

IUT- Université Bordeaux1
Département Mesures Physiques

Livre 8

Initiation au logiciel labView 6.1

Novembre 2002

Francis Cottet (ENSMA)
Rodolphe Decourt (ICMCB)
Patrice Dordor (IUT Bx1)

SOMMAIRE

1	PRINCIPE DE LABVIEW	3
2	LE LANGAGE G.....	4
2.1	RÈGLES DE CONTRÔLE DES FLUX DE DONNÉES :.....	5
2.2	EXEMPLE	ERREUR ! SIGNET NON DÉFINI.
3	UTILISATION DE LABVIEW SOUS WINDOWS.....	6
3.1	L'ENVIRONNEMENT DE LABVIEW	6
3.1.1	<i>La boîte à outils</i>	8
3.1.2	<i>La Fenêtre Commandes</i>	8
3.1.3	<i>La Fenêtre Fonctions</i>	9
3.1.4	<i>Création d'un VI</i>	9
3.1.4.1	Définition de l'interface utilisateur	10
3.1.4.2	Edition du diagramme flot de données.....	11
3.1.4.3	Encapsulation	12
4	ELÉMENTS DE PROGRAMMATION.....	14
4.1	VARIABLES DE TABLEAUX	14
4.2	CLUSTERS	1514
4.3	STRUCTURES DE PROGRAMMATION	15
4.3.1	<i>Structure de choix</i>	15
4.3.2	<i>Boucle "Pour"</i>	1716
4.3.3	<i>Structure "Tant que"</i>	1716
4.4	TECHNIQUES DE DÉBOGGAGE	1817
4.4.1	<i>Erreurs courantes</i> :.....	1918
4.4.2	<i>Sonde et points d'arrêt</i>	2019
4.5	SAUVEGARDE DES VIS.....	2019
4.6	MÉMENTO SUR LE TRACÉ DE COURBES	2120
4.6.1	<i>Graphe déroulant</i>	2120
4.6.2	<i>Graphe « oscilloscope »</i>	2221
4.6.3	<i>Graphe XY</i>	2423
5	BIBLIOGRAPHIE	2423
6	RACCOURCIS ET SUGGESTIONS	2524
6.1	RACCOURCIS.....	2524
6.2	SUGGESTIONS.....	2524

Introduction au logiciel d'instrumentation LabVIEW

LabVIEW (*Laboratory Virtual Instrument Engineering Workbench*) est un logiciel de développement d'applications d'instrumentation. Ce logiciel offre de larges possibilités de communication entre l'ordinateur et le monde physique, (par cartes d'acquisitions analogiques ou digitales, cartes GPIB, réseau, liaisons série et parallèles, etc.) ainsi que d'importantes bibliothèques mathématiques permettant de traiter les signaux mesurés.

L'idée de LabVIEW est de remplacer les instruments de mesures et d'analyse d'un laboratoire par un ordinateur muni de cartes spécifiques et d'un logiciel approprié. Les cartes permettent de convertir des signaux électriques (provenant de capteurs mesurant des grandeurs physiques) en données numériques. Ainsi, un seul ordinateur muni d'une carte d'acquisition analogique et de LabVIEW est capable de remplacer un voltmètre, un fréquencemètre ou un oscilloscope. Le programme qui pilote ces cartes est appelé VI (-Virtual Instrument).

1 Principe de LabVIEW

LabVIEW permet de réaliser, entre autre, des instruments virtuels. Par extension, on appellera VI toute application réalisée avec LabVIEW. Un VI est composé de trois parties liées :

- Une **face avant** (*Panel*) cf. Fig.1 : interface de communication avec l'utilisateur, ou avec d'autres VIs. Cette face avant, est composée d'objets graphiques : interrupteurs, des potentiomètres, des zones de graphismes, etc..
- Un **diagramme** (*Diagram*) cf. Fig.2 : cette partie décrit le fonctionnement interne du VI. On utilise le langage G, langage de programmation graphique, pour décrire ce fonctionnement.
- Une **icône** (*Icon*) cf. Fig.3 : symbolisation de l'instrument virtuel qui permettra de faire appel à un instrument virtuel déjà créé à l'intérieur d'un autre instrument virtuel, (structure hiérarchique et de modularité modulaire).

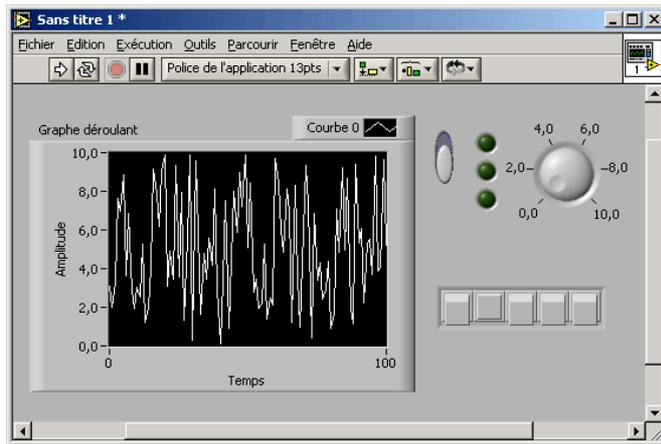


Fig.1 - Face avant (*Panel*)

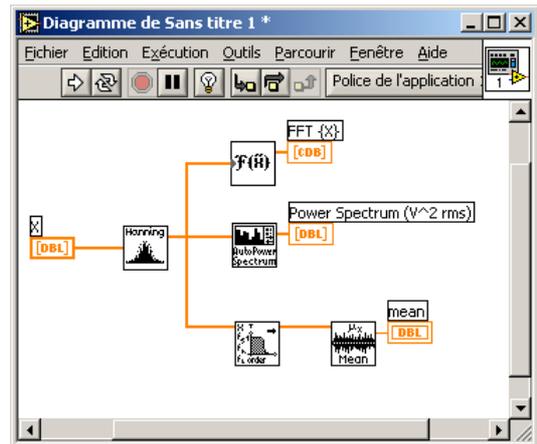


Fig.2 - Diagramme (*Diagram*)



Fig.3 - Icône (*icon*)

2 Le langage G

Le langage G se base sur le principe du **flot de données**, auquel ont été ajoutées des structures de programmation et un ensemble de fonctions très développé.

