

## PRISE EN MAIN DU LOGICIEL LABVIEW

## SOMMAIRE

## I - Présentation

<b>1) Introduction</b>	<b>p. 3</b>
<b>2) La face avant</b>	<b>p. 3</b>
<b>3) Le diagramme</b>	<b>p. 4</b>
<b>4) Le flux de données</b>	<b>p. 4</b>
<b>5) Les palettes</b>	<b>p. 5</b>
o La palette d'outils	p. 5
o La palette de commandes	p. 5
o La palette de fonctions	p. 5
o Outils de navigation des palettes	p. 6
o Détails de la palette "outils"	p. 6
o Détails de la palette "commandes"	p. 7
o Détails de la palette "fonctions"	p. 8
<b>6) Les barres d'outils</b>	<b>p. 10</b>
o De la face avant	p. 10
o Du diagramme	p. 11
<b>7) L'aide</b>	<b>p. 11</b>
o vocabulaire :	p. 11
o La fenêtre d'aide	p. 11
o L'aide en ligne	p. 11

## II – Exercices

<b>Exercice 1 : Prise en main avec support visuel.</b>	<b>p. 12</b>
1) Construction de la face avant	p. 12
2) Construction du diagramme	p. 14
3) Câblage et exécution du VI	p. 16
4) Ajout de fonctions supplémentaires	p. 17
<b>Exercice 2 : utilisation de variables booléennes</b>	<b>p. 21</b>
<b>Exercice 3 : registre à décalage</b>	<b>p. 22</b>
<b>Exercice 4 : structure Condition</b>	<b>p. 23</b>
<b>Exercice 5 : utilisation de la structure Séquence</b>	<b>p. 24</b>
<b>Exercice 6 : utilisation d'un Boîte de calcul</b>	<b>p. 25</b>
<b>Exercice 7 : création d'un tableau 1D à l'aide de la boucle For</b>	<b>p. 26</b>
<b>Exercice 8 : créer un tableau 2D</b>	<b>p. 27</b>
<b>Exercice 9 : création d'un graphe XY</b>	<b>p. 27</b>
<b>Exercice 10 : écrire dans un fichier tableur un tableau 1 D</b>	<b>p. 28</b>

<b>Exercice 11 : écrire dans un fichier tableur un tableau 2D</b>	<b>p. 29</b>
<b>Exercice 12 : lecture d'un fichier de type tableur 1D</b>	<b>p. 29</b>
<b>Exercice 13 : lecture d'un fichier de type tableur 2D</b>	<b>p. 30</b>
<b>Exercice 14 : Utilisation d'une variable locale</b>	<b>p. 31</b>
<b>Exercice 15 : Utilisation des propriétés d'une variable</b>	<b>p. 32</b>
<b>Exercice 16 : Créer un sous-VI</b>	<b>p. 33</b>
(1) Réalisation de l'icône	p. 34
(2) Le connecteur	p. 36
(3) Affectation des terminaux aux commandes et aux indicateurs	p. 37
<b>Exercice 17 : Utilisation d'un sous-VI</b>	<b>p. 38</b>

<b>III - Annexe : Les principaux afficheurs de courbes</b>	<b>p. 40</b>
------------------------------------------------------------	--------------

<b>Graphe non déroulant : mono-courbe et multi-courbes</b>	<b>p. 39</b>
<b>Graphe déroulant : mono-courbe et multi-courbes</b>	<b>p. 40</b>
<b>Graphe XY : mono-courbe et multi-courbes</b>	<b>p. 40</b>

#### IV – Références

- Manuel du cours élémentaire LabView  
National Instruments      <http://www.ni.com>  
                                         <http://digital.ni.com/worldwide/france.nsf/main?readform>
- LabView : Programmation et applications  
Francis COTTET, Editions DUNOD
- Introduction à LabView  
Robert FRANCOIS, document au format PDF

## PRISE EN MAIN DU LOGICIEL LABVIEW

### I - Présentation

#### 1) Introduction

Labview est un logiciel de programmation graphique très utilisé en instrumentation car il offre une présentation plus visuelle du problème à résoudre. L'enchaînement hermétique d'instructions d'un langage de programmation classique est remplacé par une succession d'icônes que l'on relie entre elles à l'aide de "fils". La programmation devient alors très intuitive.

Tous les programmes de Labview, communément appelés instruments virtuels (VIs), se composent d'une face avant et d'un diagramme.

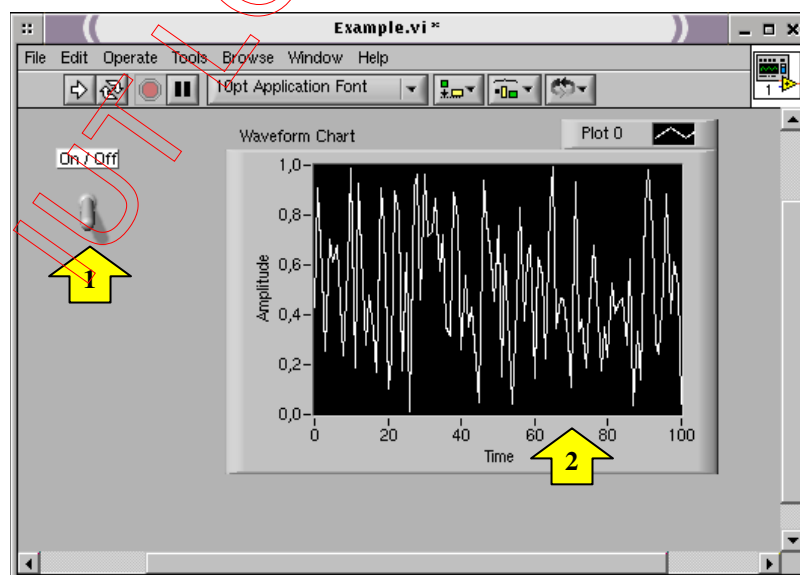
Les outils de programmation sont répartis dans 3 palettes qui contiennent un certain nombre d'options qui servent à créer et à modifier vos VIs :

- o outils
- o commandes
- o fonctions

L'exécution des programmes Labview se fait par flux de données.

#### 2) La face avant

La face avant est l'interface graphique utilisateur/programme de votre VI Labview. Cette interface réceptionne les données entrées par l'utilisateur et affiche celles fournies, en sortie, par le programme. Cette face avant peut contenir des boutons rotatifs, des boutons-poussoirs, des graphes, et d'autres sortes de commandes ou d'indicateurs.



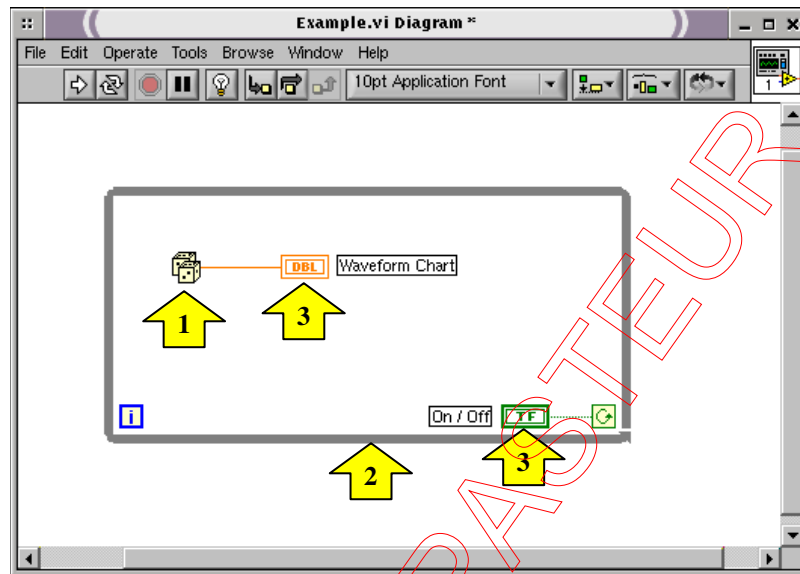
(1) Une commande (entrée)

(2) Un indicateur (sortie)

### 3) Le diagramme

Le diagramme contient le code source graphique de votre VI. Vous y programmez votre VI pour contrôler et remplir des fonctions sur les entrées et sorties créées dans la face avant.

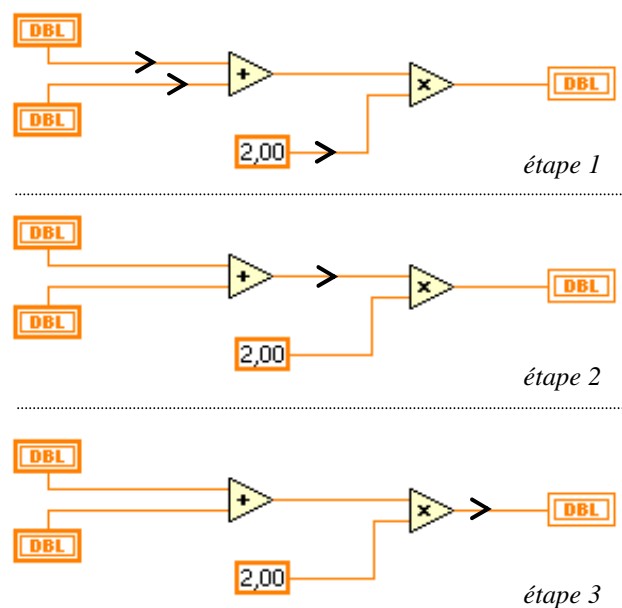
Le diagramme peut contenir des fonctions et des structures issues des bibliothèques de VIs intégrées à Labview. Il peut aussi contenir des terminaux associés à des commandes et à des indicateurs créés dans la face avant.



- (1) Une fonction
- (2) Une structure
- (3) Terminaux de la face avant

### 4) Le flux de données

Les VIs de LabVIEW suivent le concept de l'exécution de programme par flux de données. Le diagramme se compose de nœuds, tels que des VIs, des structures ou des terminaux de la face avant. Ces nœuds sont reliés entre eux par des fils, qui définissent le flux de données à travers le programme. L'exécution d'un nœud n'a lieu que lorsque toutes ses entrées sont disponibles. Dès qu'un nœud a fini de s'exécuter, il transfère toutes ses sorties au prochain nœud dans le chemin de flux de données.



## 5) Les palettes

Les palettes de Labview vous offrent les options dont vous avez besoin pour créer et éditer la face avant et le diagramme.

- o La palette d'outils

S'utilise aussi bien dans la face avant que dans le diagramme. Elle contient les outils nécessaires pour éditer et mettre au point les objets du diagramme et de la face avant.

- o La palette de commandes

Vous utilisez la palette de commandes dans la face avant uniquement. Elle contient les commandes et les indicateurs de face avant qui vous servent à créer l'interface utilisateur.

- o La palette de fonctions

Vous utilisez la palette de fonctions dans le diagramme uniquement. Elle contient les objets qui vous servent à programmer votre VI, comme par exemple des opérations arithmétiques, d'E/S d'instrument, d'E/S sur fichiers et d'acquisition de données.

palette d'outils



palette de commandes



palette de fonctions



## o Outils de navigation des palettes




bouton **Rechercher**. Très pratique au début. Il permet de trouver un VI à partir de son nom ou d'un mot contenu dans son nom.



bouton **Afficher**. Il permet de personnaliser l'affichage des éléments de la palette.

## o Détails de la palette "outils"





Outil **Doigt**. Utilisez-le pour manipuler les commandes et les indicateurs de la face avant. L'outil devient  lorsqu'il passe sur une commande de texte, comme par exemple une commande numérique ou de type chaîne de caractères.



outil **Flèche**. Il sert à sélectionner, déplacer des objets ou à changer leur taille.



outil **Texte**. C'est l'outil  qui sert à saisir du texte dans des étiquettes ou à créer des textes libres. Il devient  lorsque vous créez des étiquettes libres.



outil **Bobine**. Utilisez-le pour câbler des objets dans le diagramme. En le plaçant sur un objet à câbler, vous faites apparaître les terminaux de l'objet. En le plaçant sur un fil, vous affichez dans la fenêtre d'aides des informations qui lui sont relatives. Pour ce faire, vous devez d'abord sélectionner **Show Help Window** dans le menu **Windows**.



outil **Menu local**. Utilisez-le pour ouvrir le menu local d'un objet à l'aide du bouton gauche de la souris.



outil **Main**. Il vous permet de parcourir les fenêtres sans utiliser les barres d'outils



outil **Point d'arrêt**. Il vous permet de fixer des points d'arrêt sur des Vis, des fonctions et des structures.



outil **Sonde**. Il sert à créer des sondes (points de contrôle) sur des fils de câblage.

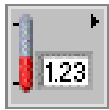


outil **Pipette**. Sert à recopier des couleurs dans d'autres objets à l'aide de l'outil **Pinceau**.



outil **Pinceau**. Permet de colorier un objet. Il permet aussi d'afficher le premier plan ou l'arrière-plan de l'objet choisi.

## o Détails de la palette "commandes"

Menu **Moderne** :

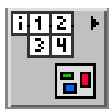
bouton **Numérique**. Il appelle les objets définissant des entrées ou des sorties de données de type numérique entier ou réel



bouton **Booléen**. Il appelle les objets définissant des entrées ou des sorties de données de type booléen (Vrai/Faux ou 0/1)



bouton **Chaîne et chemin**. Il appelle les objets définissant des entrées ou des sorties de données de type chaîne de caractères ou chemin d'accès à des fichiers.



bouton **Tableau, matrice et cluster**. Il appelle les objets définissant des entrées ou des sorties de données de type Tableau ou cluster (enregistrement de données de types différents).



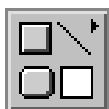
bouton **Liste et table**. Il appelle les objets définissant des entrées ou des sorties de données homogènes en type, structurées et représentées en listes ou en tables.



bouton **Graphique**. Il appelle les objets définissant uniquement des sorties de données sous forme de courbes à 2 ou 3 dimensions.



bouton **Menu déroulant et énum**. Il appelle les objets définissant des entrées ou des sorties de données homogènes présentées sous la forme de menu déroulant.



bouton **Décorations**. Il appelle les objets graphiques permettant d'améliorer la présentation de la face avant.

...

## o Détails de la palette "fonctions"

Menu **Programmation** :

sous-palette **Structures**. Elle se compose des structures comme par exemple les boucles **For** et **While**.



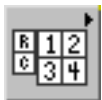
sous-palette **Numérique**. Elle se compose de fonctions arithmétiques, trigonométriques, logarithmiques et numériques.



sous-palette **Booléen**. Elle se compose de fonctions logiques et booléennes.



sous-palette **Chaîne**. Elle se compose de fonctions qui servent à manipuler des chaînes de caractères.



sous-palette **Tableau**. Elle se compose de fonctions qui servent à traiter des tableaux.



sous-palette **Cluster et variant**. Elle se compose de fonctions qui servent à traiter des **Clusters**.



