Université Paris-Sud

Licence et magistère de physique fondamentale

PROJETS EXPÉRIMENTAUX DE PHYSIQUE

Exercices d'initiation à LabView

http://hebergement.u-psud.fr/projets

Exercices d'initiation à LabVIEW

Lors du déroulement des séances d'exercices, vous trouverez une série de fichiers en partage sur le serveur SAMBA2 (Raccourci sur le bureau). Afin de ne pas saturer ce serveur, il est important, avant de commencer le travail, de télécharger les fichiers sur votre ordinateur, dans un sous-répertoire, nommé de vos nom et prénom, prévu à cet effet. À la fin de chaque séance, n'oubliez pas de sauvegarder le fruit de votre travail, contenu dans le sous-répertoire que vous avez créé, sur le serveur SAMBA2.

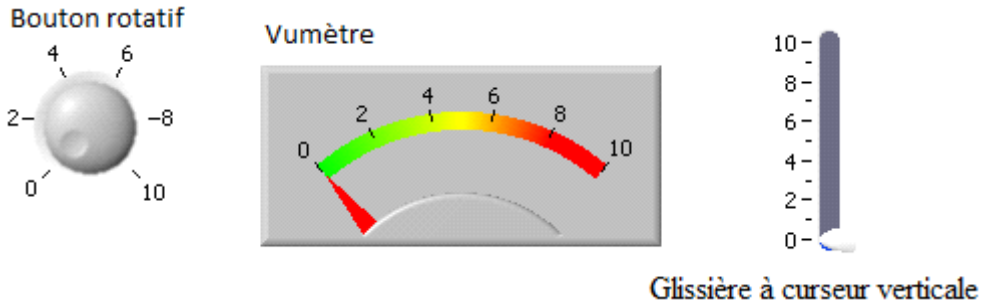
I. Premiers pas avec LabVIEW

1. Utilisation d'un programme (VI : Virtual Instrument) existant

- Sur le serveur SAMBA2, dans le dossier *partage_physta*t, vous trouverez une bibliothèque d'exemples fournie (*Demo.LLB*). Ouvrez le programme ***VI_Simple.vi*** contenu dans cette bibliothèque. Ce programme, comme tous les VI LabVIEW, possède une face avant (zone d'exécution du programme) et un diagramme (zone où est généré le code graphique).
- Essayez les divers boutons de commande sur la face avant.
- Ouvrez ensuite le diagramme de ce VI via le menu *Fenêtre/Afficher le diagramme* ou le raccourci clavier *CTRL-E* et comprenez à quoi correspondent les divers éléments du diagramme.

2. Réalisation d'un premier programme

Vous allez réaliser dans un premier temps, un programme qui consiste à afficher le résultat d'une opération simple à partir d'un nombre entré par un utilisateur :



- Ouvrir un nouveau VI.
- Afficher une commande numérique *Bouton rotatif* et un indicateur numérique *vumètre* sur la face avant (menu *Numérique* de l'onglet *Moderne* de la *Palette de commandes,* disponible dans le menu *Affichage* de la face avant).
- Dans le diagramme, construire un programme tel que *vumètre* affiche 2X+3 où X est la valeur donnée par *Bouton rotatif* (menu *Numérique* de l'onglet *Programmation* de la *Palette des fonctions,* disponible dans le menu *Affichage* du diagramme). Pour réaliser les liaisons filaires, munissez-vous de l'outil *connecter les terminaux* disponible dans la *Palette d'outils* (*Affichage/Palette d'outils*).
- Sauver le programme (*Enregistrer* ou *Enregistrer sous*) sous le nom ***Opération simple.vi***.
- Tester le fonctionnement en mode *Exécuter* puis *Exécuter en continu*.
- Modifier l'échelle de l'indicateur jusqu'à 35 par exemple.
- Changer le *Bouton rotatif* par un bouton *Glissière à curseur verticale* : vérifier l'effet. Modifier les noms de *Bouton rotatif* et de *vumètre* (par Entrée/Sortie par exemple).

