dans **Cellule cible** il faut donner l'adresse de la cellule où est précisé la formule de la fonction objectif,

— préciser si l'objectif est à **minimiser, maximiser** (hypothèse retenue par défaut), ou si l'on veut qu'il atteigne une certaine valeur à préciser,

— dans **cellules variables** il faut donner les cellules dans lesquelles seront affectées les valeurs des variables de décision en les séparant par des " :",

— il reste à exprimer l'ensemble des contraintes qui peuvent s'ajouter une à une, ou par groupe de même type, avec la commande Ajouter. Il faut mettre dans le premier champ la cellule où se trouve la formule de la ième contrainte, dans le second champ choisir le type de contrainte dans le menu déroulant et enfin, dans le dernier champ, donner la valeur du second membre de la contrainte, soit directement, soit par le nom de la cellule contenant cette information. A ces contraintes, il faut ajouter les contraintes de signe sur les variables qui sont par défaut sans contrainte de signe. Si toutes les variables sont supposées positives, alors utiliser le champ **Options** du solveur et sélectionner modèle **non-négatif**.

Une fois ces champs remplis, on peut lancer la résolution avec la commande **Résoudre**. Vous devez préciser si l'on veut (c'est ce qui est fait par défaut) que la solution optimale apparaisse sur la feuille de calcul comme valeur des variables ou si l'on ne veut pas changer les valeurs actuelles des variables de décision. De plus, vous pouvez demander 3 rapports supplémentaires :

— celui dit de **réponses** retourne les valeurs optimales des variables de décision, de la fonction objectif optimale et des ressources utilisées pour cette solution optimale ;

— celui dit de **sensibilité** retourne l'analyse de sensibilité de la solution optimale, c'est-à-dire les intervalles dans lesquels peuvent varier les coefficients de la fonction objectif ou un second membre d'une contrainte tout en gardant la même base optimale ;
— celui dit de **limites** donne pour chaque variable l'intervalle dans lequel elle peut varier et dans quel intervalle peut varier sa contribution dans la fonction objectif.

Il faut sélectionner le ou les rapports que l'on souhaite avoir avec la résolution.

Exercice 1 :

Un four peut fabriquer deux types de gâteaux A et B. Il met 2 minutes pour fabriquer un gâteau de type A, 1 minute pour un gâteau de type B.

L' usure, et donc le remplacement, des parties du four, interdit de fabriquer en une heure plus de 24 gâteaux de type A et 36 gâteaux de type B. Le refroidissement du four lui interdit de fabriquer plus de 45 gâteaux A et B par heure. Le profit réalisé sur un gâteau de type A est de 100€, sur un gâteau de type B de 200 €.

Déterminez, à l'aide d'Excel, la production horaire permettant de réaliser le profit maximal.

Exercice 2 :

Chris le campeur part en randonnée dans les pyrénées. Il ne peut emporter dans son sac à dos qu'un poids maximal de 23 kgs. Les dix objets qu'il peut potentiellement emporter possèdent chacun un poids et une valeur indiqués dans le tableau suivant :

Objets	A	B	C	D	E	F	G	H	I	J
Valeurs en €	12	11	12	4	5	10	6	12	7	9
Poids	8	9	4	3	5	8	4	6	4	6

1. Écrire un programme linéaire qui permettra à Chris de déterminer les objets qu ?il devra mettre dans le sac de manière à maximiser la valeur totale des objets sans dépasser le poids maximal autorisé pour le sac ? Résoudre ce problème avec le solveur d ?Excel.
2. Supposons à présent que Chris se voit imposer l ?une des trois contraintes suivantes :
   - Contrainte 1 : Il ne peut emporter que quatre objets au maximum.
   - Contrainte 2 : S'il décide d'emporter l'objet I, alors il doit aussi emporter l'objet E
   - Contrainte 3 : S'il décide d'emporter l'objet A alors il ne peut pas emporter l'objet H

   (a) Pour chaque contrainte, déterminer la (ou les) solution(s) qui semble(nt) la (les) meilleure (s) pour Chris.

   (b) Déterminer la (ou les) solution(s) qui semble(nt) la (les) meilleure (s) pour Chris si seules les deux premières contraintes lui sont imposées.

   (c) Déterminer la (ou les) solution(s) qui semble(nt) la (les) meilleure (s) pour Chris si seules les première et troisième contraintes lui sont imposées.