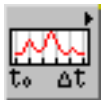
sous-palette **Comparaison**. Elle se compose de fonctions qui servent à comparer des nombres, des booléens et des chaînes de caractères.



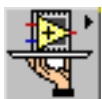
sous-palettes **Données temporelles & Boîtes de dialogue**. Elles contiennent les fonctions de gestion du temps et les fenêtres de dialogue.



sous-palette **E/S sur fichiers**. Elle se compose de fonctions et de Vis pour les Entrées/Sorties sur fichiers.



sous-palette **Waveform**. Elle se compose de fonctions de générations d'ondes, clusters comprenant le temps initial  $t_0$ , l'incrément  $\Delta t$  et les données numériques.



sous-palette **Contrôle d'application**. Elle se compose de Vis permettant le contrôle du programme en mode exécution.



sous-palette **Graphisme et son**. Elle se compose de fonctions de traitements graphiques et de gestion du son.



sous-palette **Génération de rapport**. Elle se compose de Vis permettant la génération d'un rapport, de type texte ou tableur, avec ou sans image, au cours de l'exécution du programme.



**Menu E/S de mesures :**

contient les Vis d'acquisition/génération de données et les drivers d'instruments.

**Menu Sélectionner un VI :**

ouvre une boîte de dialogue permettant de sélectionner le sous-VI à insérer dans le VI courant.

**Menu Mathématiques :**

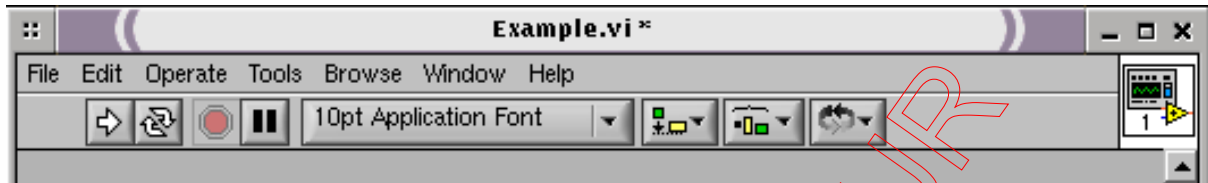
contient les VI d'analyses et traitements mathématiques avancés.

IUT LOUIS PASTEUR

## 6) Les barres d'outils


Chacune des fenêtres de face avant et du diagramme contient une barre d'outils composée de boutons de commandes et d'indicateurs d'état qui sont utilisés pour contrôler le VI. En fait, les deux barres d'outils sont disponibles et s'affichent selon que vous êtes dans la fenêtre du diagramme ou de face avant.

### o De la face avant



bouton **Exécution**. Cliquer dessus pour lancer l'exécution du VI. Il prend alors l'aspect suivant



bouton **Exécution permanente**. Il permet d'exécuter le VI de manière répétitive. En mode exécution permanente, le bouton se transforme en . En cliquant dessus, vous désactivez ce mode de fonctionnement.



bouton **Stop**. Il apparaît lorsque le VI tourne. En cliquant dessus, vous stoppez immédiatement l'exécution du VI. Ce bouton prend alors l'aspect suivant :




bouton **Flèche Brisée**. Il prend la place du bouton Exécution pour indiquer que le VI ne peut pas se compiler à cause d'erreurs. En cliquant sur ce bouton une fenêtre listant ces erreurs s'ouvre.



bouton **Pause/Reprise**. Il interrompt l'exécution du VI. Pour reprendre l'exécution, appuyez à nouveau sur ce bouton.

### o Du diagramme



bouton mode **Animation**. Il met en route le mode **Animation**. Il se transforme alors en  et vous pouvez visualiser le flux des informations qui traversent le diagramme.



bouton **Exécution semi-détaillée**. Cliquez sur ce bouton pour activer le mode Pas à pas et ainsi exécuter le VI nœud par nœud. Il clignote sur chaque nœud pour indiquer qu'il est prêt à l'exécuter. Cliquer à nouveau sur ce bouton pour exécuter une boucle, un sous-VI etc.



bouton **Exécution détaillée**. Cliquez à nouveau sur ce bouton pour exécuter de façon détaillée une boucle, un sous-VI etc.



bouton **Sortie**. Cliquer sur ce bouton pour sortir d'une boucle ou d'un sous-VI. Ce faisant, vous terminez son exécution Pas à pas et passez au nœud suivant.

## 7) L'aide

### o Vocabulaire :

- Cliquer signifiera : appuyer sur le bouton gauche de la souris.
- Poper : faire apparaître le menu contextuel d'un objet en appuyant sur le bouton droit de la souris.

Labview propose plusieurs options d'aide pour les VIs, les sous-VIs et les nœuds. Les deux options les plus souvent utilisées pour la programmation Labview sont la fenêtre d'aide contextuelle et l'aide en ligne.

### o La fenêtre d'aide

Pour afficher la fenêtre d'aide, choisissez **Afficher l'aide contextuelle** dans le menu déroulant **Aide** ou son équivalent raccourci clavier. Lorsque vous placez l'un des outils de la palette **Outils** sur les objets du diagramme ou de la face avant, la fenêtre d'aide affiche l'icône des sous-VIs, des fonctions, des constantes, des contrôles et indicateurs, ainsi que les options des boîtes de dialogue avec des fils rattachés à chaque terminal. Dans cette fenêtre, les terminaux indispensables figurent en gras, les connexions recommandées en texte ordinaire et les connexions optionnelles en gris.

### o L'aide en ligne

L'aide en ligne présente de façon détaillée la plupart des objets du diagramme. On peut y accéder en cliquant sur le menu déroulant **Aide** de la barre d'outils de la fenêtre et en sélectionnant **Rechercher dans l'aide LabView** ou **Recherche d'exemples**.

## II - Exercice 1 : Prise en main avec support visuel.

*Nous allons pas à pas construire un instrument virtuel.*

*(Il est fortement recommandé de sauvegarder tous vos VI – avec les noms proposés – car certains servent dans d'autres exercices)*

### 1) Construction de la face avant

(1) Lancez Labview en cliquant sur l'icône correspondante dans la barre des tâches en bas de l'écran et créez un nouveau VI en sélectionnant **VI vide** dans la boîte de dialogue Labview.

(2) Dans la face avant (grise), cliquez sur le menu déroulant **Affichage** et sélectionnez **Palette des commandes** si celle-ci n'est pas déjà visible à l'écran. Faites apparaître la palette d'outils : menu déroulant **Affichage > Palette des outils**.

***Palette des outils** : cliquer sur l'icône 'clé et tournevis' pour passer en mode manuel (=diode verte éteinte)*

***Palette des commandes** : - cliquer sur **Afficher>Afficher cette palette comme>Catégorie standard***

*- **Afficher>Catégories toujours visibles** : ne conserver que les catégories **Moderne** et **Système**.*

(3) Dans la palette **Commandes**, sélectionnez un interrupteur vertical à bascule **Commandes>Moderne>Booléen>Interrupteur vertical à bascule** en cliquant dessus et en maintenant le bouton de la souris enfoncé.

(4) Déplacez votre curseur sur la face avant et cliquez avec la souris pour positionner l'interrupteur.

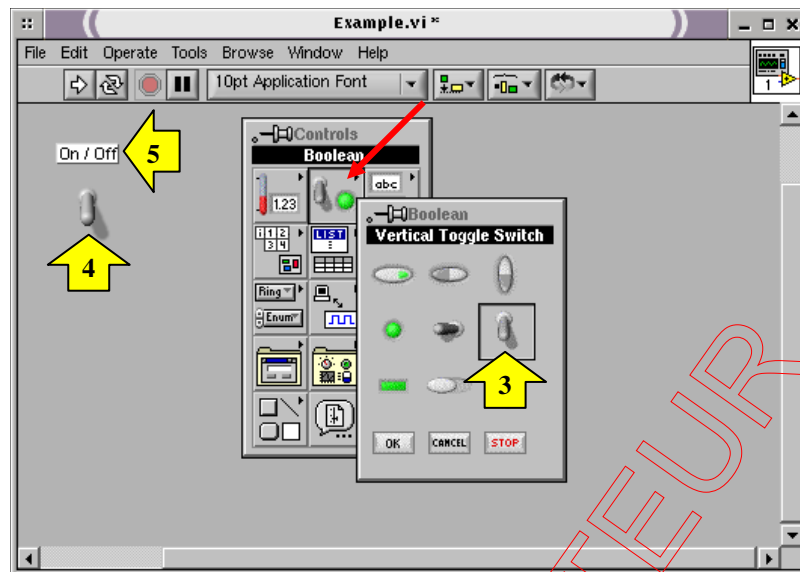
(5) Sélectionnez l'outil **texte** (dans la **Palette des outils**) et remplacez, dans l'étiquette de l'interrupteur, le mot Booléen par marche/Arrêt. Cliquez sur l'interrupteur et choisissez : **Éléments visibles>Texte booléen**

(6) Importez dans la face avant un graphe déroulant **Commandes>Moderne>Graphe>Graphe déroulant**.

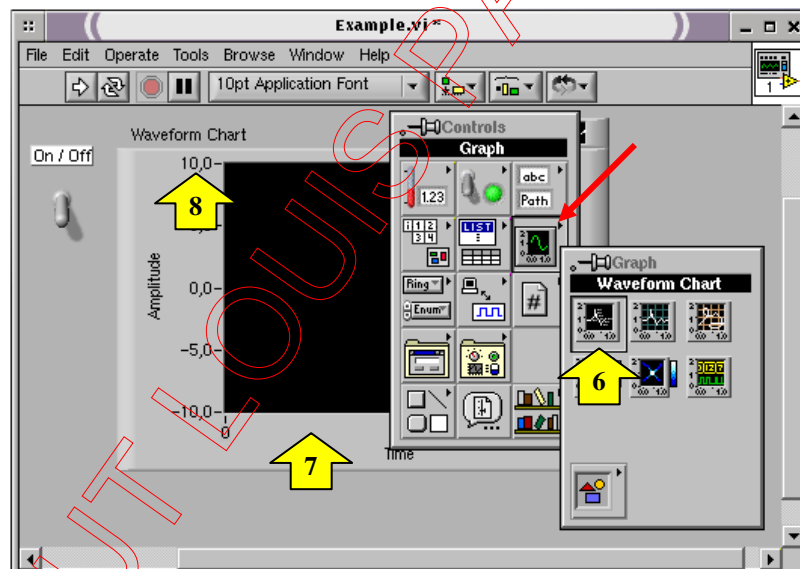
Ce graphe trace les données point par point au fur et à mesure de leur apparition.

(7) Comme précédemment, changez le nom 'Graphe déroulant' par 'Tracé aléatoire' à l'aide de l'outil **texte**.

(8) Pour changer l'échelle de ce graphe déroulant, sélectionnez l'outil **Doigt** dans la palette d'outils. Double-cliquez sur 10,0 sur l'axe des Y de l'indicateur 'Tracé aléatoire' et tapez 1,0.



Sélection de l'interrupteur vertical à bascule



Importation du graphe déroulant

Sauvegarder le programme dans votre répertoire, créé dans le répertoire de votre département, sous le nom : `exo_01.vi` (E :/votre\_département/votre\_répertoire/exo\_01.vi)

La sauvegarde peut s'effectuer indifféremment depuis la fenêtre diagramme ou la face avant. Les 2 fenêtres sont sauvegardées en même temps puisqu'il s'agit de deux visualisations différentes du même programme.

N'oubliez pas de sauvegarder régulièrement...

## 2) Construction du diagramme

Il s'agit maintenant de créer le code source de votre VI.

(1) Allez dans le diagramme (fenêtre blanche) en choisissant **Afficher le diagramme** dans le menu déroulant **Fenêtre** ou en cliquant simplement dans la fenêtre du diagramme. Les deux terminaux présents correspondent à l'interrupteur **Marche/Arrêt** et au graphe **Tracé aléatoire** de la face avant.

(2) Cliquez sur le menu déroulant **Affichage** dans la fenêtre diagramme et sélectionnez la palette de fonctions : menu déroulant **Affichage>Palette des fonctions**.

*Cliquer sur le bouton **Afficher** de la palette de fonctions et sélectionner **Afficher>Afficher cette palette comme>Catégorie (Standard)***

*Cliquer à nouveau sur le bouton **Afficher>Catégories toujours visible>** et ne conserver que les catégories suivantes : **Programmation, E/S de mesure, Mathématiques, Traitement du signal et Sélectionner un VI**.*

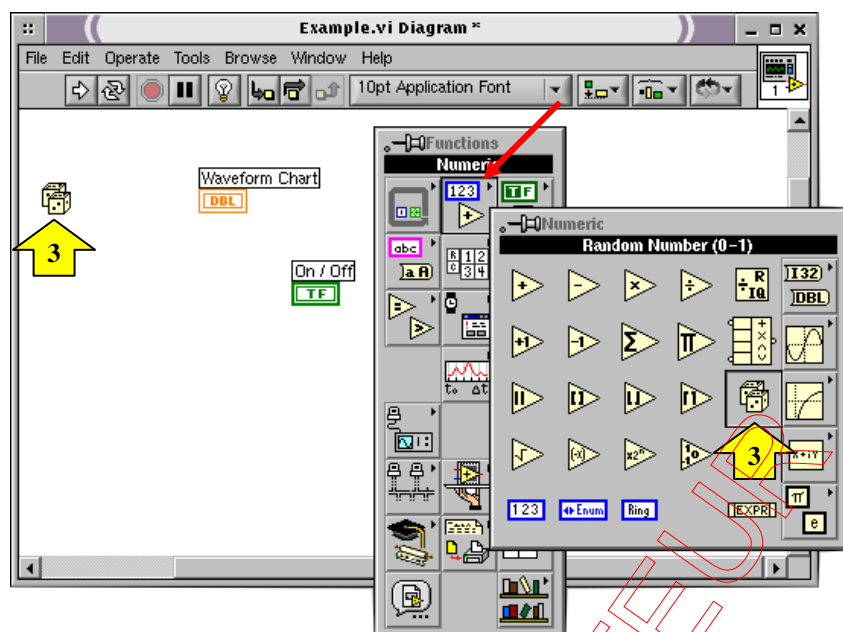
(3) Sélectionnez et placez dans votre diagramme le générateur de nombres aléatoires issu de la palette de fonctions : **Fonctions>Programmation>Numérique>Nombre aléatoire (0-1)**. Ce VI génère un nombre aléatoire compris entre zéro et 1.