Vous allez maintenant transformer ce programme en un sous-programme utilisable dans un autre programme :

- **Sur la face avant,** cliquer avec le bouton droit de la souris, sur le petit schéma situé dans le coin en haut à droite.
- Choisir *Editer l'Icone*, dessiner un rectangle blanc, écrire **Calc** à l'intérieur puis presser OK.
- Choisir ensuite *Visualiser le connecteur* dans le menu apparaissant au même endroit que précédemment en cliquant sur le bouton de droite de la souris, puis choisissez *Modèles*.
- Choisir la proposition offrant deux rectangles verticaux correspondant à une entrée et une sortie.
- Cliquer avec l'outil *connecter les terminaux* dans le rectangle de gauche, puis sur la commande apparaissant dans la face avant. La couleur du rectangle de gauche change alors. Faites de même avec le rectangle de droite et l'indicateur : vous venez de définir les entrées et sorties de l'icône qui représentera le programme **Calc** dans un autre programme.
- Aller dans *Fichier/Propriétés du VI/Documentation* et entrer un texte clair qui explique ce que fait ce VI.
- Sauver et fermer ce programme.

Vous allez maintenant insérer le programme précédent en tant qu'icône **Calc** dans le diagramme d'un autre programme et l'utiliser en tant que sous-programme :

- Ouvrir un nouveau programme, introduire une commande "X" et un indicateur "RESULTAT", et mettre dans "RESULTAT" : la valeur 2X+3 par exemple, en utilisant l'icône **Calc** correspondant au VI ***Opération simple.vi***. Pour cela, dans le diagramme, faire appel au VI ***Opération simple.vi*** via *Sélectionner un VI* dans la *Palette des Fonctions*.
- Remarquer qu'en approchant la bobine de l'icône, on voit apparaître les noms des entrées/sorties de ce VI. Vérifier que le programme fonctionne.
- Sauver ce programme sous le nom : ***Utiliser une icone.vi.***

3. Gestion des erreurs

Vous allez maintenant apprendre à visualiser et corriger une erreur :

- Lorsqu'il y a une erreur dans un programme, LabVIEW offre différents outils permettant d'en trouver l'origine. Ouvrir à nouveau le VI ***Opération simple.vi***.
- Créez une erreur volontaire en coupant un fil par exemple. Le symbole *flèche* du mode *Exécuter* change et devient une *flèche brisée*.
- Cliquez sur la *flèche brisée* : la liste des erreurs apparaît.
- Double-cliquez sur l'une d'elles, LabVIEW vous renvoie vers le lieu de l'erreur (ici le fil cassé).
- Réparez l'erreur.

Vous allez apprendre à suivre pas à pas une exécution :

- Parfois, votre programme vous renvoie un résultat inattendu, mais ne signale pas d'erreur. Pour trouver d'où vient le problème, il faut suivre son exécution pas à pas et vérifier la

valeur des différentes variables au fur et à mesure de l'exécution. Ouvrir à nouveau ***Operation simple.vi***.

- Pour mettre un point d'arrêt : utiliser l'outil en forme de rond rouge *placer/supprimer un point d'arrêt* accessible dans la *Palette d'outils*.
- Placer un point d'arrêt dans le programme en cliquant ce rond rouge sur un objet du programme qui s'entoure alors de rouge.
- Vérifier l'effet à l'exécution du programme.
- Utiliser également l'outil *sonder les données* (rond jaune avec un P dedans sur la *Palette d'outils*) en le plaçant sur des fils.
- Vérifier son effet à l'exécution.

Vous allez apprendre à utiliser l'aide :

- Lorsque l'on hésite sur l'utilisation d'un VI proposé par LabVIEW, on peut enclencher l'aide automatique que l'on obtient via les menus *Aide/Afficher l'aide contextuelle* ou par le raccourci clavier *CTRL-H*.
- Vérifier par exemple son effet quand vous placez le curseur sur les différents éléments d'un diagramme ou sur l'icône d'un VI. On observe ainsi tout l'intérêt, dès que l'on écrit un VI, de décrire de façon claire le rôle de ce VI via les menus *Fichier/Propriétés du VI/Documentation*.
- Parfois, on souhaite une aide plus complète pour voir en détail le fonctionnement, les options et des exemples associés à une fonctionnalité. Dans ce cas, aller dans les menus *Aide/Rechercher dans l'Aide LabVIEW*. A titre d'exemple, chercher l'aide sur les *Graphes déroulants*.