   (d) Déterminer la (ou les) solution(s) qui semble(nt) la (les) meilleure (s) pour Chris si seules les deuxième et troisième contraintes lui sont imposées.

   (e) Déterminer la (ou les) solution(s) qui semble(nt) la (les) meilleure (s) pour Chris si les trois contraintes lui sont imposées.

2 Logiciel AMPL

AMPL est un langage de modélisation algébrique puissant dédié à la résolution de problèmes d'optimisation. Nous nous intéresserons ici uniquement au cas de la programmation linéaire en variables continues. AMPL permet d'utiliser des notations compactes pour formuler un modèle d'optimisation, qui sera traduit sous une forme directement exploitable par un solveur et fait donc ensuite appel à des solveurs appropriés pour la résolution (par exemple MINOS ou CPLEX pour la programmation linéaire). AMPL est commercialisé, cependant il est possible d'en obtenir une version démo gratuite qui est limitée sur la taille des problèmes (500 variables, 500 contraintes + la fonction objectif, pour la programmation linéaire). Cette version peut être téléchargée sur `http://www.ampl.com/try-ampl/download-a-freedemo/`. Il existe une version exécutable sur une console, moins conviviale, et une version liée à un éditeur, amplide, installée au CRIO. Si vous installez AMPL chez vous, il faudra bien mettre dans le même répertoire l'exécutable d'AMPL, les solveurs et les fichiers décrivant vos problèmes à optimiser.

La formulation utilisée est proche de la représentation algébrique d'un programme linéaire, c'est-à-dire basée sur l'utilisation des notations mathématiques traditionnelles pour décrire l'objectif et les contraintes. L'utilisation d'AMPL est particulièrement recommandée pour la résolution répétée de programmes linéaires sujets à changements. AMPL fournit à l'utilisateur un moyen simple pour décrire/modifier un modèle, en manipulant des notations algébriques, et en séparant le modèle des paramètres. Il est à noter que l'intérêt majeur se trouve dans la façon de décrire séparément à la fois le modèle et à la fois les données de gros programmes linéaires complexes.

Pour la résolution d'un programme linéaire (noté PL dans toute la suite), il faut commencer par rentrer ce PL au format AMPL, que nous allons vous présenter, charger ce modèle puis en demander la résolution.