(4) Une boucle **While** exécute le code tout entier à l'intérieur de ses limites tant que la valeur sur le terminal conditionnel reste vraie (TRUE). Si la condition devient fausse (FALSE), la boucle **While** cesse de s'exécuter. Le terminal conditionnel est le carré jaune à flèche verte et ronde dans le coin inférieur droit de la boucle.

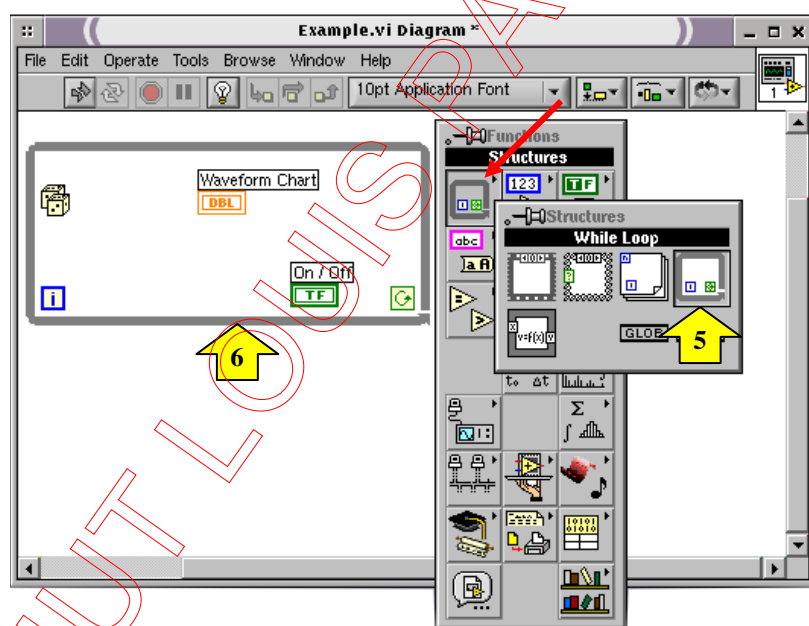
(5) Sélectionnez une boucle **While** dans la palette de Fonctions : **Fonctions>Programmation>Structures>Boucle While**.

(6) Placez votre pointeur dans le diagramme là où vous voulez ancrer le coin supérieur gauche de la boucle. Cliquez et maintenez la pression sur le bouton de la souris. Tirez la boucle en diagonale pour y inclure la fonction **Nombre aléatoire (0-1)**, l'interrupteur et le graphe. Relâchez le bouton de la souris. Ces trois éléments doivent se trouver à l'intérieur de la boucle. Dans le cas contraire, sélectionnez l'outil **Flèche** dans la palette d'outils. Cliquez sur l'objet à déplacer et le positionner à l'intérieur de la boucle.

(7) Poper sur le terminal de condition d'arrêt de la boucle **While** et choisir : **Continuer sur condition vraie**.



Sélection du générateur de nombres aléatoires



Sélection de la boucle while

### 3) Câblage et exécution du VI

Il faut que vous ajoutiez du câblage pour définir le flux de données dans votre VI. Une fois votre VI terminé, vous pourrez l'exécuter à partir de la face avant pour visualiser le tracé des données sur votre graphe déroulant.

(1) Sélectionnez l'outil **Bobine** dans la palette d'outils.

(2) Câblez la fonction **Nombre aléatoire (0-1)** au terminal du graphe déroulant. Pour ce faire, cliquez sur la fonction **Nombre aléatoire (0-1)**, allez sur le terminal du graphe déroulant **Tracé aléatoire**, et cliquez à nouveau dessus.

*Conseils : Lorsque vous positionnez l'outil Bobine sur un terminal, celui-ci clignote et une info-bulle apparaît à son sujet. Lorsque le bon terminal clignote, cliquez dessus pour câbler vers ou en provenance de celui-ci. Il est possible d'effectuer des changements de direction à 90 ° en cliquant en cours de route.*

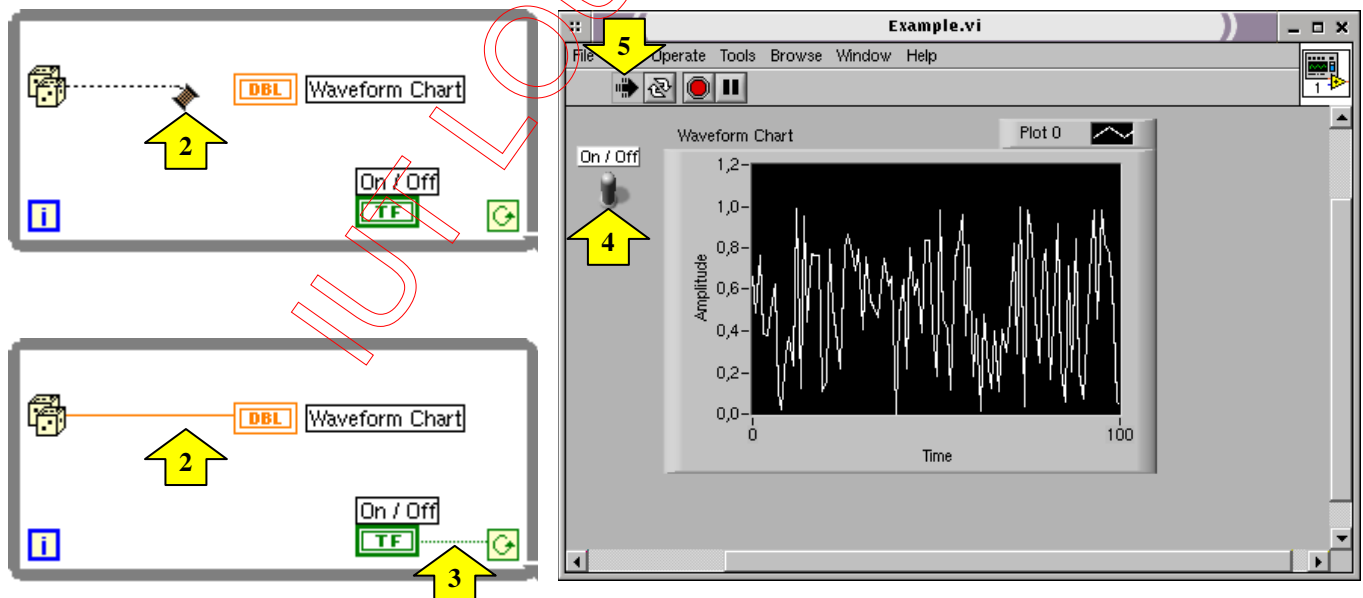
*Si vous voyez des fils noirs en tirets s'afficher, sélectionnez **Supprimer les fils brisés** dans le menu **Edition** ou CTRL B.*

(3) Câblez le terminal de l'interrupteur **Marche/Arrêt** au terminal conditionnel sur la boucle **While**.

(4) Allez dans la face avant et sélectionnez l'outil **Doigt**. Cliquez ensuite sur l'interrupteur pour le mettre en position haute (équivalente à la valeur **Vraie**).

(5) Cliquez sur le bouton **Exécution** de la barre d'outils pour exécuter le VI.

(6) Pour interrompre l'exécution du VI, cliquez à nouveau sur l'interrupteur **Marche/Arrêt** pour le mettre en position basse (équivalente à Faux). Dans la mesure où la boucle **While** s'exécute uniquement tant que son terminal conditionnel reste vrai, le fait de changer la valeur de l'interrupteur arrête la boucle.





#### 4) Ajout de fonctions supplémentaires

##### Temporisez votre VI

Vous pouvez ajouter un délai de temporisation à votre VI pour tracer des points plus lentement sur le graphe déroulant.

(1) Allez dans le diagramme. Sélectionnez la temporisation dans la palette de fonctions :

**Fonctions>Programmation>Informations temporelles>Attendre un multiple de ms.**

(2) Placez la fonction **Attendre un multiple de ms** dans la boucle **While**.

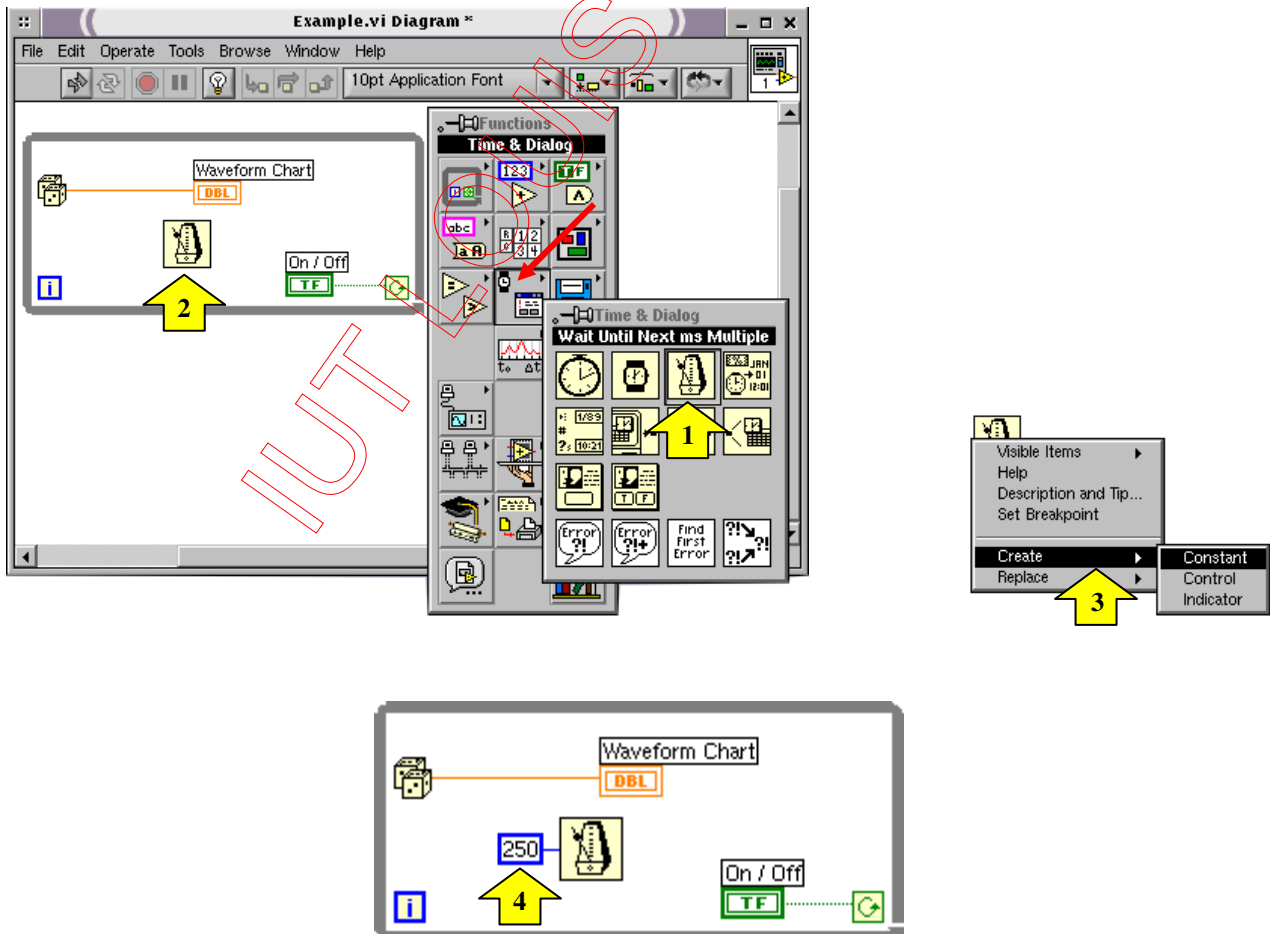
(3) Placer l'outil **Bobine** sur le terminal gauche (= entrée) de la fonction de temporisation. Ce terminal clignote alors. Poper (clic droit) sur le terminal pour faire apparaître son menu local. Choisissez **Créer>constante**. Une constante est créée, automatiquement connectée au terminal. Si la constante n'est pas câblée au terminal (mauvais positionnement de l'outil **Bobine**), la câbler à l'aide de la bobine au terminal gauche

(4) Tapez 250 dans le cadre pour générer et afficher un point toutes les 250 ms.

(5) Allez dans la face avant, cliquez sur l'interrupteur **Marche/Arrêt** pour le mettre en position haute, puis exécutez le VI afin de voir l'effet de la temporisation. Cliquez maintenant sur l'interrupteur pour stopper le VI.

Sauvegarder le programme dans votre répertoire sous le nom : *exo\_01.vi*

(E:\votre\_département\votre\_répertoire\exo\_01.vi)



Ajoutez des fonctions d'analyse et d'E/S sur fichiers dans votre VI

Vous pouvez calculer la moyenne des points de données aléatoires que vous avez collectés et enregistrer vos données dans un fichier au format tableur.

(1) Allez dans la fenêtre 'diagramme' du VI.

(2) Sélectionnez la fonction 'moyenne' dans la palette de Fonctions :

**Fonctions>Mathématiques>Probabilités et statistiques>Moyenne.**

(3) Placez le VI 'moyenne' dans le diagramme à l'extérieur et à droite de la boucle **While**.

(4) Prendre l'outil **Bobine** et le placer sur le terminal de sortie (= à droite) du VI 'moyenne'. Lorsque le terminal clignote, cliquer et choisir **Créer>Indicateur**. Un indicateur numérique est automatiquement créé et connecté au terminal de sortie. Sur la face avant, la moyenne des données aléatoires s'affichera dans cet indicateur.

(5) Dans la palette de fonctions, sélectionnez le VI 'Ecrire dans un fichier tableur' :

**Fonctions>Programmation>E/S sur fichiers>Ecrire dans un fichier tableur**

(6) Placez ce VI dans le diagramme à l'extérieur et à droite de la boucle **While**.

(7) En utilisant l'outil **Bobine**, câblez la fonction **Nombre aléatoire (0-1)** au terminal d'entrée **x** du VI 'moyenne'. Créez cette branche en câblant à partir de la portion existante.

*Conseils :*

*La portion de fil se met à clignoter si l'outil **Bobine** est correctement positionné pour fixer un nouveau fil à partir de la portion existante.*

*L'outil Bobine vous permet de visualiser les étiquettes info-bulles des terminaux sur les nœuds du diagramme.*

(8) Créez une autre branche à partir de celle créée à l'étape précédente. Câblez cette nouvelle branche à l'entrée **Données 1D** du VI **Ecrire dans un fichier tableur**. Vous utilisez **Données 1D** parce que la boucle **While** fournit une rangée unidimensionnelle de données à partir des nombres aléatoires générés.

(9) Le tunnel orange sur la boucle **While** est un terminal de sortie de données.