II. Les différents types de données

Sur le diagramme, les couleurs des fils et des icônes sont représentatives du type de données avec lesquelles on travaille. Dans ce projet, vous serez amenés à manipuler des nombres (entier et/ou réels), des chaînes de caractères, des booléens et des tableaux de ces différents types de données.

1. Les nombres

Vous allez constater que les entiers sont associés à la couleur **bleue** et qu'ils peuvent être affichés suivant différents formats ou représentation : décimal, octal, hexadécimal, ou binaire :

- Sur la face avant d'un nouveau VI, installer une commande numérique correspondant à un entier positif inférieur à 256, c'est-à-dire un entier sur 8 bits allant de 0 à 255 (2^8-1). Sur cette commande numérique, faire un clic droit et choisir *Représentation/U8*. Relier cette commande à 3 indicateurs numériques **entiers** différents, affichant respectivement l'entier en représentation hexadécimale, octale ou binaire. Pour cela faire un clic droit et choisir *Propriétés/Format d'affichage*.
- Sauver ce VI sous le nom ***Représentation entier.vi.***

Les réels, eux, sont associés à la couleur **orange**.

- Sur la face avant d'un nouveau VI, installer une commande numérique réelle (par défaut) et 2 indicateurs numériques, l'un réel et l'autre entier positif de représentation U32.
- Relier la commande aux 2 indicateurs numériques et observer le résultat.
- Sauver ce VI sous le nom ***Réel entier.vi.***

2. *Les chaînes de caractères*

LabVIEW utilise la couleur **rose** pour signaler que l'on travaille avec des chaînes de caractère.

- Afficher sur un *indicateur chaîne de caractères* (*Palette de commandes/Moderne/Chaîne et chemin),* une chaîne de caractères fournie par une *commande chaîne de caractères* et donner, sur un indicateur numérique, sa longueur correspondant au nombre de caractères qu'elle contient. Utiliser pour cela l'opérateur *Longueur d'une chaîne* obtenu via les menus *Palette de fonctions/Programmation/Chaîne*.
- Sauver ce VI sous le nom ***Chaines de caractères.vi***

3. *Les booléens*

Les booléens apparaissent avec une couleur **verte**.

- Afficher une commande *Interrupteur à glissière* et un indicateur *LED circulaire* (*Palette de commandes/Moderne/booléen)*.
- Changer le texte du booléen (en général, par défaut, ON/OFF) de telle sorte que la commande affiche le choix STOP/MARCHE et l'indicateur ROUGE/VERT.
- Les relier ensemble afin que lorsqu'on clique sur STOP, l'indicateur affiche ROUGE sur un fond rouge et que lorsque l'on clique sur MARCHE, l'indicateur affiche VERT sur un fond vert.
- Exécuter le programme en continu et cliquer sur les booléens avec l'outil en forme de main pour changer l'état des boutons. Vérifier le bon fonctionnement de ce programme.
- Sauver ce VI sous le nom ***Booléens.vi***

Vous allez maintenant tester les *Actions mécaniques* associées aux boutons booléens :

- On associe à tout bouton booléen une *Action mécanique* qui est accessible en cliquant sur le booléen dans la face avant avec le bouton droit de la souris. Cette *Action mécanique* définit le type d'interrupteur que l'on utilise. Par exemple, certains interrupteurs reviennent à leur position initiale quand on les a pressés, d'autres restent dans la position dans laquelle on les a placés.
- Reprendre le VI précédent et tester les différentes *Actions mécaniques* possibles en observant attentivement, en mode *Exécuter en continu*, ce qui se passe pendant que vous appuyez sur le bouton, puis quand vous relâchez la pression.

4. Les tableaux

On peut aussi manipuler toutes les données précédentes dans des **tableaux**, la couleur est alors conservée pour les différents types de données mais le fil reliant les différentes icônes devient plus épais.