Un diagramme flot de données exprime une fonction, de **manière graphique**.

2.1 Interfaces

Un tel diagramme (*cf.* Fig.4) est composé de :

- **terminaux**: définissent les entrées (rectangle gras) et les sorties (rectangle fin) de la fonction,
- **nœuds**: définissent les opérations à effectuer. Ils sont représentés par un carré pouvant éventuellement contenir une image illustrant leur fonctionnalité,
- **arcs orientés**: relient nœuds et terminaux et permettent d'indiquer le passage de données d'un nœud vers un autre. Par convention, ces arcs sont orientés implicitement de gauche à droite.
- **jetons**: ils représentent les données transitant sur les arcs. Ils sont représentés par des points (*visibles seulement en mode pas à pas*).

2.2 Règles de contrôle des flux de données :

Plusieurs ~~regles-règles~~ simples permettent de gérer le flux (ou flot) des données dans le diagramme :

- à l'initialisation, les terminaux d'entrées produisent chacun un jeton,
- lorsqu'un nœud possède un jeton sur chacun de ses arcs entrants, il peut être exécuté: chaque jeton en entrée est consommé et le nœud produit un jeton sur chacun de ses arcs sortants.

Une partie d'un diagramme peut être encapsulée afin d'être réutilisée, en tant que nœud, par d'autres diagrammes. Les terminaux du diagramme deviennent alors les entrées/sorties du nœud.

Le diagramme de la Fig.4 exprime le calcul de " $d = a + b * (b - c)$ " et de " $e = b - c$ ". La Fig.5 représente la forme encapsulée de ce diagramme.

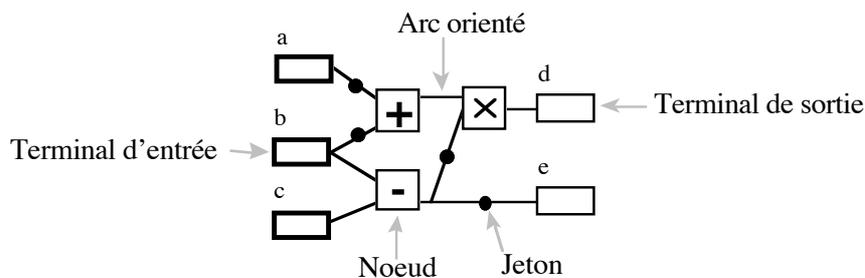


Fig.4 - Diagramme flot de données

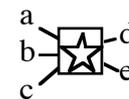
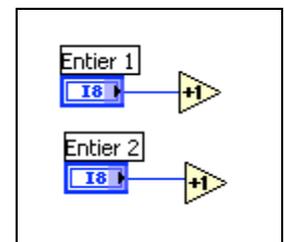
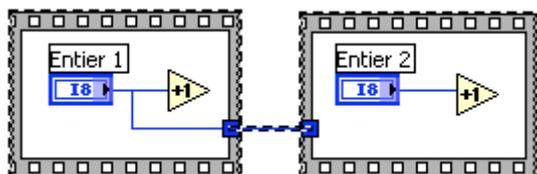


Fig.5 - Encapsulation d'un diagramme flot de données

A l'instant considéré, le nœud "+" peut être exécuté, et le nœud "-" l'a été. Le nœud "*" ne peut être exécuté pour l'instant puisqu'un de ses arcs d'entrée ne possède pas de jeton. Cela ne sera possible lorsque le nœud "+" aura été lui-même activé et qu'il aura produit un jeton sur son arc de sortie

L'ordre d'exécution de ~~tâches-tâches~~ mises en parallèle n'est pas défini

Exemple de chaînage forcé de tâches parallèles



3 Utilisation de LabVIEW sous Windows

Pour lancer LabVIEW, suivre la séquence :

Démarrer/programmes/National Instruments /LabVIEW 6.1

ou bien, cliquer sur le raccourci bureau LabVIEW (cf. Fig.6).

ou cliquer sur la flèche → dans la barre de menus



Raccourci bureau

Fig.6 -

3.1 L'environnement de LabVIEW

Lorsque LabVIEW est lancé, celui-ci ouvre par défaut un nouveau VI. Deux fenêtres vierges apparaissent à l'écran : une **face avant** ou *Panel* (sur fond gris) et un **diagramme** ou *Diagram* (sur fond blanc) (cf. Fig.7).

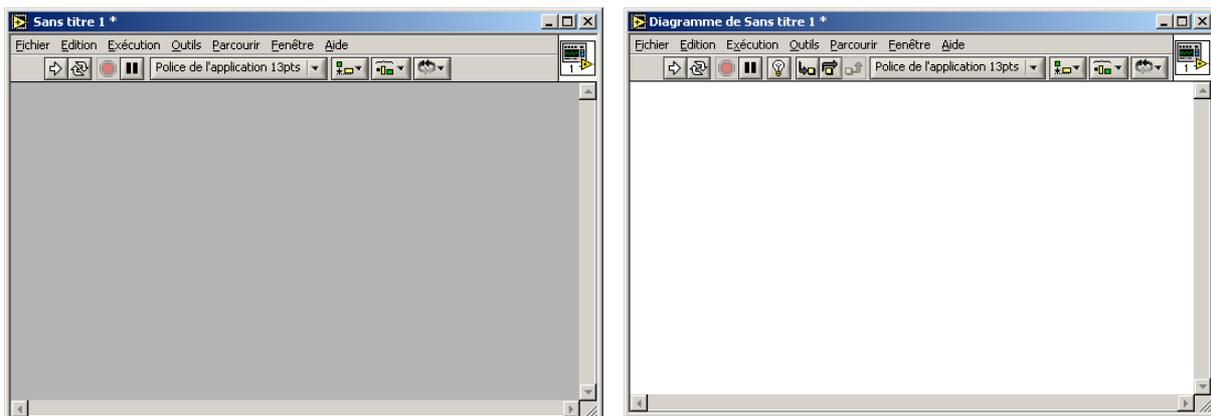


Fig.7 - Nouveau VI

Chacune de ces fenêtres est composée de :

- **une barre de titre**, contenant le titre du VI. L'en-tête *Diagramme* indique que la fenêtre considérée est le diagramme. La fenêtre dont la **barre de titre est bleue** indique que celle-ci a le **contrôle actif**, ~~e'est à dire~~ cela signifie que e'est à elle ~~que s'adressent~~ reçoit les informations provenant du clavier ou de la souris. On donne le contrôle- (ou focus) à une fenêtre en cliquant dessus.
- **une barre de menus déroulants**. Ceux-ci permettent de gérer la sauvegarde des programmes, de gérer les différentes fenêtres, etc.