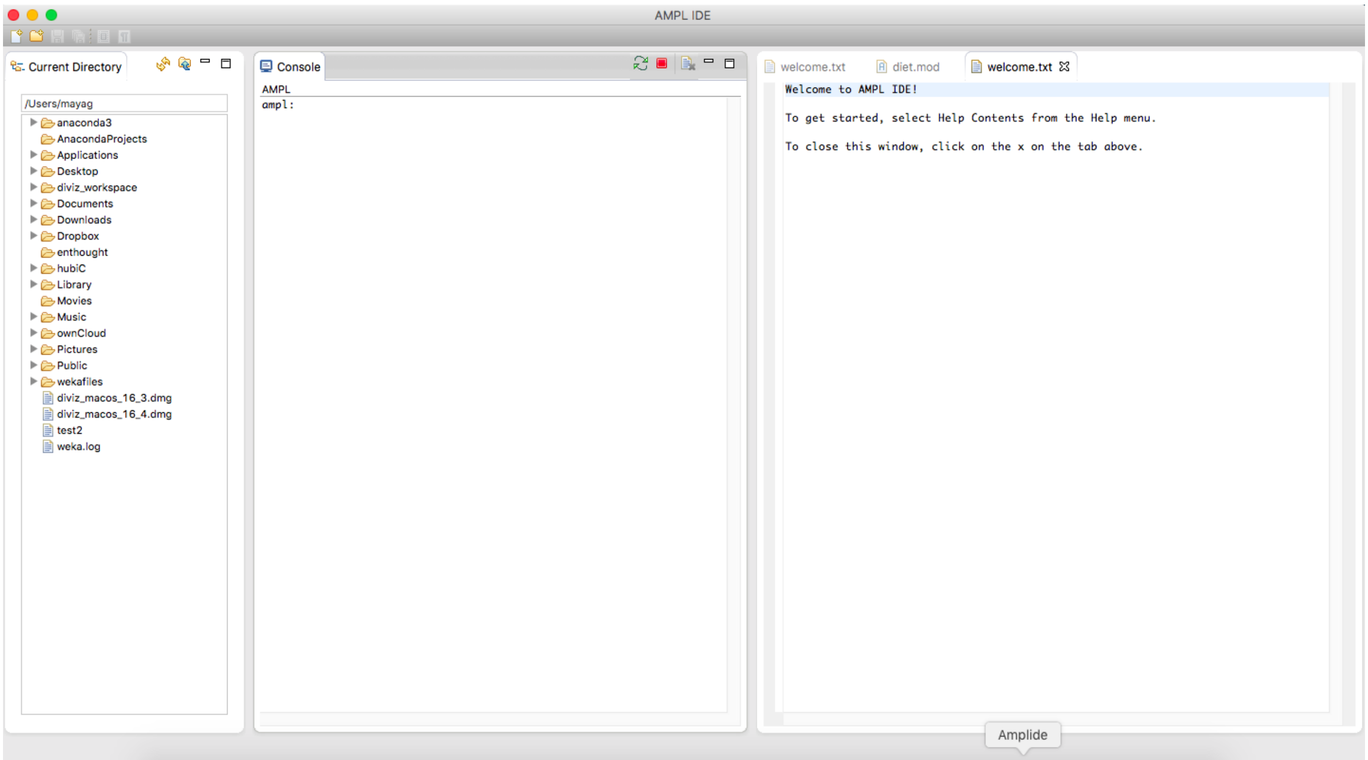


FIGURE 1 – Environnement de travail - AMPLIDE

A Dauphine, pour lancer AMPL, utiliser le raccourci vers l'exécutable AMPLIDE qui est sur le bureau. L'environnement de travail d'AMPL est représenté dans la figure ci-dessus. Sur la gauche, se trouve l'arborescence de vos dossiers de travail et le dossier courant. Un dossier de travail contiendra les fichiers suivants :

— un ou plusieurs fichiers dits modèles, ayant l'extension .mod

— un ou plusieurs fichiers de données, ayant l'extension .dat (il peut n'exister aucun fichier .dat)

Au centre, se trouve la console dans laquelle vous taperez les commandes ampl pour charger un problème, le résoudre et afficher un certain nombre d'informations. Enfin, sur la droite se trouve l'éditeur permettant de créer vos différents fichiers.

2.1 Exemple

$$(PL)\left\{\begin{array}{lllllll} \max & 2000\,x_1 & + & 3000\,x_2 & & & \\ s.c. & & & & & & \\ & x_1 & + & 6\,x_2 & \leq & 30 & (R1) \\ & 2\,x_1 & + & 2\,x_2 & \leq & 15 & (R2) \\ & 4\,x_1 & + & x_2 & \leq & 24 & (R3) \\ & x_1, & & x_2 & \geq & 0 & \end{array}\right.$$

2.2 Premier modèle

Commencer par créer dans l'éditeur à droite, un nouveau fichier que l'on appellera `Exemplecours1` avec l'extension `.mod`. Il va falloir traduire le PL ci-dessus au format AMPL que l'on sauvera dans le fichier `Exemplecours1.mod`. La description AMPL de ce programme linéaire est alors la suivante :

```
var x{i in 1..2};
maximize CA : 2000 * x[1] + 3000 * x[2];
subject to R1 : x[1] + 6* x[2] <= 30;
subject to R2 :2*x[1]+2*x[2] <= 15;
subject to R3 : 4*x[1]+x[2] <= 24;
subject to positivite{i in 1..2} : x[i]>=0;
```

Dans ce modèle, la syntaxe est la suivante :

— le mot-clé **var** permet de déclarer une ou plusieurs variables qui par défaut sont des nombres réels (sinon, il faudra préciser leur type, par exemple *integer*, *binary* ...) après leur nom. Le fait d'utiliser les accolades derrières le nom permet de définir un ensemble d'indices associés à ce nom et donc définir un ensemble de variables. Ainsi, la déclaration `var x{i in 1..2};` permet de créer les variables `x[i]` pour `i = 1::2`. On aurait tout aussi bien pu écrire `var x{1..2}`. Ce sont ces valeurs que le solveur devra calculer,

— le type de *1..2* est considéré comme un ensemble,

— le mot-clé **maximize** (ou **minimize**) permet de déclarer la fonction objectif à optimiser. La syntaxe permet alors de donner un nom à cette fonction objectif, ici `CA`, suivi de sa formule,

— les contraintes sont spécifiées chacune à l'aide du mot-clé `subject to` suivi du nom, puis la formule de celle-ci (la multiplication nécessite l'opérateur $\star$, la relation $\leq$ est traduite par <=).