Vous allez générer et travailler avec un tableau de réels :

- Afficher un tableau sur la face avant d'un nouveau VI via *Palette des commandes/Moderne/Tableau, matrice, cluster/Tableau* et placer à l'intérieur de celui-ci une commande numérique : cela permet de définir le type d'éléments constituant ce tableau, ici le tableau sera un tableau de réels.
- Entrer 4 valeurs dans le tableau. On peut décaler la visualisation d'un élément du tableau en agissant sur le curseur visible sur la gauche du tableau. On peut aussi visualiser plusieurs éléments du tableau en agrandissant celui-ci avec la souris.
- Multiplier ces 4 éléments par 2. On remarquera ici que l'on effectue directement une même opération arithmétique sur l'ensemble des éléments du tableau et que les fils apparaissent alors plus épais sur le diagramme.

Vous allez générer un tableau de booléens et observer le clignotement des LED correspondantes :

- Afficher, à l'aide d'un tableau d'indicateurs *LED circulaire*, l'état des bits (bit à 0 ⇔ diode éteinte ; bit à 1 ⇔ diode allumée) de l'octet représentant un nombre entier positif inférieur à 256 (*représentation U8*).
- Présenter les bits de poids faible de l'octet à droite (comme en écriture binaire). On utilisera pour cela l'opérateur *Nombre en tableau de booléen* du menu *Palette des Fonctions/Programmation/Numérique/Conversion,* ainsi que l'opérateur *Renverser un tableau 1D* dans *le* menu *Palette des Fonctions/Programmation/Tableau.*
- Exécuter en mode *Exécuter en continu*, changer la valeur de l'entier et observer le résultat.
- Sauver sous ***Tableau de booléens.vi***.

III. Les représentations graphiques des données numériques

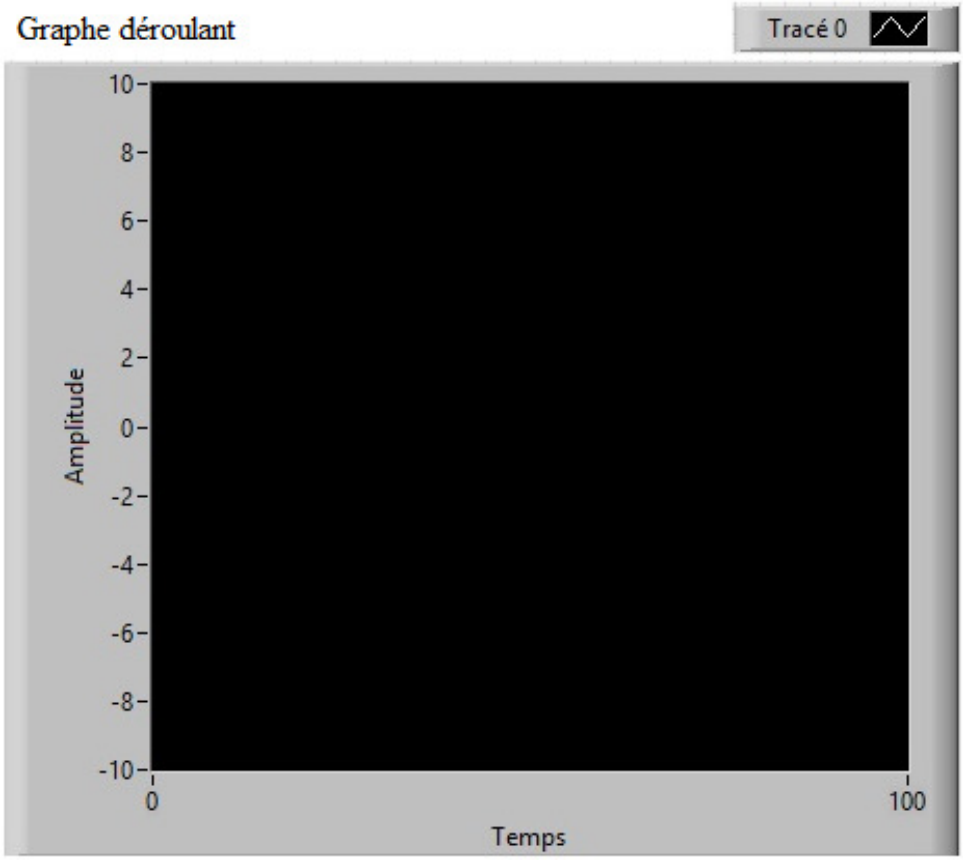
Dans LabVIEW, il y 3 types de représentations graphiques : les graphes déroulants, les graphes et les graphes XY. Nous ne travaillerons ici qu'avec les graphes déroulants et les graphes XY.