- une barre d'outils d'exécution/édition (cf. Fig.8) se retrouve dans-en haut de chacune des fenêtres et permet de lancer l'exécution d'un VI, de l'arrêter, etc...

Différentes fenêtres Trois palettes flottantes permettent de créer des Vis VI :

- la boîte-à-palette d'outils va permettre de manipuler les objets graphiques créés,
- la fenêtre-palette Commandes va permettre de créer des objets graphiques sur la **face avant**,
- la fenêtre-palette Fonctions va permettre de créer des objets graphiques dans le **diagramme**.

Menu contextuel :

Le fait de cliquer avec le **bouton droit** de la souris **donne accès** à un menu contextuel dépendant de l'objet désigné. Celui-ci et permettant d'ene changer les propriétés de cet objet.

Dans la face avant, un clic *droit*, donne accès à la fenêtre-palette Commandes et dans le *diagramme*, à la fenêtre-palette fonction

Les fenêtre-palettes de LabVIEW sont généralement des fenêtre-palettes fugitives, c'est-à-dire qu'elles n'existent que le temps de la manipulation désirée. Cependant, on peut rendre leur **affichage permanent** en cliquant sur la punaise (cf. Fig.9) qui se trouve en haut à gauche de ces fenêtre-palettes.



Fig.9 – Punaise

Utilisée pour fixer un panneau

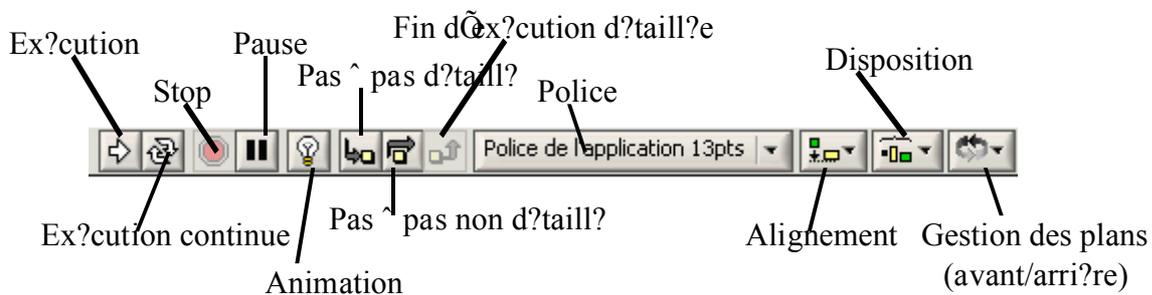


Fig.8 - Barre d'exécution/édition

3.1.1 La boîte à palette d'outils

~~La~~ Cette boîte à outils (cf. Fig.10) permettant de sélectionner ou de manipuler les objets graphiques. Elle apparaît en suivant la séquence de menu *Fenêtres/Palette d'outils* ou bien en faisant un clic *droit* dans la fenêtre avec la touche shift appuyée.

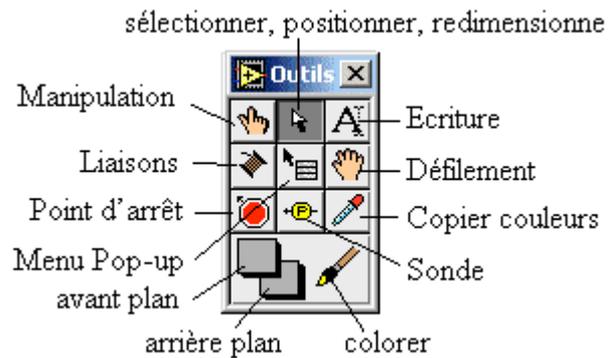


Fig.10 - Boîte à outils

Une fois apparue, la boîte à outils est toujours visible. On sélectionne l'outil désiré en cliquant dessus.

Raccourcis : touche de tabulation ou barre d'espace permettent de commuter d'un outil à l'autre.

3.1.2 La Fenêtre palette des Commandes

Lorsqu'une fenêtre de type *face avant* est active, on peut faire apparaître une fenêtre flottante nommée *Commandes* (cf. Fig.11) permettant de positionner des objets graphiques sur la face avant depuis la barre de menus *Fenêtre / Palette de commandes* ou bien en faisant un clic *droit* dans la face avant. Une fenêtre sous menu contenant des objets graphiques apparaît lorsque l'on clique sur les icônes de cette fenêtre (cf. Fig.11).

La sélection d'un objet dans un de ces sous menus permet de le disposer dans la *face avant*. De plus, elle lui fait automatiquement correspondre un terminal dans le *diagramme* associé (cf. Fig.12).