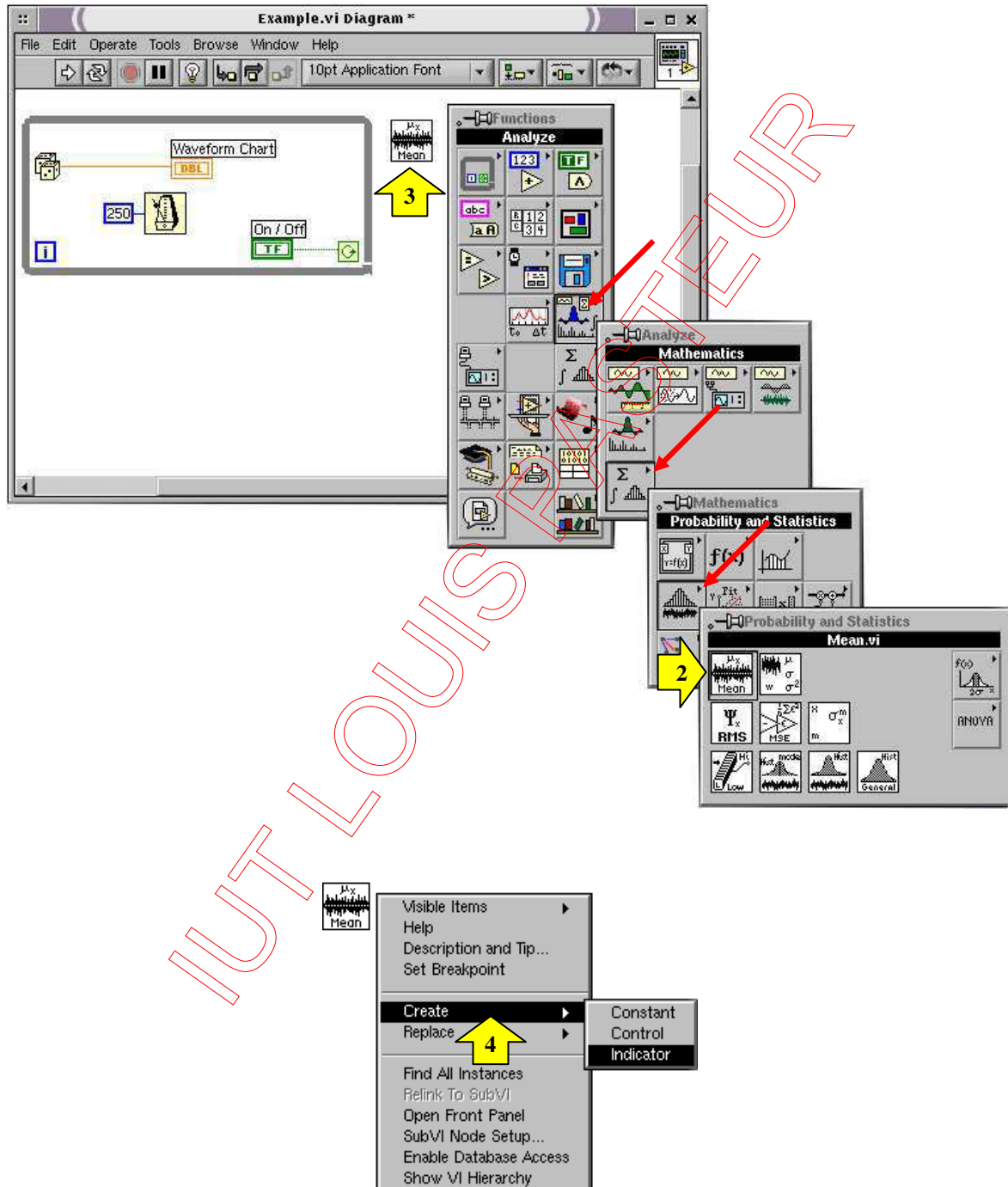
Dans le menu local de ce tunnel orange (clic droit), choisissez d'**Activer l'indexation**. Les fils en tirets se transforment en fils pleins oranges. Activer l'indexation permet à la boucle **While** d'accumuler les données puis de les acheminer au VI 'moyenne' sous forme groupée lorsque la boucle s'arrête.

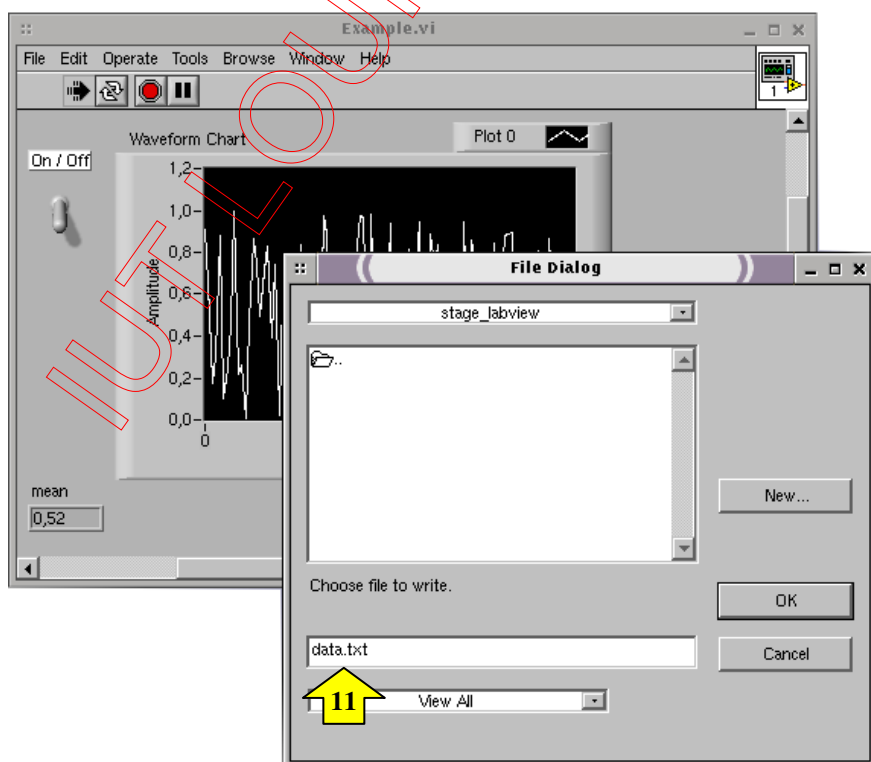
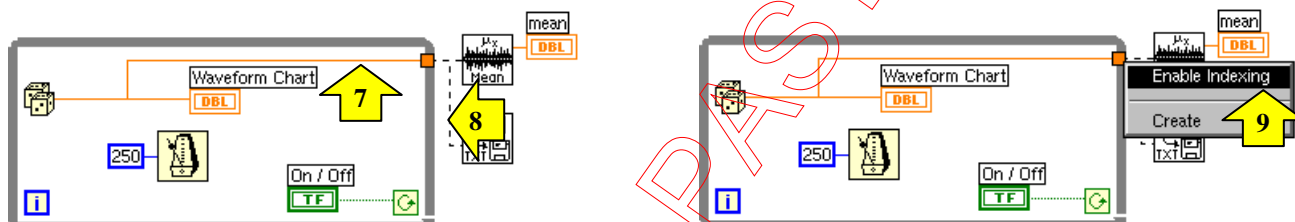
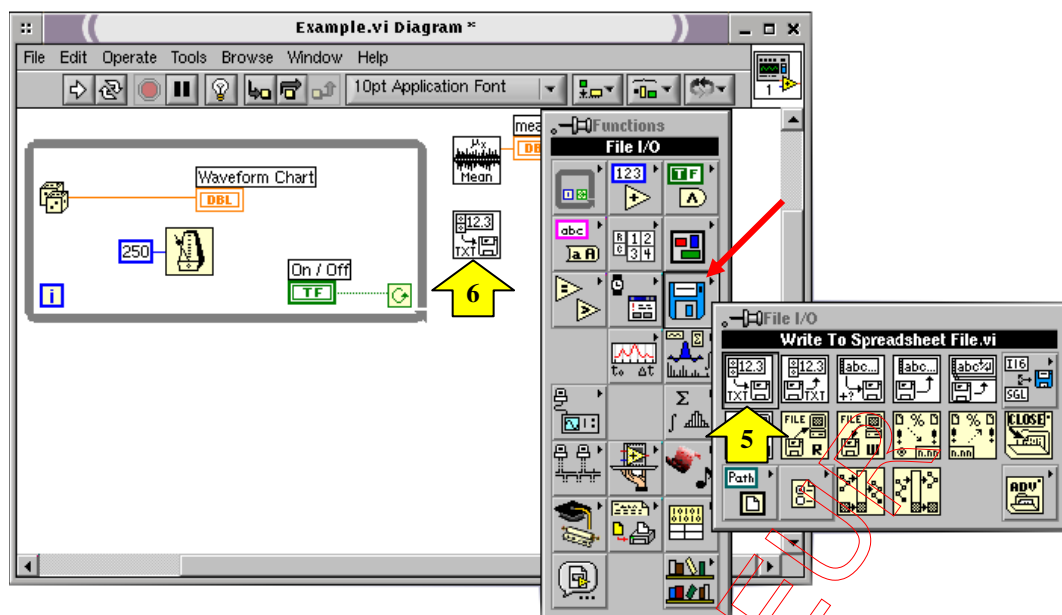
(10) Allez dans la face avant. Avec l'outil **Doigt**, placez l'interrupteur Marche/Arrêt en position haute (Vrai) et exécutez le VI.

(11) Lorsque vous cliquez sur l'interrupteur pour stopper le programme, vous voyez s'afficher la moyenne des données (dans l'indicateur numérique) et une boîte de dialogue vous demande le nom du fichier à créer. Tapez « data.txt » et enregistrez le fichier dans votre répertoire courant, c'est à dire E:/votre\_département/votre\_répertoire/data.txt.

Visualiser le contenu de ce fichier à l'aide d'un éditeur de texte, le Bloc-Notes par exemple en double cliquant sur votre fichier texte à partir de votre 'répertoire perso'.

*Conseil : la moyenne ne s'affiche pas tant que la collecte des données n'est pas terminée, c'est-à-dire lorsque vous cliquez sur l'interrupteur à bascule Marche/Arrêt en position basse (faux).*





**Exercice 2 : utilisation de variables booléennes**

Vous allez créer un VI qui compare deux nombres que l'on peut modifier à volonté et allume un voyant lumineux si ces nombres sont égaux.

(1) Ouvrez une nouvelle face avant en choisissant **Nouveau VI** dans le menu déroulant **Fichier**.

(2) Affichez les fenêtres face avant et diagramme.

(3) Créez 2 commandes numériques. Vous les utiliserez pour entrer les nombres que vous souhaitez comparer. **Commandes>Moderne>Numériques>Commande numérique** à placer sur la face avant

Taper un nom ou un numéro dans l'étiquette de chacune des commandes numériques (A et B par exemple), puis appuyez sur le bouton **Valider** de la barre d'outils de la fenêtre ou sur Entrée du pavé numérique.

Attention, **Entrée** (= retour chariot) du clavier alphabétique ajoute une ligne de texte alors que **Entrée** du pavé numérique valide la frappe.

(4) Créez le voyant lumineux booléen. Cet indicateur s'allumera lorsque les deux nombres seront égaux, et restera éteint dans le cas contraire. Poper dans la face avant et choisir **Commandes>Moderne>Booléen>Led circulaire**. Tapez "A=B" dans l'étiquette puis cliquez à l'extérieur de celle-ci lorsque vous avez fini ou **Entrée**.

Chaque fois que vous créez une commande ou un indicateur sur la face avant, Labview crée automatiquement le terminal correspondant dans le diagramme. La représentation du terminal rappelle le type de données de la commande ou de l'indicateur. Ainsi, un terminal DBL représente un nombre à virgule flottante en double précision alors qu'un terminal TF représente une variable booléenne (True/False).

Notez que les bordures d'un terminal de commande sont plus épaisses que celles d'un terminal d'indicateur.

(5) Activez la fenêtre du diagramme et sélectionnez la fonction **Fonctions>Programmation>Comparaison>Egaux?**. La fonction **Egaux?** compare deux nombres et indique Vrai s'ils sont égaux ou Faux dans le cas contraire.

(6) Avec l'outil **Bobine**, câblez les icônes.

Pour mettre en place un fil de liaison entre deux terminaux, cliquez sur le premier terminal avec l'outil **Bobine**, déplacez l'outil jusqu'au second, puis cliquez dessus. Vous pouvez câbler indifféremment dans les deux sens. Pour faciliter le câblage, on peut ouvrir un menu local sur la fonction **Egaux?** et choisir **Eléments visibles>Terminaux**.

(7) Dans le diagramme, poper sur les deux terminaux et choisir **Représentation>I16 (Mot)**. Que remarque-t-on ?

(8) Activez la face avant et entrez des nombres dans les commandes numériques, soit en cliquant dans la boîte et en tapant les chiffres au clavier, soit en agissant sur les petites flèches à l'aide de l'outil **Doigt**.

(9) Lancez le VI. Essayez plusieurs nombres différents.

(10) Activez l'**Exécution permanente** et modifiez les valeurs de vos indicateurs.

**Sauvegarder ce VI : exo\_02.vi**

**Exercice 3 : registre à décalage**

Les registres à décalage (disponibles dans les boucles While et For) sont des variables locales qui transfèrent des données entre deux itérations consécutives d'une même boucle.

Objectif : câbler la suite numérique suivante et déterminer sa limite quand  $n \rightarrow \infty$ .

$$\text{On a : } U_n = \frac{1}{1 + U_{n-1}} \text{ avec } U_0 = 0.$$

(1) Inclure dans le diagramme d'un nouveau VI une boucle **For**. Poper sur le bord droit de cette boucle et ajouter un registre à décalage (**Ajouter un registre à décalage**).

(2) Inclure dans la boucle **For** un opérateur "+", une constante numérique égale à 1 et un opérateur "1/x". Câbler ces éléments et relier la sortie de l'opérateur "1/x" au registre à décalage à droite de la boucle **For**. Poper sur le fil reliant 1/x au registre et ajouter un indicateur numérique (**Créer>Indicateur**) que vous nommerez "U<sub>n</sub>". Câbler le registre à décalage à gauche de la boucle **For** sur l'opérateur "+". Créer un indicateur numérique nommé "U<sub>n-1</sub>" à la sortie du registre côté gauche de la boucle.

(3) Avec la bobine, poper sur le registre à décalage à gauche de la boucle **For** et **Créer>Constante** (c'est U<sub>0</sub>).