1. Les graphes déroulants

Vous allez afficher une variable, en temps réel, au court du temps :

- Placer dans le diagramme d'un nouveau VI, un générateur de nombres aléatoires compris entre 0 et 1 (*Palettes des Fonctions/Programmation/Numérique/Nombre aléatoire)* qui va simuler une mesure physique.
- Appliquer à cette fonction aléatoire, l'opération 2X+3 et envoyer le résultat dans un indicateur.
- Sauver sous ***Simulation manip.vi***.

- Editer l'icône de ce VI et écrire « Simul Manip », puis choisir un seul rectangle dans *Modèles* et la connecter à la sortie correspondant au nombre aléatoire 2X+3.
- Sauvegarder.
- Afficher sur la face avant d'un nouveau VI, un *Graphe déroulant*, obtenu depuis la face avant, via le menu *Palette des commandes/Moderne/Graphe* ainsi qu'un indicateur numérique.
- Mettre ***Simulation Manip.vi*** en entrée de ces 2 indicateurs.
- Exécuter en mode *Exécuter en continu* et observer le résultat.
- Sauver ce VI sous le nom ***Exemple déroulant.vi.***



Vous allez maintenant tester les différentes options du *Graphe déroulant* :

- Cliquer avec le bouton droit de la souris sur le G*raphe déroulant*, choisir *Eléments visibles,* essayer les différentes options et visualiser les changements induits.
- Changer l'échelle du graphe déroulant en ordonnée en remplaçant un nombre par un autre, via l'outil *Editer du Texte* de la *Palette d'outils* et en cliquant sur le nombre.
- Essayer l'option *Echelle automatique en X (clic droit/Echelle des X/Mise à l'échelle automatique des X)* et en Y *(clic droit/Echelle des Y/Mise à l'échelle automatique des Y).*

2. Les graphes XY

Les graphes XY seront manipulés par la suite. Sachez qu'ils permettent de tracer une quantité en fonction d'une autre et qu'ils n'acceptent que des tableaux en entrée.

IV. Les structures

Attention, pour la suite, ne surtout plus utiliser l'option *Exécuter en continu.*

1. *La structure « Boite de calcul »*

Cette structure « Boite de calcul » permet d'écrire des formules mathématiques.

- Ouvrir le VI ***Opération Simple.vi*** et le sauver sous le nouveau nom ***Opération avec Boite de calcul.vi***. On va remplacer les opérateurs arithmétiques par une *Boite de calcul*, accessible via les menus *Palettes des fonctions/Programmation/Structure*.
- Placer une entrée et une sortie en positionnant le curseur de la souris sur le cadre de la *Boite de calcul* et en cliquant sur le bouton droit de la souris.
- Ecrire le nom des variables (par exemple X en entrée et Y en sortie).
- Ecrire, dans le cadre de la *Boite de calcul*, la formule à effectuer suivie d'un point-virgule comme sur l'exemple : Y=2*X+3; et tester son fonctionnement. Essayer ensuite une formule à plusieurs variables d'entrées et calculer par exemple :

  z=cos(2*3.14*sqrt(y*y+x*x)) .

  Rmq : Si une virgule apparaît à l'intérieur de la *Boite de calcul* alors que vous avez entré un point, appelez l'enseignant. Il faut, en effet, changer le symbole décimal définis par défaut dans Windows dans les paramètres régionaux.

2. *La structure « Condition »*

Vous allez utiliser une structure *Condition* pour répondre à une question :