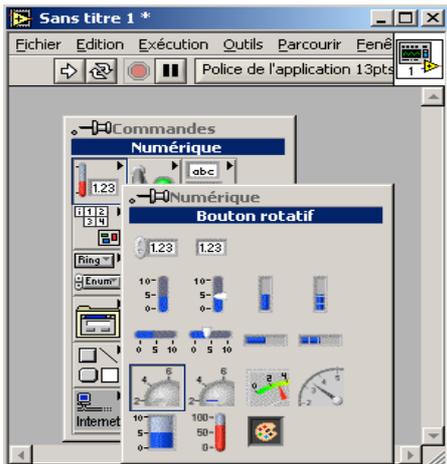


Fig.11 - La **fenêtre-palette commandes** et ses sous menus

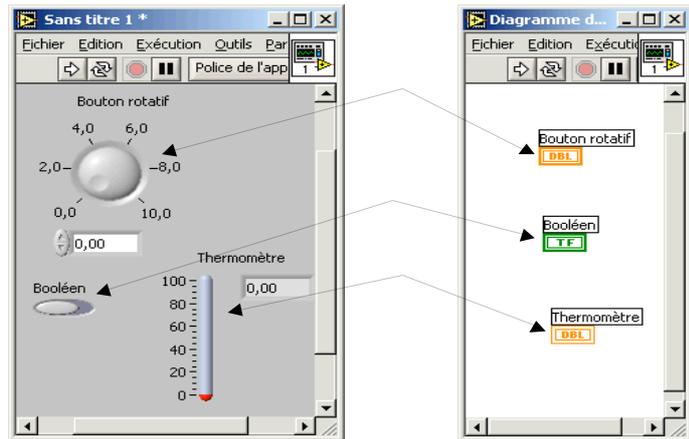


Fig.12 - Correspondance des objets entre *face avant* et *diagramme*

3.1.3 La Fenêtre-palette des Fonctions

De la même manière que pour la *face avant*, lorsqu'une fenêtre de type *diagramme* est active, on peut faire apparaître une fenêtre flottante nommée *Fonctions* (Fig. 13) permettant de positionner des opérateurs dans le diagramme en cliquant sur la barre de menus : *Fenêtre/Palette de fonctions* ou bien en faisant un clic *droit* dans la fenêtre diagramme

Une fenêtre appelant sous menu contenant des nœuds (fonctions) apparaît lorsque l'on clique sur les icônes de cette fenêtre (Figure 13). La sélection d'un opérateur permet de le déposer dans le diagramme.

3.1.4 Création d'un VI

La création d'un VI se déroule en quatre phases :

- Analyse de la tâche, définition des variables d'entrées et de sortie,
- Création de l'interface utilisateur, souvent invisible pour les sous **Vis VI**,
- Edition du diagramme flot de données,
- Encapsulation du VI.

Lors de la création d'un nouveau VI, il faut partir d'un VI vierge accessible par la commande :

menu *Fichier /Nouveau*

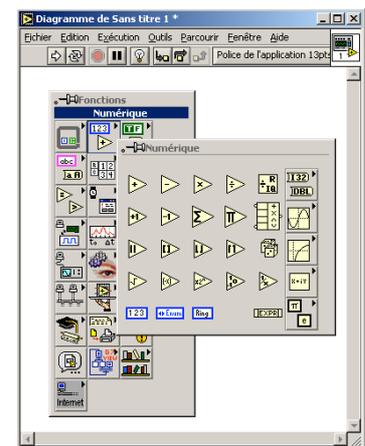


Fig.13 - La **fenêtre-palette Fonctions** et se sous menu

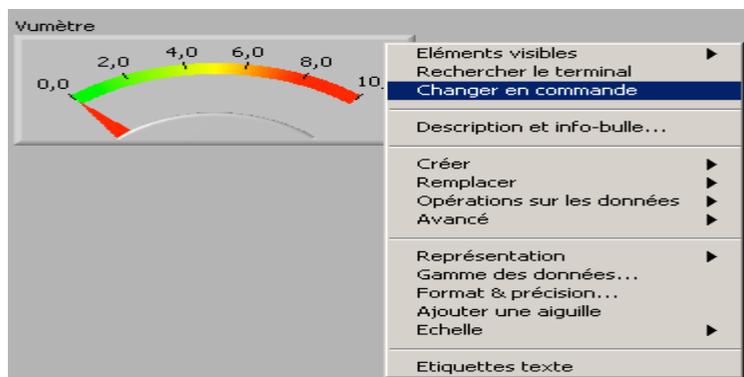
3.1.4.1 Définition de l'interface utilisateur

Il s'agit de placer dans la *face avant* les différents objets permettant de communiquer avec l'utilisateur. Ces objets proviennent de la fenêtre-palette *Commandes* et peuvent être de deux natures :

- **commandes** : variable d'entrée, l'utilisateur pourra manipuler cet objet pour donner des consignes. Par exemple un bouton ou bien un potentiomètre.

- **indicateur** : variable de sortie, l'utilisateur lira la valeur de cet objet pour avoir des informations. Par exemple un vu-mètre ou bien un graphique.

Un objet de type *commande* peut être transformé en *indicateur* (ou un *indicateur en commande*). Pour cela, il suffit de faire un clic *droit* sur l'objet et de choisir le



menu *Changer en indicateur* (ou *Changer en commande*). Il est à noter que ce *clic droit* fait apparaître un menu qui permet de changer l'ensemble des caractéristiques de l'objet (Figure 14).

Fig.14 - Un *indicateur* type vu-mètre

Lors de la création de ces objets dans la *face avant*, les terminaux associés apparaissent automatiquement dans le diagramme. Les terminaux correspondant à une *commande* sont entourés d'un bord épais (cf. Fig.15), ceux correspondant à un *indicateur* sont entourés d'un bord fin (cf. Fig.16).

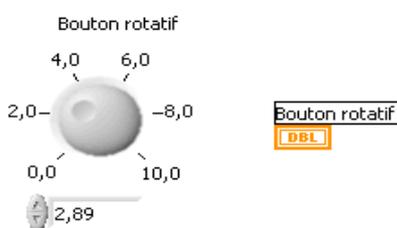


Fig.15 - Une *commande (face avant)* et son terminal entouré d'un bord épais

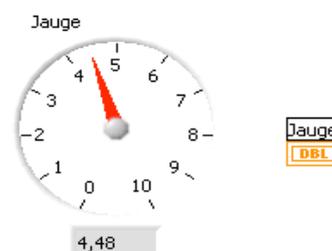


Fig.16 - Un *indicateur (face avant)* et son terminal entouré d'un bord fin

3.1.4.2 Edition du diagramme flot de données

Lorsque l'interface utilisateur a été définie, tous les terminaux nécessaires au diagramme flot de données se trouvent dans la fenêtre *diagramme*. Il faut maintenant placer les nœuds (ou opérateurs) du diagramme. Ces nœuds se trouvent dans la fenêtre *Fonctions*. Il suffit de les sélectionner et de les **déposer** dans le diagramme (cf. Fig.17).



Fig.17 - Disposition des nœuds dans le diagramme

Fonction réalisée par ce diagramme :
convertir si nécessaire la vitesse en m/s

Les opérateurs sont regroupés par type de fonctions, ainsi toutes les fonctions arithmétiques usuelles sur les nombres se trouvent-elles dans le sous menu *Numérique*.