(4) Dans la face avant, inclure un graphe déroulant. Dans la fenêtre diagramme, on le positionnera dans la boucle **For**. Câblez la sortie de l'opérateur "1/x" à ce graphe déroulant.

(5) Dans le diagramme, poper sur **N** et créer une commande (**Créer>Commande**), poper sur le "i" bleu (dans le carré bleu) et créer un indicateur. Ajouter une temporisation (**Attendre un multiple de ms**) dans la boucle **For**. Poper sur son entrée et **Créer>Commande**.

(6) Organiser la face avant, renommer au besoin les différents contrôles et indicateurs. On prendra au départ les valeurs suivantes : Tempo = 1000 ms = 1 seconde, U<sub>0</sub> = 0, N = 10.

(7) Exécuter le VI. Jouer sur les valeurs de N et U<sub>0</sub>. Vers quelle valeur tend cette suite ? On pourra au besoin changer le format d'affichage de l'indicateur numérique en popant et choisissant **Format et précision**.

Rq :

- il est possible d'utiliser autant de registres à décalage que nécessaire dans la même boucle sans que les données contenues dans ces registres ne se mélangent.
- il est également possible d'avoir un registre à plusieurs décalages en popant sur le registre à décalage de droite et en choisissant **ajouter un élément**.

(8) Lancer le VI et constater que la valeur affichée par l'indicateur "U<sub>n</sub>" se retrouve dans l'indicateur "U<sub>n-1</sub>" lors de la boucle suivante.

**Sauvegarder ce VI : exo\_03.vi**

**Exercice 4 : structure Condition**

Nous allons perfectionner l'exemple de l'exercice 2, ce qui va nous permettre de découvrir la structure **Condition**.

L'objectif : les valeurs de 2 générateurs aléatoires vont être multipliées par 10 ou par 100, arrondies et comparées. 3 LED de couleurs différentes indiqueront l'état de la comparaison :  $>$ ,  $<$  ou  $=$ . Un graphe déroulant ainsi que des indicateurs numériques permettront de visualiser les valeurs et des compteurs permettront de faire des statistiques : combien de fois (ou pourcentage) a-t-on eu  $A > B$ ,  $A < B$  ou  $A = B$ .

- (1) Créer une grande boucle **For** dans un diagramme vierge. Y inclure 2 **Nombre aléatoire (0-1)** ainsi qu'une structure **Condition** (les générateurs à l'extérieur de la condition).
- (2) Dans la structure **Condition (Faux)**, inclure deux opérateurs **Multiplication** et une constante numérique dans laquelle on entrera la valeur 10.
- (3) Câbler chacun des deux générateurs aléatoires sur une entrée des opérateurs **Multiplication**. Sur chacune des deux autres entrées, on câblera la constante 10.
- (4) Changer la condition de la structure **Condition (Vrai)** et répéter les étapes (2) et (3) en remplaçant la valeur 10 par 100.
- (5) A l'extérieur de la condition, inclure deux opérateurs **Arrondir à l'entier le plus proche**. Pour chacune des deux conditions (Faux et Vrai), reliez la sortie de chacun des deux produits à une fonction d'arrondi.
- (6) Ajouter les 3 opérateurs de comparaison, chacun relié à une LED (rouge, verte, et jaune) importée en face avant.
- (7) Câbler les deux fonctions d'arrondi sur les trois comparateurs en respectant le même ordre de branchement sur les trois opérateurs.
- (8) Poper sur la sortie des deux '**Arrondir...**' et ajouter un indicateur. Afficher les valeurs sur un graphe déroulant bi courbe (voir 'Annexe : les principaux afficheurs de courbes', pages 40-41).
- (9) Câbler chacun des 3 opérateurs sur une fonction de conversion booléen→nombre **Fonctions>Programmation>Booléen>Booléen en (0-1)**). Câbler la sortie de chaque opérateur de conversion sur un opérateur "+". Poper sur le bord droit de la boucle **For** et créer 3 registres à décalage distincts. Câbler la sortie de chaque opérateur "+" sur un registre à décalage (bord droit). Sommer la sortie de chaque opérateur de conversion avec la valeur correspondante issue de la boucle précédente (registre à décalage bord gauche). Ne pas oublier d'initialiser les trois registres (bord gauche) en popant et créant une constante égale à zéro.
- (10) Poper sur les 3 fils reliant la sortie des opérateurs "+" et créer 3 indicateurs. On pourrait même ajouter des indications en pourcentage...Comment pourriez vous faire (2 méthodes possibles, ne pas faire le câblage associé) ?
- (11) Changer le mot "Faux" de la condition par "Coeff 10", "Vrai" par "Coeff 100".
- (12) Dans la face avant, inclure un menu énumération : **Commandes>Moderne>Menu déroulant et énum>Enum**. Taper " Coeff 10" (outils **texte**) dans le cadre vide au centre de l'objet. Poper au centre de l'objet et **Ajouter un élément après**. Dans le nouveau cadre vide, taper "Coeff 100".

(13) Dans le diagramme, câbler le "?" qui est le terminal de sélection de la structure **Condition** à la boîte **Enum**.

(14) Poper sur **N** et **Créer>Constante**. Taper 100.

(15) Dans le diagramme, ajouter une temporisation, poper sur son entrée et créer une commande.

(16) Compléter la face avant en nommant judicieusement les indicateurs et lancer le VI en choisissant le cas Coeff 10 ou Coeff 100 (puis changer) avec une temporisation lente (1000 ms) et un petit nombre de boucle.

Relancer le VI en changeant les paramètres. Observer les indications en pourcentage concernant l'égalité des tirages. Dans quelle situation est-il le plus grand : dans le cas où le facteur de multiplication est 10 ou 100 ?

**Sauvegarder ce VI : exo\_04.vi**

### Exercice 5 : utilisation de la structure Séquence

La structure **Séquence** exécute des diagrammes de manière séquentielle. Dans les langages de programmation classique, les instructions de programmes s'exécutent dans l'ordre où elles sont écrites, alors que dans la programmation par flux de données, une boîte de calcul s'exécute lorsque les données sont disponibles sur toutes ses entrées. Cependant, il est parfois nécessaire d'exécuter une boîte de calcul particulière avant une autre. Ceci peut être réalisé avec la structure **Séquence** qui exécute le bout de programme contenu dans le cadre 0 avant celui qui se trouve dans le cadre 1 et ainsi de suite. Il faut que le premier cadre ait fini son exécution pour que le second démarre et ainsi de suite.

(1) Dans un nouveau VI, inclure une séquence : **Fonctions>Programmation>Structures>Structure séquence empilée**.

Dans le cadre inclure une boucle **For** (N=10) dans laquelle on placera une temporisation de 500 ms.

(2) Poper sur **"i"** et **Créer>Indicateur**. Poper sur le cadre de la séquence et choisir **d'ajouter une étape après**. Dans la nouvelle étape, inclure une boucle **While** dans laquelle vous placerez un générateur de nombres aléatoires associé à un graphe déroulant (en face avant) ainsi qu'une temporisation de 10 ms.

(3) Dans la face avant, inclure une commande booléenne (interrupteur ou bouton) que vous câblerez, dans le diagramme, à la condition d'arrêt de la boucle **While**.

(4) Réorganiser la face avant et nommer les différents indicateurs ou terminaux de commande.

(5) Lancer le VI, le but ici étant uniquement de montrer que le tirage des valeurs aléatoires de la deuxième séquence ne démarre que lorsque la boucle **For** du premier cadre de la séquence est finie.

**Sauvegarder ce VI : exo\_05.vi**



**Exercice 6 : utilisation d'une "Boîte de calcul"**

La boîte de calcul est un outil qui permet de calculer des équations mathématiques sans avoir à les programmer de façon graphique. Ceci est un avantage si les équations sont compliquées.

(1) Dans un nouveau VI, inclure une boucle **For** dans laquelle on placera une **Boîte de calcul** : **Fonctions>Programmation>Structures>Boîte de calcul**.

Taper à l'intérieur de celle-ci l'équation suivante :  $y = x^{**2} - 2*x + 3;$ .

( $x^2$  s'écrit  $x^{**2}$  ici, attention, le « ; » fait partie de l'instruction...)

(2) Poper sur le cadre de gauche de la **Boîte de calcul** et choisir **Ajouter une entrée**. Taper x dans cette case. Poper sur le cadre de droite et **Ajouter une sortie**. Taper y dans cette nouvelle case.

(3) Câbler la sortie y sur un graphe déroulant et ajouter les éléments nécessaires pour faire varier x de -50 à +50 par pas de 1 et lancer le VI. Comment pourrait-on faire pour faire varier x par pas de 0,1 ? Le faire...

On va maintenant tracer la courbe du gain en dB d'un filtre passe-bas du premier ordre de fréquence de coupure  $f_c$ .

La fonction de transfert harmonique d'un tel filtre peut s'écrire :  $H(f) = \frac{1}{1 + j \frac{f}{f_c}}$  ce qui donne  $|H(f)| = \frac{1}{\sqrt{1 + \left(\frac{f}{f_c}\right)^2}}$

Le gain en décibels s'écrit :  $G = 20 \cdot \log(|H(f)|)$

(4) A l'aide d'une nouvelle boucle **For**, réaliser et afficher les équations suivantes pour f variant de 0 à 1000 par pas de 1, avec  $f_c = 10$  Hz :

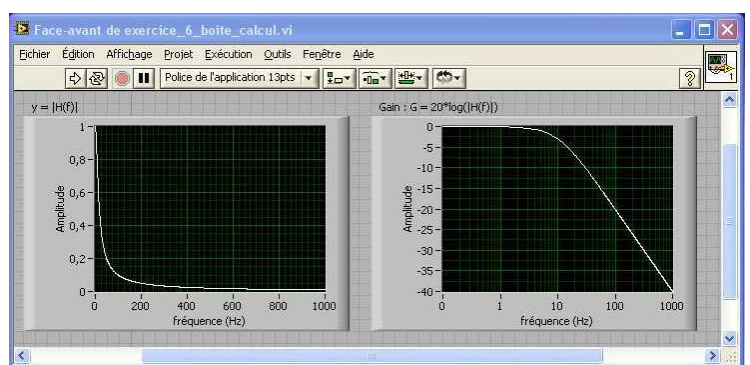
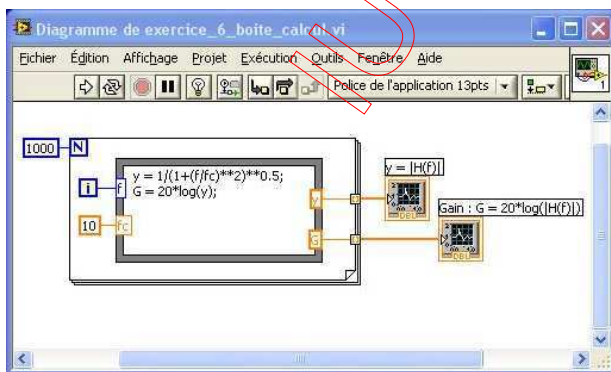
$y = 1 / (1 + (f/f_c)^{**2})^{**0.5};$  (y correspond à  $|H(f)|$ , attention c'est bien 0.5 et non pas 0,5)

$G = 20 * \log(y);$  (le « ; » est le séparateur des équations dans la boîte de calcul)

Prévoir les entrées/sorties nécessaires. On visualisera la fonction y sur un graphe et G sur un autre graphe. (Voir images ci-dessous)

(5) Poper sur le graphe du gain et choisir **Formatage> Onglet « Echelles »>échelle des X>logarithmique**.

Sauvegarder ce VI : **exo\_06.vi**



**Exercice 7 : création d'un tableau 1D à l'aide de la boucle For**

Nous allons apprendre dans cet exercice à tracer la fonction  $\sin(x)$  sur une période, avec 100 points et écrire les points calculés dans un tableau.

On a :  $y = \sin(2\pi i/100)$  avec  $i$  variant de 0 à 99 et donc  $x = 2\pi i/100$ .

(1) Ouvrir un nouveau VI.

(2) Dans le diagramme, placer une boucle **For**.

(3) Dans ce cadre, placer la fonction sinus :

**Fonctions>Mathématiques>Fonctions élémentaires et spéciales>Fonctions Trigonométriques>Sinus.**

(4) Créer  $x = 2\pi i/100$ . La constante  $2\pi$  existe déjà dans **Fonctions>Mathématiques>Numériques>Constantes mathématiques et scientifiques>2 $\pi$**  et la variable "i" sera prise à partir du terminal donnant le numéro de la boucle **For** en cours d'exécution.

(5) Appliquer le résultat ( $2\pi i/100$ ) sur l'entrée du sinus.

(6) Attribuer la valeur 100 à N en popant sur N, **Créer>Constante**.

(7) Importer dans la face avant un graphe non déroulant **Commandes>Moderne>Graphe>Graphe**

(8) Dans la face avant importer un tableau : **Commandes>Moderne>Tableau, matrice et Cluster>Tableau**

(9) Dans ce tableau, au centre de la case grise, placer un indicateur numérique :

**Commandes>Moderne>Numérique>Indicateur numérique.**

Vous voyez apparaître à côté du tableau une case. Elle contient le n° de l'élément visualisé.

(10) Revenir sur le diagramme et trouver le tableau créé [DBL] ainsi que le terminal **Graphe**. Les sortir de la boucle s'ils s'y trouvent. Relier la sortie du sinus à l'entrée du tableau ainsi qu'au terminal **Graphe**. On remarque que l'épaisseur du fil change entre l'intérieur de la boucle et l'extérieur, car on passe d'un élément n°  $i$  au tableau complet.

(11) Lancer le VI, passer sur la face avant et observer le contenu du tableau.

*Pour visualiser un élément du tableau, il suffit d'entrer son n° dans l'indice.*

*Pour en visualiser plusieurs, avec l'outil **Flèche**, se placer sur le bord du tableau et l'agrandir horizontalement ou verticalement.*

**Sauvegarder ce VI : exo\_07.vi**

**Exercice 8 : créer un tableau 2D**

On peut souhaiter mettre dans un tableau non seulement les valeurs  $y$  calculées précédemment,  $y = \sin(2\pi i/100)$  mais aussi les valeurs de  $x$  correspondantes,  $x = 2\pi i/100$ .

On va alors créer un tableau à deux dimensions, la première ligne pour les  $x$ , la seconde pour les  $y$ .

(1) Ouvrir le VI `exo_07.vi`.

(2) Sur le diagramme, placer la fonction **Construire un tableau** à l'extérieur de la boucle :

**Fonctions>Programmation>Tableau>Construire un tableau**. Agrandissez le VI avec l'outil **Flèche** pour qu'il contienne 2 lignes (2 entrées, une pour les  $x$ , une pour les  $y$ ).