- Réaliser un programme qui répond à une question apparaissant dès que l'on lance le programme. L'utilisateur doit avoir le choix de répondre OUI ou NON à la question posée sur la fenêtre qui s'ouvre devant lui au début de l'exécution du programme. En fonction de sa réponse, le programme lui renvoie un commentaire différent.
- Pour cela, utiliser une *Boite de dialogue à 2 boutons* accessible dans le menu *Palette des Fonctions/Programmation/Dialogue et interface utilisateur* à laquelle vous reliez des constantes *chaînes de caractères* permettant d'introduire la question et de proposer le choix de réponse OUI ou NON.
- Relier la sortie booléenne de cette *Boite de dialogue à 2 boutons* à la structure *Condition*, accessible depuis *Palettes des fonctions/Programmation/Structure*.
- Mettre dans les deux fenêtres VRAI et FAUX de la structure *Condition* une *Boite de dialogue à un bouton* accessible dans *Palette des* Fo*nctions/Programmation/Dialogue et interface utilisateur* et introduire le message (constante chaine de caractères) à faire apparaître à l'utilisateur en fonction de sa réponse à la question posée.
- Sauver sous le nom ***Question Réponse.vi***.

3. *La structure « Boucle For »*

Vous allez utiliser une *Boucle For* qui permet d'effectuer une opération, un nombre de fois spécifié à l'avance :

- Réaliser un programme qui affiche sur deux indicateurs différents, dans la face avant d'un nouveau VI, l'indice d'une *Boucle For* (*Palettes des fonctions/Programmation/Structure*). Sur le diagramme, l'un de ces 2 indicateurs sera placé à l'intérieur de la *Boucle For*, l'autre à l'extérieur de cette même boucle. Pour pouvoir connecter l'indice de la boucle à l'indicateur situé à l'extérieur de la boucle, il faudra utiliser l'option *Activer/Désactiver l'indexation* du fil en cliquant sur le bouton droit de la souris, sur le petit carré apparaissant au passage du fil au travers de la boucle. Cette option permet soit de faire sortir de la boucle un tableau (fil épais) de tous les indices ayant circulé sur le fil, soit de faire sortir de la boucle la dernière valeur ayant circulé sur ce même fil (fil fin).
- Ajouter une fonction *Attendre (ms)* (*Palettes des fonctions/Programmation/Informations temporelles*) dans la boucle, pour ralentir le fonctionnement du programme, en reliant une commande numérique à la fonction *Attendre (ms)*, pour spécifier le nombre de millisecondes souhaitées entre chaque tour de boucle.
- Sauver ce VI sous le nom ***Boucle For simple.vi***.
- Changer ensuite l'option *Activer/Désactiver l'indexation* et relier l'indice à l'extérieur de la *Boucle For* à un indicateur tableau d'entiers situé sur la face avant. Observer le changement de l'épaisseur du fil de sortie : un tableau (fil épais) sort cette fois-ci de la boucle.
- Sauver ce VI sous le nom ***Boucle For et tableau.vi***.

Vous allez utiliser une *Boucle For* pour un calcul incrémenté :

- Réaliser un programme utilisant une *Boucle For* pour calculer la somme des N premiers entiers, l'utilisateur choisissant la valeur de N dans une Commande numérique située sur la face avant et reliée au nombre de tour de boucle. Pour cela, on doit utiliser ce que l'on appelle un *Registre à décalage* qui permet de faire des calculs incrémentés. Ce *Registre à décalage* utilise, pour le tour de boucle i+1 et dans une case située sur le flan gauche de la boucle, la valeur d'un calcul obtenue au tour de boucle précédent i et placée dans la case située sur le flanc droit de la boucle. Pour faire apparaître un *Registre à décalage*, on se place sur un flan vertical de la boucle, on clique sur le bouton droit de la souris et on sélectionne *Ajouter un registre à décalage*.
- Sauver ce VI sous le nom ***Boucle For et Registre à déclage.vi***.
- Exécuter ce VI et vérifier que la somme des N premiers entiers est correcte.

Vous allez générer un tableau de mesures et calculer la moyenne et l'écart type des mesures :

- Dans le diagramme d'un nouveau VI, insérez, dans une *Boucle For*, l'icône du VI ***Simulation manip.vi*** pour générer N mesures aléatoires.
- Placer une commande numérique sur la face avant permettant à l'utilisateur de choisir le nombre N de mesures à effectuer. Ce programme revient à simuler un échantillonnage de N mesures.
- Tracer le tableau de ces N mesures ainsi générées dans un *Graphe déroulant*.
- Calculer et afficher la moyenne et l'écart type de ce tableau de mesures en utilisant l'opérateur *Variance et écart-type* que vous trouverez via les menus *Palettes des Fonctions/Mathématiques/Probabilités et Statistique*.
- Sauver ce VI sous le nom ***Simulation echantillonnage.vi***.
- Que se passe-t-il si N = 0 ?