Lorsque les différents objets nécessaires au diagramme ont été sélectionnés, les arcs permettant de les relier sont créés à l'aide de l'outil bobine (cf. Fig.18) en cliquant sur les points d'entrée ou de sortie des différents nœuds à relier.

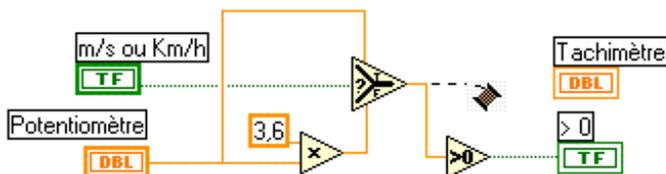


Fig.18 - Création des arcs à l'aide de l'outil bobine

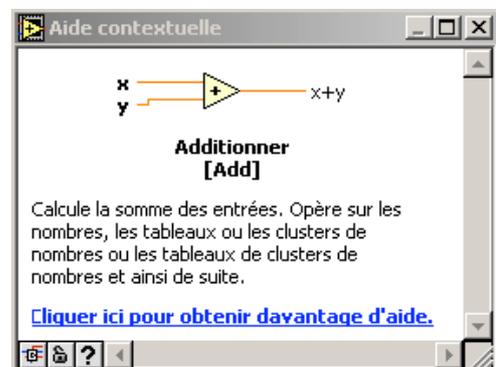


Fig.19 - Fenêtre d'aide

Il est possible de faire apparaître une **fenêtre d'aide** (cf. Fig.19) en suivant la séquence de menu *Aide/Aide contextuelle*. (raccourci : **CTRL+H**). Cette fenêtre permet de connaître quels sont les points d'entrées/sorties du VI, ainsi qu'une description de leurs fonctions.

Types de données :

LabVIEW propose des types de données de base (cf. Fig.20). **Chaque type** est repéré à l'aide de **couleurs spécifiques**. Les arcs d'un type donné ont la couleur de ce type, leur représentation peut varier ainsi. Tableau 1D : fil épais, tableau 2D : fil double (Fig. 21)

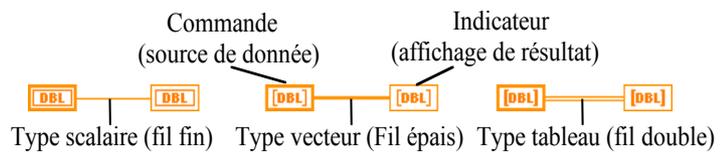


Fig.21 - Différentes structures de données

Fig.20 - Différents types de données

3.1.4.3 Encapsulation

Pour encapsuler un VI, afin de pouvoir l'utiliser comme sous-VI, il faut définir le nœud qui lui correspond ainsi que ses entrées/sorties.

On peut écrire directement un sous VI comme on écrirait une procédure en langage C. On peut également fabriquer un sous VI en sélectionnant une zone dans un diagramme.

Ecriture directe :

- 1) réaliser la face avant et le diagramme.
- 2) Faire apparaître les connexions du nœud correspondant à un VI en faisant un clic *droit* sur l'icône de la fenêtre *face avant* du VI (coin haut droit) et en choisissant le menu *Visualiser le connecteur* (cf. Fig.22).
- 3) Définir les entrées/sorties du nœud correspondant en cliquant alternativement sur les connecteurs et sur les objets de la *face avant*, Si le nombre de connecteurs ne convient pas, le changer à l'aide du menu *Modèles*, en cliquant sur l'icône du VI en haut à droite de l'écran.
- 4) Générer un dessin à l'intérieur de l'icône du VI à partir de l'éditeur d'icône situé dans l'icône de la face avant du VI

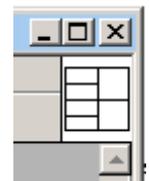


Fig.22 - Le connecteur

Création par sélection d'une zone du diagramme:

Dans ce cas, il faut sélectionner une zone pour la transformer en sous VI, mais sans inclure les contrôles ou les indicateurs associés aux entrées (Fig. 23). Le choix des connexions est alors implicite et automatique.