(3) Relier alors les  $x$  (sortie de l'opérateur division) et les  $y$  (sortie du sinus) aux entrées du tableau.

(4) Sur la sortie du tableau, créer un indicateur (clic sur cette sortie et **Créer>Indicateur**). Dans l'exercice précédent, le tableau a d'abord été créé sur la face avant, puis utilisé dans le diagramme. Ici, il est créé dans le diagramme et nous l'utilisons sur la face avant.

Vous remarquez que les données sont disponibles en sortie de boucle sous forme de tableau 2D = trait double.

(5) Lancer le VI et observer la face avant. Redimensionner le tableau 2D, pour observer les deux lignes de données.

*Sauvegarder ce VI : `exo_08.vi`*

**Exercice 9 : création d'un Graphe XY**

(1) Sur le diagramme précédent, à l'extérieur de la boucle, importer la fonction **Assembler** (=Bundle) :

**Fonctions>Programmation>Cluster et variant>Assembler**. Elle permet de préparer les données qui sortent de la boucle en un groupe de tableau prêt à l'affichage. Les deux variables d'entrée sont les  $x$  et les  $y$ . Connecter les  $x$  à l'entrée «  $x$  » de **Assembler**, et les  $y$  à l'entrée «  $y$  » de **Assembler**.

(2) Passer sur le panneau avant et importer un graphe XY : **Commandes>Moderne>Graphe>Graphe XY**.

(3) Sur le diagramme, connecter l'entrée de ce graphe à la sortie de **Assembler**.

(4) Tester, observer la face avant.

*Sauvegarder ce VI : `exo_09.vi`*

**Exercice 10 : écrire dans un fichier tableur un tableau 1 D**

Quelques rappels :

format ASCII (fichier texte) : toutes les données, et en particulier les nombres, sont convertis en chaînes de caractères. Les informations contenues dans un tel fichier peuvent être consultées ou créées à partir d'un éditeur. Les informations de type mesures sont aisément transférables à des logiciels de calculs statistiques, des tableurs ou des bases de données.

format binaire : les éléments d'un tel fichier sont représentés de la même façon que les données en mémoire. Un entier de type int occupera deux octets en mémoire. Les données ne peuvent être lues ou écrites que par programme. Ce format permet un stockage plus compact et plus rapide des données (extension courante: bin).

Pour résumer, un fichier ASCII (ou texte) est traduit en langage intelligible, un fichier binaire est stocké en langage brut. Les opérations sur un fichier binaire sont donc beaucoup plus rapides que sur un fichier texte (transfert, accès direct possible) et ce type de fichier doit être préféré pour le traitement de grandes quantités de données. Les fichiers au format ASCII restent cependant les plus courants.

Les fichiers tableurs : une façon très courante de sauvegarder des données dans un fichier consiste à formater le fichier texte de façon à pouvoir l'ouvrir avec un tableur. Dans la plupart des tableurs, des tabulations séparent les colonnes et des caractères de fin de ligne (EOL) séparent les lignes.

Pour Labview, les tableaux de données sont relatifs aux lignes : chaque ligne est un tableau. Dans un tableau, on utilise souvent les colonnes comme tableaux de données (les x dans la première, les y dans la seconde ... ). Une option de transposition à la lecture d'un fichier tableur est disponible.

(1) Ouvrir le VI `exo_07.vi`.

2) Importer dans le diagramme le VI qui permet d'écrire dans un fichier de type tableur :

**Fonction>Programmation>E/S sur fichiers>Ecrire dans un fichier tableur.**

Ce VI convertit le tableau en chaîne de caractères au format tableur et l'écrit dans le fichier. Il gère l'ouverture et la fermeture du fichier. Si vous ne donnez pas le nom du fichier, une boîte de dialogue apparaît au lancement et vous le demande. **Ecrire dans un fichier tableur** écrit un tableau à une ou deux dimension(s) dans un fichier. Dans notre exemple, le tableau final étant un tableau à 1 dimension, vous n'avez pas à câbler l'entrée 2 D.

(3) Relier la sortie du tableau à une dimension (tunnel de sortie de la boucle) à l'entrée 1D de **Ecrire dans un fichier tableur**.

(4) Lancer l'exécution du VI, donner un nom de fichier, puis observer le contenu du fichier grâce à un éditeur de texte, BlocNotes ou Wordpad.

N'aurait-on pas préféré avoir le tableau dans l'autre sens !? Quelle entrée sur le VI **Ecrire dans un fichier tableur** peut être bien utile ? A vous de jouer grâce à l'aide pour avoir le tableau en colonne lorsqu'on l'ouvre sous un éditeur (une contante booléenne se commute avec l'outil **doigt**).

**Sauvegarder ce VI : `exo_10.vi`**

**Exercice 11 : écrire dans un fichier tableur un tableau 2D**

- (1) Ouvrir le VI `exo_08.vi`
- (2) Dans le diagramme, importer **Ecrire dans un fichier tableur**. Dans cet exemple, la sortie de **Construire un tableau** est un tableau 2 D.
- (3) Donc connecter la sortie de **Construire un tableau** sur l'entrée 2D de **Ecrire dans un fichier tableur**.
- (4) Ne pas oublier de transposer les lignes et colonnes. Choisir un format d'écriture des données, "%+.5e" par exemple.
- (5) Exécuter le VI, entrer un nom de fichier et contrôler le résultat obtenu avec l'éditeur de texte.

**Sauvegarder ce VI : `exo_11.vi`**

**Exercice 12 : lecture d'un fichier de type tableur 1D**

- (1) Ouvrir un nouveau VI et importer dans le diagramme la fonction **Lire un fichier tableur**.
- (2) Afficher l'aide et repérer tous les terminaux.
- (3) Avec l'outil **bobine**, poper sur la sortie "**Toutes les lignes**" et créer un indicateur.
- (4) En ne câblant aucun des autres terminaux les paramètres par défauts seront utilisés. Lancer le VI et lire le fichier texte sauvegardé dans l'exercice 10. Observer le résultat.
- (5) Avec l'outil **bobine**, poper sur l'entrée "**Chemin du fichier**", créer une constante dans laquelle vous entrez le nom du fichier de l'exercice 10 avec le chemin complet : `E:\votre_département\votre_repertoire\votre_fichier.txt`

*attention, pour le chemin d'accès, Labview veut des \ et non pas des /...*

- (6) Dans la face avant, importer un **Graphe** (graphe non déroulant) et le câbler à la sortie "**Toutes les lignes**".
- (7) Lancer le VI : la boîte de dialogue ne s'ouvre plus et les données s'affichent sur le graphe.
- (8) Inclure dans la face avant un interrupteur booléen. Dans son étiquette, taper "transposer ?". Câbler cette commande au terminal correspondant et relancer le VI en agissant sur ce paramètre.
- (9) Dans le diagramme, poper sur le terminal "**Nombre de lignes**" avec l'outil **bobine** et créer une commande.
- (10) Lancer le VI plusieurs fois en agissant sur ce paramètre.

**Sauvegarder ce VI : `exo_12.vi`**

**Exercice 13 : lecture d'un fichier de type tableur 2D**

- (1) Ouvrir le VI `exo_12.vi` et le sauvegarder sous le nom `exo_13.vi`.
- (2) Remplacer le nom du fichier texte à ouvrir par celui généré dans l'exercice 11 (tableau 2D).
- (3) Prendre les paramètres par défaut et lancer le VI. Observer les données dans le tableau et dans le graphe.
- (4) On aurait préféré avoir sur le graphe les données de la deuxième colonne en fonction de celles de la première... modifier le diagramme pour afficher la bonne courbe.

***Sauvegarder ce VI : `exo_13.vi`***

IUT LOUIS PASTEUR

**Exercice 14 : Utilisation d'une variable locale**

- (1) Créer une séquence à 2 étapes : **Fonctions>Programmation>Structures>Structure séquence empilée**
  - (2) Dans l'étape 0, inclure une boucle **For** ( $N = 100$ ) dans laquelle on placera un générateur de nombres aléatoires **Nombre aléatoire (0 - 1)**.
  - (3) Poper sur le bord droit de la séquence et choisir **Ajouter une variable locale de Séquence**. Un terminal apparaît. Il permet le passage de paramètres d'une étape à la suivante d'une séquence.
  - (4) Relier la sortie **Nombre aléatoire** à ce nouveau terminal. Une flèche orange y apparaît, donnant le sens du flux de données.
  - (5) Dans l'étape 1, le sens de la flèche du terminal indique que les données proviennent de l'étape précédente vers l'étape actuelle. Inclure un VI **Moyenne**. Relier son entrée (à gauche) au terminal de la séquence.
  - (6) Poper sur le terminal de sortie du VI **Moyenne** et **Créer>Indicateur**.
  - (7) Exécuter le programme plusieurs fois. L'indicateur numérique donne la moyenne, dans l'étape 1, des 100 tirages effectués dans l'étape 0. On voit que le tableau de données issu de la boucle **For** de l'étape 0 est passé au VI **Moyenne** de l'étape 1.
- Une variable doit parfois servir en plusieurs endroits du programme. Pour éviter un enchevêtrement de fils trop compliqué, on peut créer une image de cette variable, que l'on peut ainsi utiliser dans d'autres parties du programme en minimisant la connectique. Une variable locale peut être utilisée en lecture (on lit ou récupère la valeur de la variable) ou en écriture (on modifie la valeur de la variable). Les opérations suivantes décrivent la façon de procéder pour créer et utiliser une variable locale.
- (8) Dans l'étape 0 et à l'aide de la **Bobine**, poper sur le fil de liaison, à l'extérieur de la boucle **For**, entre le générateur de nombres aléatoires et le terminal de la séquence : **Créer>Indicateur**. Un terminal de type **Tableau** se connecte automatiquement sur le fil. Changer le nom du tableau, par exemple 'Tirages'.
  - (9) Poper sur ce tableau et choisir **Créer> Variable locale**. Une variable locale est alors créée, qui peut être utilisée ailleurs dans le programme. Le bord fin de la variable indique qu'elle est configurée en écriture. Une variable locale utilisée en lecture possède un bord épais.
  - (10) A l'aide de l'outil **Flèche**, sélectionner cette variable et la positionner à l'intérieur de l'étape 1 de la séquence. Poper sur la variable et choisir **Changer en lecture**. Noter la modification de la bordure (fine ↔ épaisse, écriture ↔ lecture).
  - (11) Inclure dans la face avant un **Graphe**.
  - (12) Dans la fenêtre diagramme, placer le terminal de ce graphe dans l'étape 1 de la séquence et le relier à la variable locale 'Tirages'.
  - (13) Lancer l'exécution du programme. Que constatez-vous ?

**Sauvegarder ce VI : exo\_14.vi**

**Exercice 15 : Utilisation des propriétés d'une variable : Nœud de Propriété**

(1) Ouvrir un nouveau VI. Dans la fenêtre diagramme, inclure un VI de génération d'une onde sinusoïdale :

**Fonctions>Traitement du signal>Génération de signaux>Signal sinusoïdal**

(3) En face avant, inclure un **Graphe**. Dans la fenêtre diagramme, avec l'outil **bobine**, connecter ces deux éléments entre eux

(4) Dans la fenêtre diagramme, cliquer sur le terminal **Graphe** et choisir **Créer>Nœud de Propriété>Visible**.

*L'icône qui apparaît gère tous les paramètres d'affichage du terminal **Graphe**. Presque tous les objets possèdent certaines propriétés auxquelles on peut accéder par l'intermédiaire du **Nœud de Propriété**. Ces paramètres sont accessibles en mode lecture et/ou écriture.*

(5) Cliquer sur la case **Visible** et choisir **Changer en écriture**. Noter la modification de la petite flèche dans la case.

(6) A l'aide de l'outil **bobine**, cliquer sur ce terminal **Visible** et choisir **Créer>Commande**.

(7) En face avant, lancer l'exécution permanente du programme et agir sur l'interrupteur 'Visible'. Que constatez-vous ?

(8) Fenêtre diagramme : à l'aide de l'outil **Flèche**, cliquer sur le **Nœud de Propriété** du **Graphe** et choisir **ajouter un élément**. Cliquer sur ce nouvel élément et choisir : **Propriétés>Echelle des X>Gamme>Maximum**.

(9) Avec l'outil **bobine**, cliquer sur la case **EchelleX.Maximum** et **Créer>Commande**.

(10) En face avant, cliquer sur le graphe et ~~désélectionner les options~~ **Echelle des X>Mise à l'échelle automatique des X**. (idem pour les Y).

(11) Lancer l'exécution permanente de ce programme en modifiant plusieurs fois le paramètre **EchelleX.Maximum**.

(12) Ajouter encore un élément au **Nœud de Propriété**. Changer le nom de cet élément en cliquant dessus et en choisissant : **Propriétés>Curseur>Position du Curseur>Tous les éléments**.

(13) Cliquer sur cet élément et choisir **Changer en lecture**. *Cette fois, la variable est utilisée en mode lecture.*

(14) Avec l'outil **bobine**, cliquer sur cet élément et **Créer >Indicateur**.

(15) En face avant, cliquer sur le graphe et choisir **Éléments visibles>Légende des Curseurs**. A l'aide de l'outil **doigt**, cliquer dans le cadre blanc de **curseur** et choisir **créer un curseur>courbe unique** (= curseur lié à la courbe).

(16) En face avant, lancer l'exécution du programme plusieurs fois en déplaçant le curseur en X à l'aide de l'outil **doigt**.