4. La structure « Boucle While »

Vous allez utiliser une *Boucle While* qui permet d'effectuer une opération, autant de fois qu'on le souhaite ou tant qu'une condition est vérifiée :

- Réaliser un programme qui affiche, dans deux indicateurs numériques différents situés sur la face avant d'un nouveau VI, l'indice d'une *Boucle While*. Dans le diagramme, l'un de ces 2 indicateurs sera placé à l'intérieur de la *Boucle While*, l'autre à l'extérieur de cette même boucle. L'arrêt de la *Boucle While* est commandé par une commande bouton booléen mis sur la face avant.
- Ajouter dans la boucle, une fonction *Attendre (ms)* pour ralentir l'exécution du programme.
- Entrer dans une commande numérique reliée à la fonction *Attendre (ms)*, le nombre de millisecondes souhaitées entre chaque tour de boucle.
- Sauver ce VI sous le nom ***Boucle While simple.vi***.

5. La structure « Séquence »

Vous allez utiliser la structure *Séquence* qui permet d'effectuer plusieurs opérations à la suite l'une de l'autre, dans un ordre bien précis :

- Créer un programme qui multiplie, dans un premier temps, par 2 la valeur d'un nombre entré sur la face avant dans une commande numérique, puis qui attend, dans un deuxième temps, 2 secondes et finalement qui affiche, dans un troisième temps, la valeur du résultat obtenu dans un Indicateur numérique mis sur la face avant. Ce programme se fait en utilisant la *Structure Séquence empilée* que l'on trouve via les menus *Palettes de Functions/Programmation/Structure.*
- Pour ajouter les deuxième et troisième étapes à la première, cliquer sur le bouton droit de la souris lorsque le curseur de la souris est placé sur le cadre de la *Séquence*, choisir dans le menu proposé *Ajouter une étape après*.
- Pour faire passer le nombre généré par la première étape à la troisième étape, cliquer sur le bouton droit de la souris lorsque que le curseur de la souris est placé sur le cadre de la *Séquence* et choisissez dans le menu proposé *Ajouter une variable locale de séquence* : cela crée une petite boîte dans laquelle on peut stocker une valeur via un fil. Cette valeur peut être récupérée ensuite lors des étapes suivantes de la séquence via la boîte.
- Sauver ce VI sous le nom ***Utilisation Sequence.vi***.

V. Les graphes XY

1. *Tracé de Y fonction de X :* Graphe XY *à la sortie d'une boucle*

On a déjà vu l'utilisation des *Graphes déroulants* qui tracent des valeurs uniques en fonction du temps. Vous allez maintenant tracer un tableau Y en fonction d'un tableau X générés automatiquement à la sortie d'une *Boucle For :*

- Utiliser une *Boucle For* pour générer automatiquement un tableau des N premiers entiers et un tableau des N premiers entiers au carré.
- Envoyer ces deux tableaux vers un *Graphe XY*. N'oublier pas d'insérer l'opérateur *Assembler* que l'on obtient via les menus *Palettes de Fonctions/Programmation/Cluster, classe et variant* pour générer un agrégat de tableau de X et de Y, comme le montre l'aide (*CTRL-H*) lorsque l'on place la souris sur l'icône du *Graphe XY* dans le diagramme.
- Sauver ce VI sous le nom ***Tracé d'une courbe XY.vi***.

Vous allez maintenant tracer deux courbes (X,Y) en générant un tableau d'agrégats :

- Utiliser une *Boucle For* pour générer un tableau des N premiers entiers, un tableau des carrés des N premiers entiers, et un tableau des cubes des N premiers entiers. Utiliser deux opérateurs *Assembler* pour agréger ces N premiers entiers (X) avec leur valeur au carré et au cube (Y1 et Y2 respectivement).
- Utiliser ensuite l'opérateur *Construire un tableau* via le menu *Palettes de Fonctions/Programmation/Tableau* pour faire un tableau à partir des deux agrégats précédents (quand l'outil *Construire un tableau* apparaît, il n'a qu'une entrée et il faut donc l'étirer avec la souris pour faire apparaître les deux entrées nécessaires).
- Envoyer le tableau d'agrégats ainsi construit vers le G*raphe XY*.
- Sur la face avant, étirer la légende en haut à droite du G*raphe XY* pour voir les deux légendes, puis choisir une autre couleur et un autre tracé pour la deuxième courbe (*clic droit sur la légende du graphe*).
- Sauver ce VI sous le nom ***Tracé de 2 courbes XY.vi***.