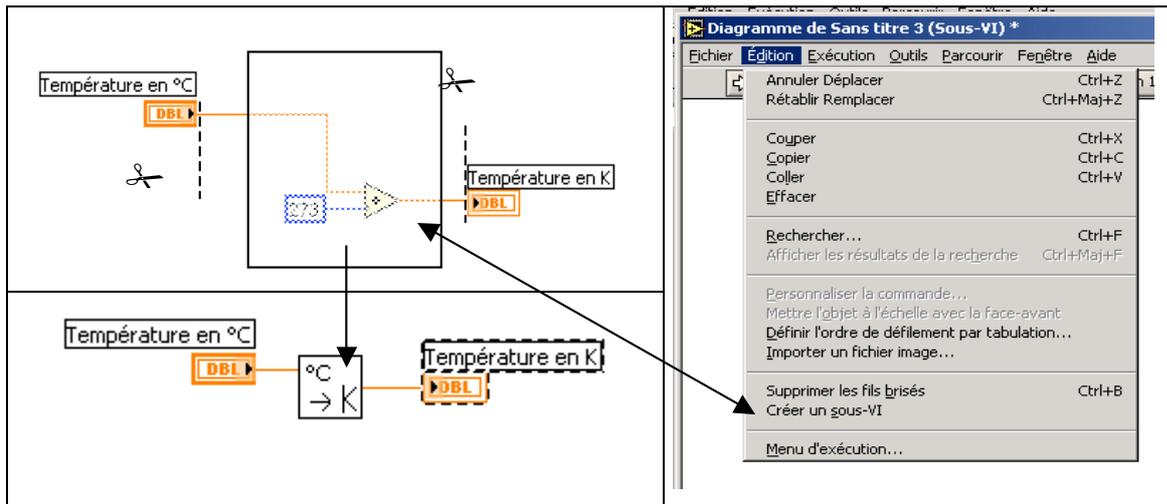
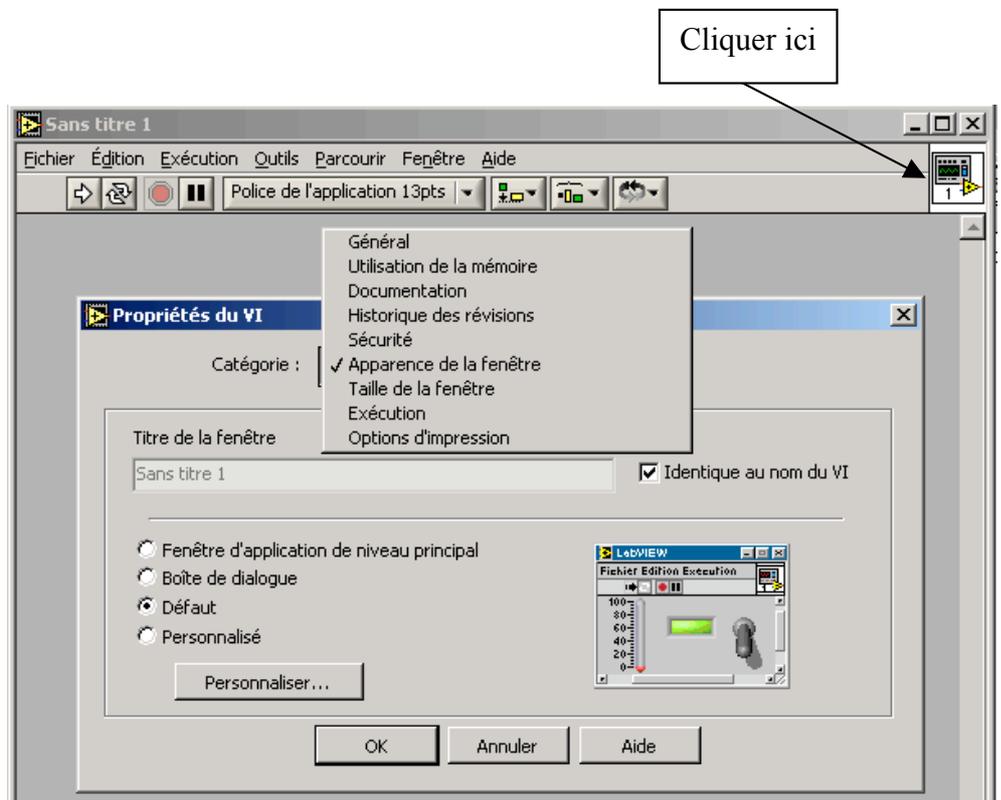


Fig.23 Création automatique d'un sous VI par sélection de zone

Contrôle des propriétés d'un VI

Le ~~contrôle~~ des propriétés d'un sub VI peuvent être modifiées en cliquant dans son icône en haut à droite de l'écran.

Fig.24 Propriétés d'un VI



4 Eléments de programmation

4.1 Variables de tableaux

Définition : On appelle **tableau** tout ensemble ordonné d'éléments de **nature-type** identique : tableau de réels, tableau de booléens, tableau de graphiques

Création : Déposer l'interrupteur dans la structure de tableau vide crée le tableau d'interrupteur nommé « Tableau 1D ».

Transformation 1D → 2D : tirer vers le bas sur la fenêtre précisant le N° de l'élément de tableau

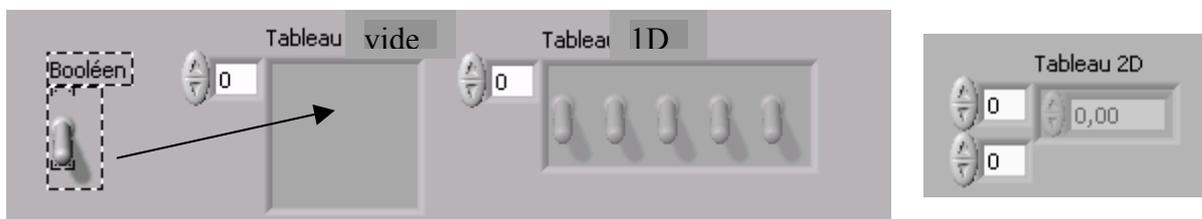
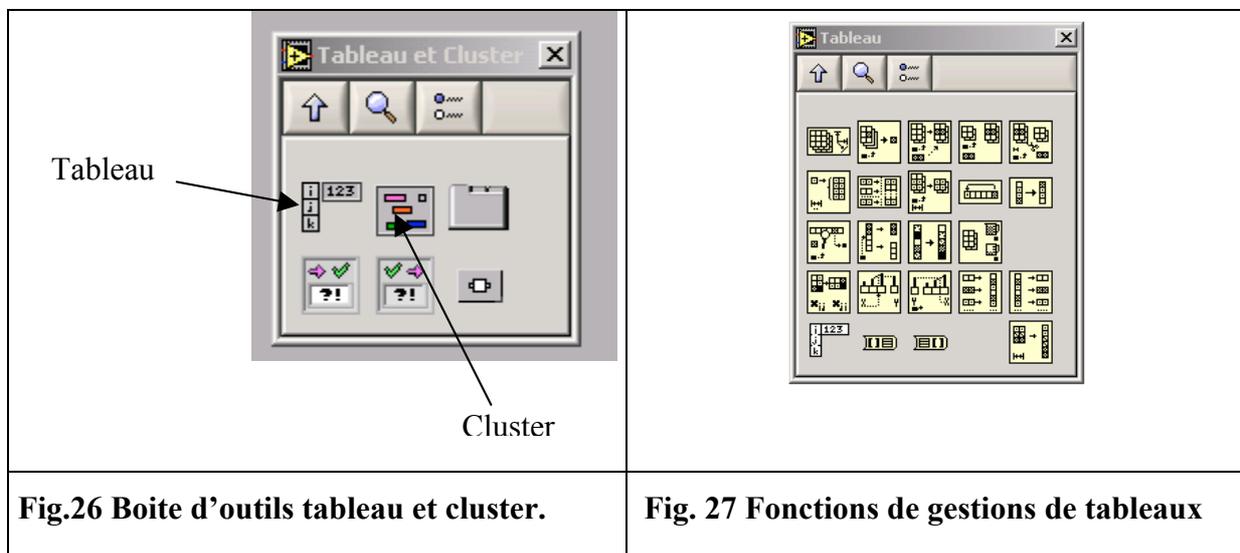


Fig.25 Etapes de la construction d'un tableau



4.2 Clusters

On appelle **cluster** tout ensemble d'éléments de **nature—type** éventuellement **disparate-différent** : entiers, réel, booléens, interrupteurs, graphique ... mais **de-dont type-le mode d'interaction** (commande ou indicateur) **est identique**

4.3 Structures de programmation

Les structures de programmation se trouvent dans le menu *Structures* de la fenêtre *Fonctions* (cf. Fig.28). Ce sont:

- La séquence,
- Le choix,
- La boucle "Pour",
- La boucle "Tant que",
- ~~La calculatrice~~ noeuds de calcul,
- Les variables locales et globales.