**Sauvegarder ce VI : exo\_15.vi**

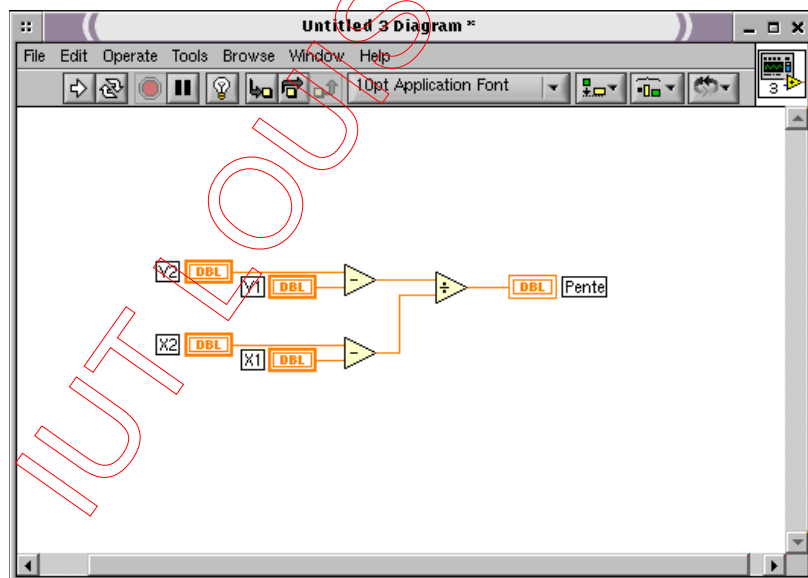
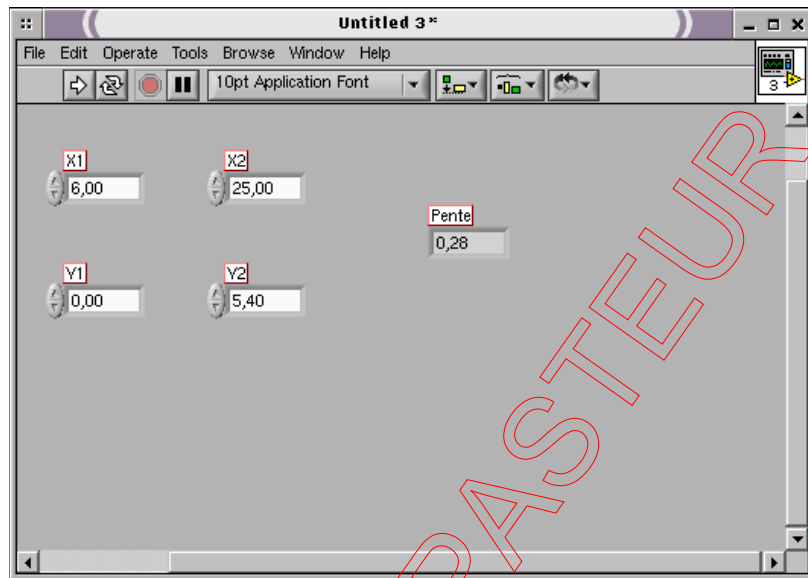


**Exercice 16 : Créer un sous-VI**

Une fois créé, un VI peut s'utiliser comme sous-VI dans le diagramme d'un autre VI. Cette approche modulaire facilite la mise au point, la compréhension et la maintenance des applications.

Les sous-VIs correspondent aux fonctions ou aux sous-programmes d'un langage de programmation traditionnel.

Prenons comme exemple un VI qui calcule la pente entre deux points d'après leurs coordonnées. La face avant et le diagramme correspondant sont donnés ci dessous. Câblez ce VI et enregistrez le sous le nom "pente.vi" dans votre répertoire.



Un VI utilisé comme sous-VI a besoin d'une icône qui le représentera dans le diagramme du VI appelant. Il doit aussi avoir un connecteur avec des terminaux pour pouvoir acheminer les données vers ou en provenance d'autres VIs.

La création d'un sous-VI passe donc par plusieurs étapes :

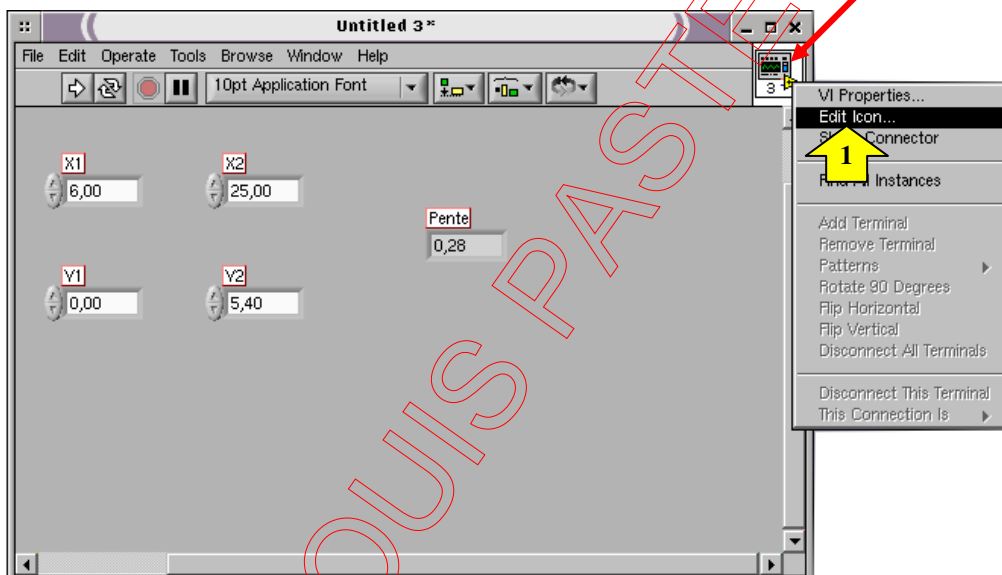
la création de l'icône et du connecteur

l'affectation des terminaux aux commandes et aux indicateurs

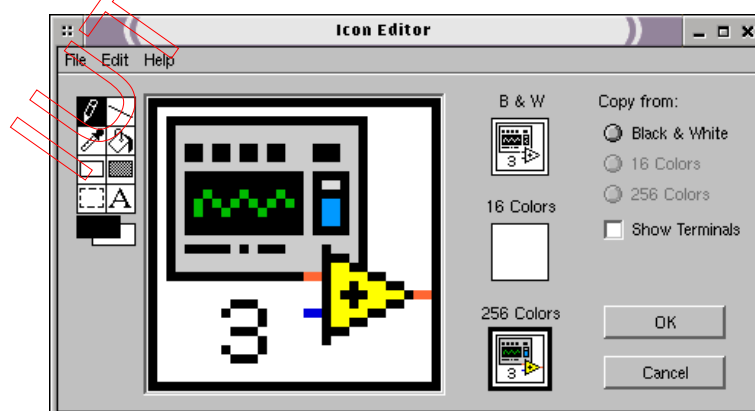
### (1) Réalisation de l'icône

Chaque VI dispose d'une icône par défaut affichée en haut à droite des fenêtres de la face avant et du diagramme. Pour les VIs, l'icône par défaut est le logo de LabView et un numéro indiquant le nombre de nouveaux VIs que vous avez ouverts depuis le démarrage de LabView. Avec l'**Editeur d'icônes**, vous pouvez personnaliser l'icône en activant et en désactivant les pixels un à un. Pour activer cet éditeur, ouvrez le menu local de l'icône par défaut en haut à droite de la fenêtre de la face avant et choisissez **Editer l'icône** comme ci-dessous.

Editer l'icône.



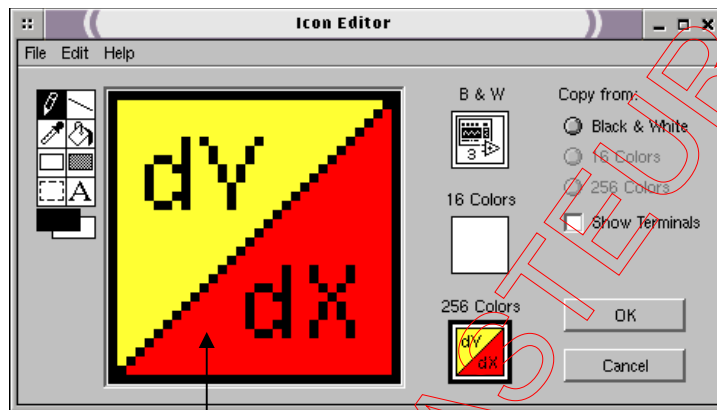
La fenêtre suivante apparaît.



Avec les outils situés à gauche, vous pouvez créer le dessin de votre icône dans la zone d'édition où chaque pixel est représenté par un gros point. L'icône apparaît dans sa taille réelle dans l'une des boîtes situées à droite de la zone d'édition.

En fonction du type de moniteur que vous utilisez, vous pouvez concevoir votre icône en mode noir et blanc, 16 couleurs ou 256 couleurs. Chaque version de l'icône sera dessinée et enregistrée séparément. Par défaut, l'éditeur se met en mode **Noir & blanc**, mais vous pouvez aussi changer de mode en cliquant sur les autres options de couleur. Dans tous les cas, créez toujours une image noir et blanc dans un souci de portabilité.

Modifier l'icône de façon à ce qu'elle corresponde à la fonction qu'elle réalise, par exemple :



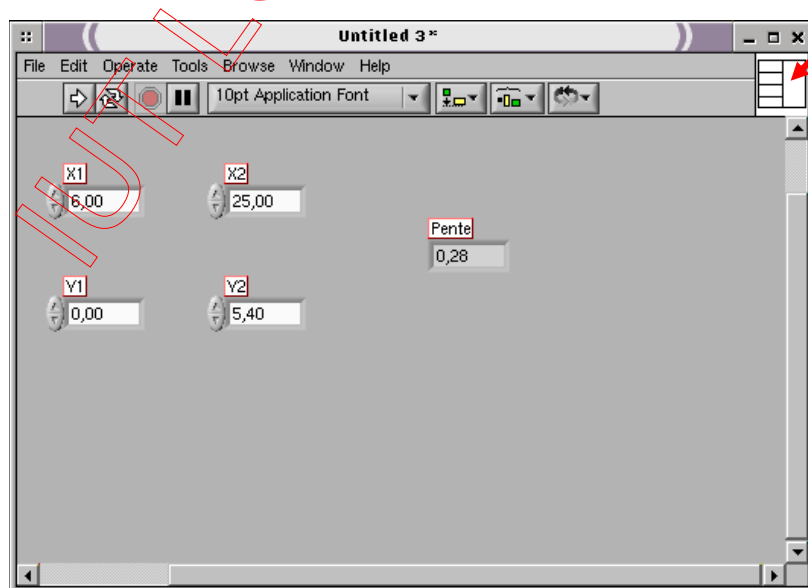
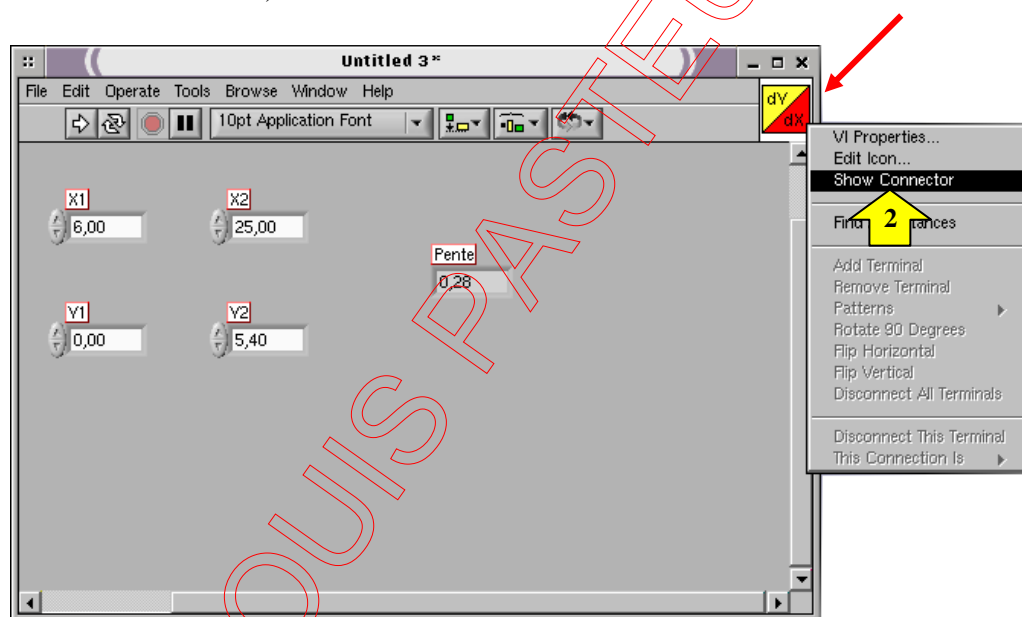
$$\frac{dy}{dx} = \frac{dy}{dx} = \text{pente} \dots$$

## (2) Le connecteur

Dans un VI, le connecteur est une interface de liaison inter-VIs. Si vous utilisez les commandes et les indicateurs de la face avant pour acheminer les données entre sous-VIs, les terminaux correspondants doivent être placés sur le cadre connecteur. Vous définissez les connexions en choisissant le nombre de terminaux nécessaires au VI, et en affectant une commande ou un indicateur à chacun de ces terminaux.

Pour définir un connecteur, ouvrez son menu local en cliquant sur le cadre icône (sur la face avant) et choisissez **Visualiser le Connecteur**. La fenêtre du diagramme n'a pas de cadre connecteur.

Le connecteur remplace l'icône en haut à droite de la fenêtre de la face avant. LabView choisit le modèle de terminal qui convient le mieux à votre VI, avec des commandes sur le côté gauche du cadre connecteur et des indicateurs à droite. Le nombre de terminaux sélectionnés dépend du nombre de commandes et d'indicateurs présents sur votre face avant. (On peut accessoirement ajouter ou enlever des terminaux)

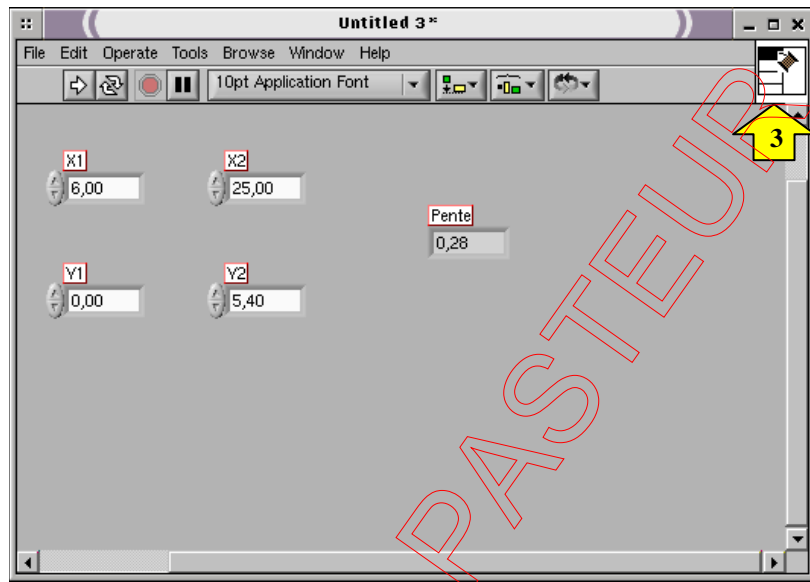


Chaque rectangle présent dans le connecteur symbolise un terminal. Vous pouvez utiliser ces rectangles aussi bien comme entrée que comme sortie du VI. Vous pouvez aussi, le cas échéant, sélectionner une autre forme de terminal pour votre VI.