2. *Graphes XY au cours du temps*

Vous allez maintenant afficher un G*raphe XY* à chaque tour de boucle :

- Reprendre, pour cela, le programme ***Tracé d'une courbe XY.vi*** et ajouter un G*raphe XY* sur la face avant.
- Puis, grâce à l'utilisation d'un *Registre à décalage* et de l'opérateur *Construire un tableau*, construire un tableau X des N premiers entiers et un tableau Y des N premiers entiers au carré, dont le nombre d'éléments augmente d'une unité à chaque tour de boucle.
- L'opérateur *Assembler* sera ensuite utilisé pour agréger les 2 tableaux à chaque tour de boucle. L'agrégat ainsi formé sera relié au G*raphe XY* récemment importé et mis à l'intérieur de la *Boucle For* sur le diagramme.
- On n'oubliera pas d'initialiser le tableau en reliant le *Registre à décalage* à une constante tableau nulle (*clic droit sur le registre à décalage*/*Créer/constante*) et d'ajouter une fonction *Attendre (ms)* pour ralentir le fonctionnement du programme.
- Exécuter le programme et observer le *Graphe XY* situé à l'intérieur de la boucle se construire point par point et celui situé à l'extérieur s'afficher à la fin du programme.
- Sauver ce VI sous le nom ***Tracé d'une courbe XY point par point.vi***.

VI. Les sauvegardes sur fichiers « texte »

Vous allez enregistrer des données à la sortie d'une boucle :

- Ouvrir ***Fichier_Texte_Write.vi*** dans Demo.llb.
- Ajouter une *Commande Chemin* (*Palette des commandes/Moderne/Chaîne et chemin)* pour que le chemin d'accès soit dirigé sur votre répertoire.
- **Comprendre** comment fonctionne le programme et l'utiliser pour créer le fichier *MonPremierEssai.txt* constitué d'enregistrements de deux nombres réels (attention, le suffixe «.txt » est important).

Vous allez enregistrer des données à chaque tour de boucle :

- Ouvrir ***Fichier_Texte_Write_Seq.vi*** dans Demo.llb.
- Modifier la constante *Commande Chemin* pour que le chemin d'accès soit dirigé sur votre répertoire.
- **Comprendre** le programme. Quelle est la différence par rapport au programme ***Fichier_Texte_Write.vi*** ?
- Modifier-le, pour qu'il enregistre en plus, en première colonne, l'indice *i* de la boucle après l'avoir sauvegardé sous le nouveau nom ***Mon_Fichier_Texte_Write_Seq.vi***.
- Utiliser ce VI pour créer ***MonDeuxièmeEssai.txt.***

VII. Pour aller plus loin…..

1. Simulation d'une mesure

- Ecrire un programme simulant une mesure de température fluctuant de manière sinusoïdale autour d'une valeur moyenne qui augmente linéairement de 300 K à 1500 K. A chaque température, on associe une mesure de tension fictive V = 4*T+5. Sauver ce programme sous le nom ***TempeCroiss.vi***.

2. Prise d'une mesure à intervalles réguliers

- Ecrire un programme qui simule une mesure de température (à l'aide de ***TempeCroiss.vi***) et qui sauvegarde régulièrement les mesures de température (1 mesure sur 5 par exemple) dans un fichier.
- Sauver ce programme sous le nom ***Acquisition Reguliere.vi***.

3. Prise d'une mesure à la demande

- Ecrire un programme qui simule une mesure de température (à l'aide de ***TempeCroiss.vi***) lorsque l'utilisateur appuie sur un bouton « MESURE ». Prévoir également un bouton « ARRET » pour quitter le programme.
- Sauver ce programme sous le nom ***Acquisition à la demande.vi.***