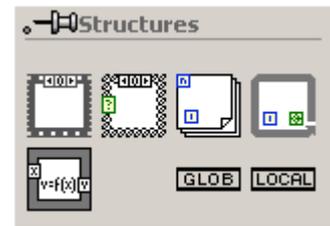
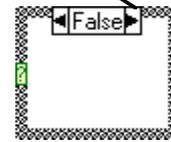


Fig.28 - Sous-menu *Structures*

4.3.1 Structure de choix

Cette structure (cf. Fig.29) permet d'exécuter des flots de données différents suivant la valeur reliée à l'entrée. Cette entrée notée  doit être reliée à un booléen (structure type « si alors sinon ») ou bien à un entier (structure type « **switch** »).

Cliquer ici pour voir le cas **true**



A l'intérieur de chaque fenêtre de cette structure seront édités les diagrammes flot de données correspondant au différent cas.

Fig.29 - Structure de choix

Lorsque le paramètre de choix est un entier ou un type énumération, (accessible par *Commandes/Menu déroulant & énumération /énumération*) on ajoute des fenêtres supplémentaires à cette structure à l'aide d'un *clic droit* sur une des flèches et en sélectionnant le menu "*Ajouter une condition après ou avant*".

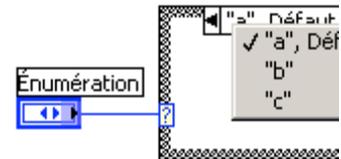


Fig.30 - Structure de choix avec énumération

4.3.2 Boucle "Pour"

Cette structure (cf. Fig.25) permet d'effectuer des calculs itératifs. Le flot de données placé à l'intérieur de cette structure est exécuté N fois. L'indice de boucle i variant de 0 à N-1. (N étant connu).

Lorsqu'un tableau arrive sur une structure de boucle, un **tunnel** est créé: le **tableau est automatiquement indexé** de sorte qu'à chaque itération (l'indexation est représentée par des petits crochets), l'élément d'indice i du tableau est passé dans la boucle.

Cependant il est possible d'envoyer un tableau directement à l'intérieur d'une boucle en faisant un clic *droit* sur le tunnel et en sélectionnant l'option "*Désactiver l'indexation*"

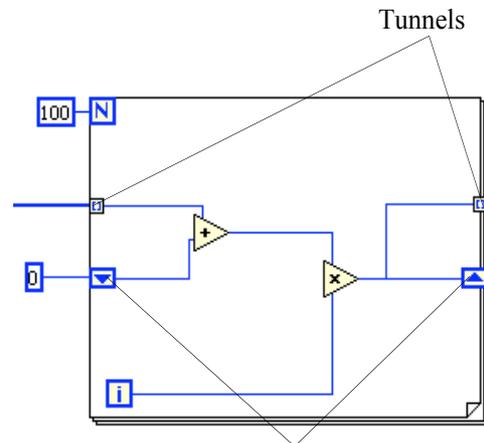


Fig 31 Structure de boucle "Pour"

Inversement lorsqu'une valeur sort de la boucle, elle est automatiquement stockée dans un tableau à l'indice i. Si l'option "*Désactiver l'indexation*" est sélectionnée, seule la dernière valeur est émise.

Registres à décalage

Il est possible de récupérer un résultat provenant d'une itération précédente ($t = T-1$) à l'aide des **registres à décalages** (ou *Shift Register*). Mode de création : faire un clic *droit* sur le bord de la structure de boucle et sélectionner le menu *Ajouter un registre à décalage*..

De plus si l'on « tire vers le bas » l'accroche du registre on accède à la valeur à l'itération T-2 .

Remarque : les registres à décalage **ont besoin d'être initialisés**.

4.3.3 Structure "Tant que"

La structure de boucle "Tant que" permet de faire des calculs itératifs où le nombre d'itérations n'est **pas forcément connu à l'avance** (cf. Fig.26). Une condition **d'arrêt** booléenne doit alors être connectée: la **boucle s'exécute tant que cette valeur booléenne sera vraie**. Une boucle "Tant que" s'exécute toujours au moins une fois.

Comme la boucle "Pour", la boucle "Tant que" possède des registres à décalages et des tunnels, cependant ces derniers sont par défaut en mode "*Désactiver l'indexation*".

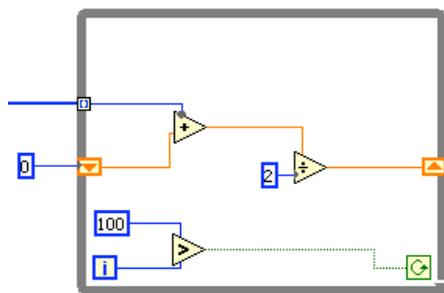


Fig.32 a - Structure de boucle Tant que Avec tunnel indexés

Sortie au bout de 100 tours

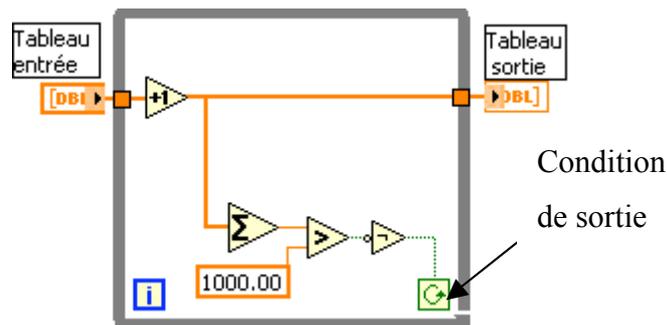


Fig.32 b - Structure de boucle Tant que Avec tunnels désindexés

Sortie si la somme des éléments du tableau est >1000

Remarques :

- un clic sur la flèche de sortie permet de sélectionner **arrêter / continuer sur condition vraie**.
- Toujours **mettre une temporisation** dans les boucles pour limiter la consommation des ressources systèmes

4.4 Techniques de débogage

Lors de l'édition du diagramme, des erreurs peuvent apparaître. Lorsqu'elles surviennent la flèche d'exécution se brise indiquant que le VI ne peut s'exécuter (cf. Fig.33).