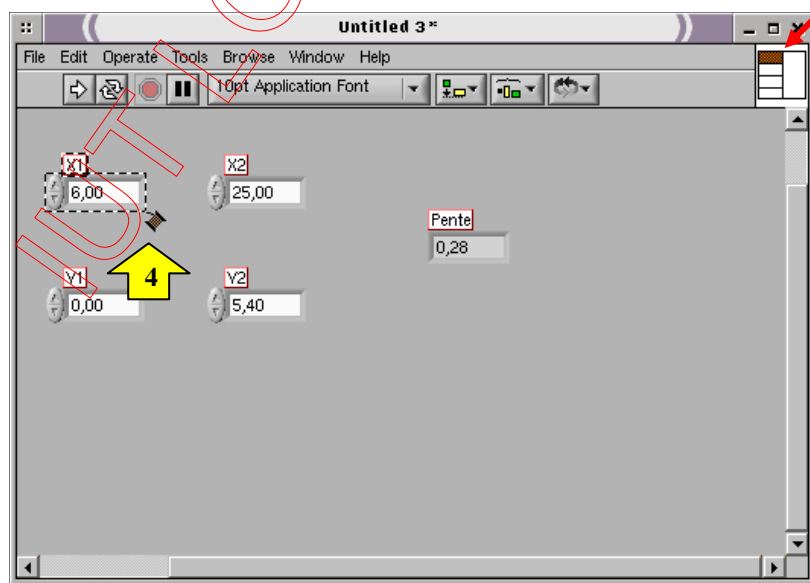
*rappel : les terminaux d'entrée sont en principe à gauche du vi et les terminaux de sortie à droite...*

### (3) Affectation des terminaux aux commandes et aux indicateurs

L'affectation des commandes et des indicateurs de la face avant aux terminaux se fait au moyen de l'outil **Bobine**. Cliquez sur un terminal du connecteur. Le terminal s'assombrit.



(4) Cliquez sur la commande ou l'indicateur que vous voulez affecter au terminal sélectionné. Une marqueuse (pointillés) entoure alors celui-ci.



Si votre curseur se trouve dans une zone libre de la fenêtre de la face avant, cliquez sur le bouton de la souris et vous verrez que la ligne en tirets qui entourait la commande ou l'indicateur disparaît et que le terminal sélectionné s'éclaircit. C'est bien la preuve que la commande (ou l'indicateur) que vous avez choisie correspond au terminal.

*Remarque : bien que vous utilisiez l'outil Bobine pour attribuer des terminaux du connecteur aux commandes et indicateurs de la face avant, vous noterez l'absence de fil sur la face avant.*

Répétez les 2 étapes précédentes pour chaque commande et indicateur que vous voulez connecter.

Vous pouvez également sélectionner la commande ou l'indicateur avant le terminal. Vous pouvez utiliser une forme de connecteur présentant plus de terminaux que nécessaire. Les terminaux non utilisés ne perturbent en rien le fonctionnement du VI. Inversement, vous pouvez aussi avoir plus de commandes et indicateurs que de terminaux.

#### (5) *Enregistrer votre VI : pente.vi dans votre répertoire*

Maintenant, vous pouvez utiliser ce VI comme sous-VI dans n'importe quel diagramme.

*Remarque : il est possible de créer une aide détaillée pour les sous-VI..*

### Exercice 17 : Utilisation d'un sous-VI

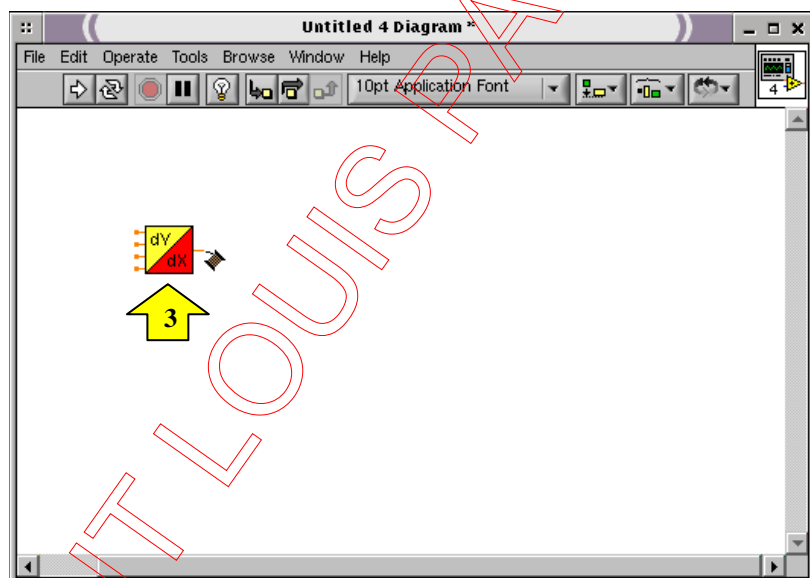
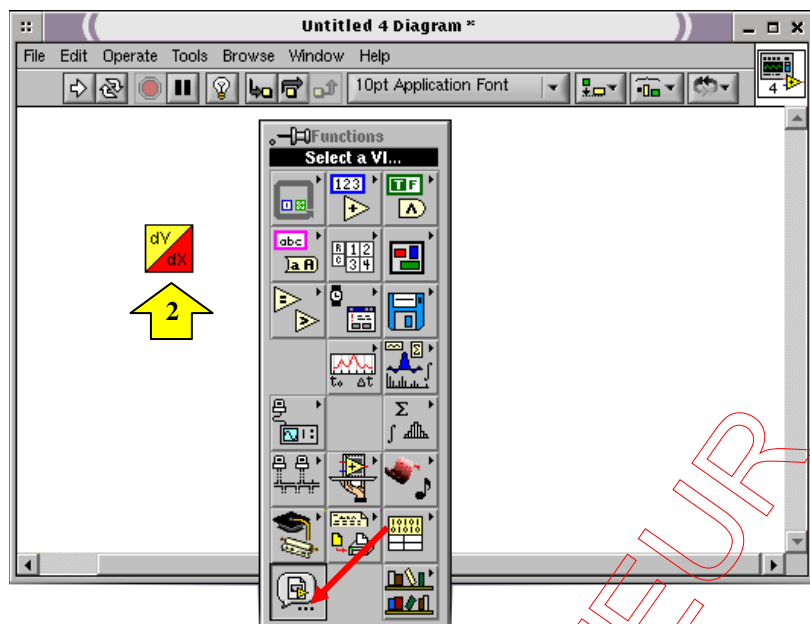
- (1) Ouvrir un nouveau VI et inclure dans le diagramme une boucle **For** dans laquelle on placera une temporisation variable.
- (2) Inclure dans la boucle **For** le VI pente.vi : **Fonctions>Sélectionner un VI> pente.vi**
- (3) Afficher l'aide et promener l'outil bobine sur l'icône de ce sous-VI. Câbler les entrées X1 et Y1 sur une constante égale à zéro.
- (4) Câbler l'entrée X2 sur une constante égale à 10. Relier l'entrée Y2 à "i". Le point [X1,Y1] sera donc fixe et seule l'abscisse Y2 du deuxième point sera variable.
- (5) Poper sur N et lui affecter la valeur 10. Poper sur la sortie du sous-VI **pente** et créer un indicateur.
- (6) Organiser la face avant, choisir une temporisation convenable (1000 ms par exemple) et lancer ce nouveau VI.

**Sauvegarder ce VI : exo\_17.vi**

*Remarque :*

*En double-cliquant sur l'icône du sous-VI, vous ouvrez sa face avant. Vous pouvez alors ouvrir la fenêtre du diagramme du sous-VI en choisissant **Afficher le Diagramme** dans le menu déroulant **Fenêtre**.*

*Tant que vous n'enregistrez rien, les modifications effectuées sur le sous-VI n'affectent que sa copie utilisée en mémoire. Sachez que ces modifications, une fois enregistrées, affectent tous les appels de ce sous-VI.*

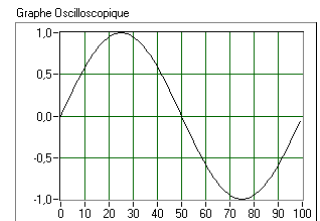


### III - Annexe : Les principaux afficheurs de courbes

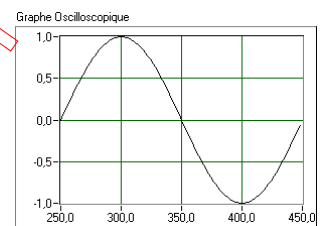
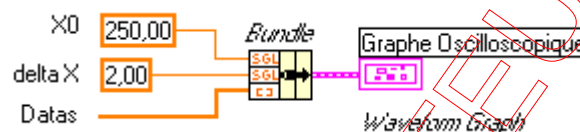
#### Graphe (graphe NON déroulant)

##### Graphes mono-courbe

L'indicateur **Graphe** (*waveform graph*) permet de tracer des courbes à la manière d'un oscilloscope. La courbe à tracer est envoyée en une seule fois sous la forme d'un tableau de données 1-D (indiqué par un fil gras). Par défaut, la courbe sera tracée à parti de  $X_0 = 0$  et l'incrément Delta X entre chaque point de la courbe sera de 1.

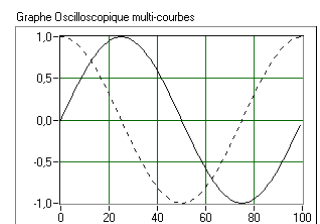
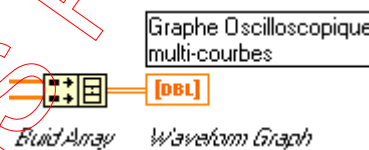


Il est possible de changer  $X_0$  et Delta X à l'aide de la fonction **Assembler** (Bundle), disponible dans le menu **Cluster et variant** de la palette **Fonctions>Programmation** en respectant l'ordre des paramètres.

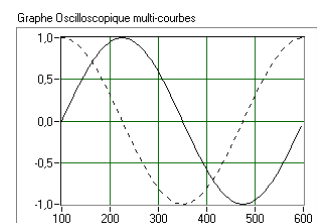
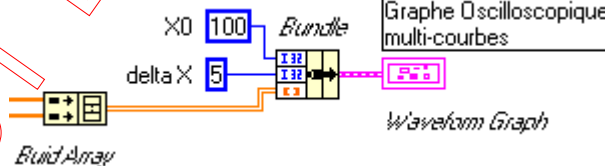


##### Graphes multi-courbes

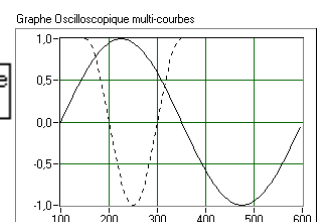
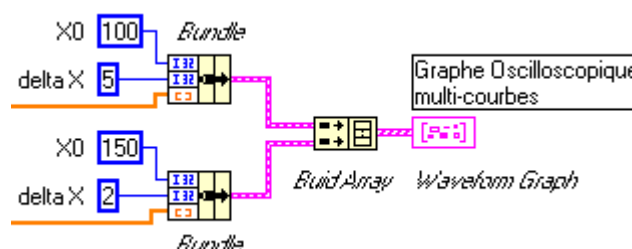
Pour avoir plusieurs courbes sur un seul graphe, il faut les assembler à l'aide de la fonction **Construire un tableau** (build Array) du menu **Tableau** de la palette **Fonctions>Programmation**.



Il est possible de spécifier un  $X_0$  et un Delta X pour les multiples courbes à l'aide de la fonction **Assembler** (Bundle) du menu **Cluster et variant** de la palette **Fonctions>Programmation**.



Il est également possible de spécifier un  $X_0$  et un Delta X différents pour les différentes courbes en agissant donc avant la fonction **Construire un tableau** (Build Array) du menu **Tableau** de la palette **Fonctions>Programmation**.

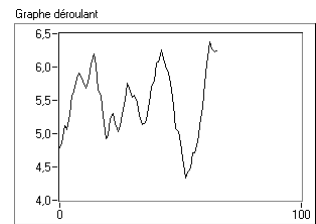




## Graphe déroulant

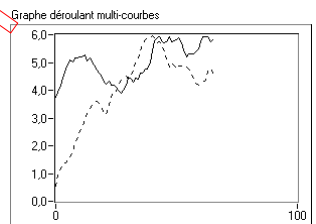
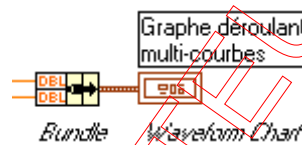
### Graphe mono-courbe

L'indicateur **Graphe déroulant** (Waveform Chart) permet de tracer des courbes comme sur un enregistreur papier à défilement. Les données (de type scalaire, indiqué par un trait fin) sont envoyées à l'afficheur au fur et à mesure, l'une après l'autre.



### Graphe multi-courbes

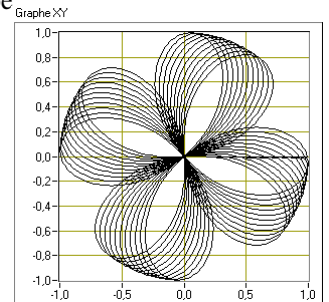
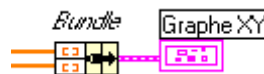
Ici aussi, pour afficher plusieurs courbes à la fois sur un même graphe déroulant, il faut les assembler à l'aide de la fonction **Assembler** (Bundle) du menu **Cluster et variant** de la palette **Fonctions>Programmation**.



## Graphe XY

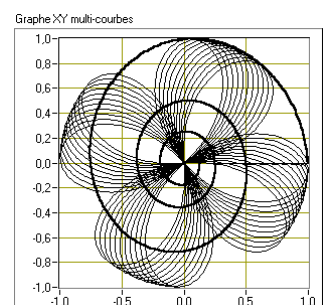
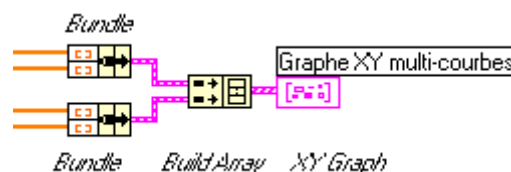
### Graphe mono-courbe

L'indicateur **Graphe XY** permet de tracer des courbes paramétriques à la manière d'un oscilloscope. La courbe à tracer est envoyée en une seule fois sous la forme de deux tableaux 1-D que l'on assemble à l'aide de la fonction **Assembler** (Bundle) du menu **Cluster et variant** de la palette **Fonctions>Programmation**. Le premier tableau contient l'ensemble des abscisses des points de la courbe, le second tableau contient l'ensemble des ordonnées des points de la courbe.



### Graphe multi-courbes

Là aussi, pour afficher plusieurs courbes à la fois sur un même graphe, il faut les assembler à l'aide de la fonction **Assembler** (Bundle) du menu **Cluster et variant** de la palette **Fonctions>Programmation**.



MCours.com