Les noms et formules sont séparés par un " :" et à la fin de chaque déclaration il faut mettre un " ;".

2.3 Résolution

Dans la fenêtre du centre, nommée `console`, il est alors possible de demander sa résolution en effectuant les commandes suivantes. La ligne de commande de la console commence par :

```
ampl:
```

Il faut commencer par charger le modèle en mettant l'instruction :

```
model Exemplecours1.mod;
```

Si des problèmes apparaissent à l'exécution, il faut modifier le fichier `.mod` puis penser à le sauvegarder à nouveau pour que les modifications soient prises en compte. Avant de relancer la commande précédente, exécuter la commande suivante :

```
reset;
```

Quand la commande `model` s'est bien exécutée, on peut en demander la résolution à l'aide de l'instruction :

```
solve;
```

Afin de visualiser les valeurs optimales des variables ou de la fonction objectif, utiliser l'instruction `display` de la façon suivante :

```
display _varname, _var; display _objname, _obj;
```

ou suivie directement du nom des variables, contraintes, objectifs ... Si en plus, nous souhaitons sauvegarder ces valeurs dans un fichier `exemplecours1.out`, il suffit de rediriger l'affichage sur le fichier de la façon suivante :

```
display _varname, _var > exemplecours1.out;
```

Si le fichier existait déjà, le texte est ajouté à la suite, sinon le fichier est créé dans le répertoire courant.

Exercice 3 :

1. Utiliser AMPL pour obtenir la résolution des deux exemples introductifs (P_1) et (PL) ci-dessus.
2. Utiliser AMPL pour résoudre les programmes linéaires des exercices 1 et 2.

Exercice 4 :

Un assembleur de mobiles doit fournir par contrat 20000 téléphones dans les quatre prochaines semaines. Le client payera 20 € pour chaque mobile livré avant la fin de la première semaine, 18 € pour ceux livrés avant la fin de la deuxième semaine, 16 € pour ceux livrés avant la fin de la troisième semaine et 14 € avant la fin de la quatrième. Chaque ouvrier peut assembler 50 mobiles par semaine. La société ne peut honorer la commande avec ses 40 ouvriers, ainsi elle doit embaucher et former des travailleurs temporaires. Chacun des 40 ouvriers permanents peut être affecté à la formation d'une classe de trois travailleurs temporaires. Après une semaine de formation, ceux qui ont suivi la formation peuvent soit monter des mobiles soit instruire des ouvriers non qualifiés.

A cet instant il n'y a pas d'autre contrat en cours mais tous les ouvriers, permanents ou temporaires, seront payés jusqu'à la fin des quatre semaines (même si certains sont inoccupés).

Un ouvrier qui produit des mobiles, est inactif ou instruit reçoit un salaire de 200 € par semaine alors qu'un ouvrier en formation perçoit 100 € par semaine. Le coût de production (sans compter les salaires) est de 5 € par mobile.

Par exemple, la compagnie peut adopter le programme de fabrication suivant.

- Semaine 1 :
  - 10 assembleurs, 30 instructeurs, 90 apprentis
  - Salaires des travailleurs : 8000 €
  - Salaires des apprentis : 9000 €
  - Profit sur les 500 mobiles : 7500 €
  - Perte nette : 9500 €
- Semaine 2 :
  - 120 assembleurs, 10 instructeurs, 30 apprentis
  - Salaires des travailleurs : 26000 €
  - Salaires des apprentis : 3000 €
  - Profit sur les 6000 mobiles : 78000 €
  - Profit net : 49000 €
- Semaine 3 :
  - 160 assembleurs
  - Salaires des travailleurs : 32000 €
  - Profit sur les 8000 mobiles : 88000 €
  - Profit net : 56000 €
- Semaine 4 :
  - 110 assembleurs, 50 inactifs
  - Salaires des travailleurs : 32000 €
  - Profit sur les 5500 mobiles : 49500 €
  - Profit net : 17500 €

Ce programme de planification qui rapporte 113000 € à la compagnie est l'un des nombreux possibles. La compagnie souhaite faire le meilleur bénéfice possible.

Formulez ce problème sous la forme d'un Programme Linéaire et le résoudre avec le logiciel AMPL.