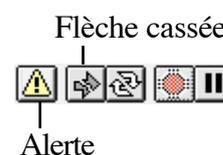


Fig.33 - Signalisation des erreurs

Pour connaître la cause de l'erreur, on peut faire apparaître une fenêtre de report-rapport d'erreur (cf. Fig.34) en suivant le menu *Fenêtres / Liste des erreurs* ou bien en cliquant sur la flèche brisée.

Le symbole d'alerte (point d'exclamation dans triangle) signale un problème potentiel (qui risque de se transformer en erreur). Le report-rapport des alertes se fait également dans la fenêtre de report-rapport d'erreurs. Pour localiser plus facilement une erreur, sélectionner le texte la signalant puis cliquer sur le bouton *Rechercher* ou double-cliquer sur le message d'erreur.

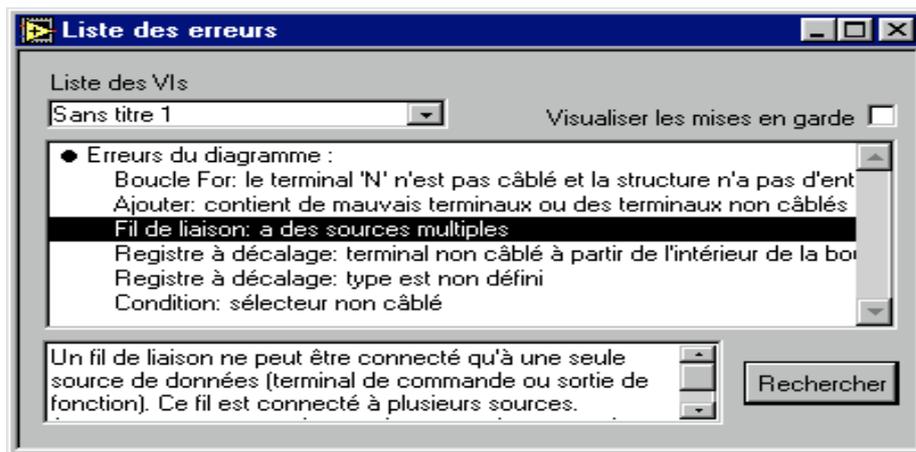


Fig.34 - Fenêtre de ~~report~~ **rapport** d'erreurs

La **flèche d'exécution** peut se briser pour diverses raisons :

- une **e** erreur a été commise et il faut alors y remédier immédiatement,
- un nœud dont la connexion de certaines entrées est obligatoire a été déposé dans le diagramme: la flèche est brisée jusqu'à la connexion de ces entrées.

4.4.1 Erreurs courantes :

- tentative de relier des éléments incompatibles (un entier à un booléen),
- relier deux *commandes* ou deux *indicateurs* entre eux. Ce type d'erreur se traduit par un fil pointillé (cf. Fig.34) à la place d'un fil continu. Il faut alors enlever le fil ou la portion de fil incorrecte.
- Le raccourci clavier CTRL B efface tous les fils pointillés du diagramme (bad wires).

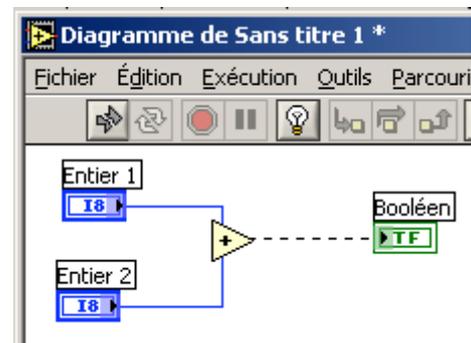


Fig.34 - Erreur de câblage

Appuyer de la **touche animation** (cf. Fig.35) de la barre d'outil de la fenêtre *diagramme* pour **visualiser les jetons** circulant sur les arcs du diagramme flot de données.



Fig.35 - Animation

4.4.2 Sonde et point d'arrêt

Il est possible d'afficher de façon non fugitive la valeur d'une variable, pour cela faire un clic droit avec la bobine de fil sur la connexion concernée.

Il est également possible de placer un point d'arrêt sur une connexion (Fig 36).

Le mode suppression est identique au mode de création.

Les points d'arrêt peuvent également être posés avec le gros point rouge dans la palette d'outils.

Cliquer sur la flèche habituelle de la barre d'outils pour revenir au déroulement normal continu

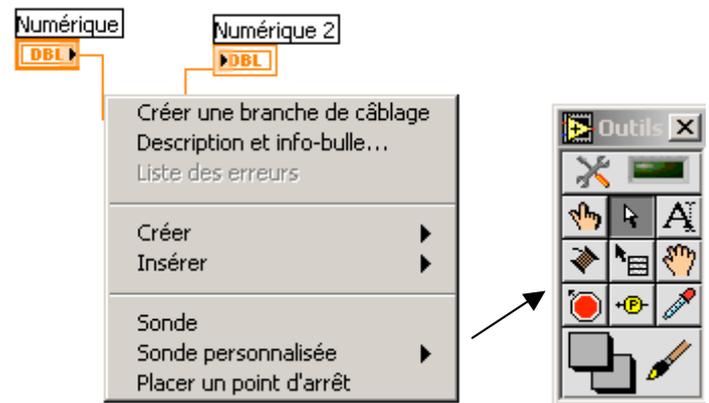


Fig.36 Création d'une sonde et point d'arrêt

4.5 Sauvegarde des ~~Vis~~VI

La sauvegarde d'un VI se fait à l'aide de la séquence de menu *Fichier/Enregistrer*.

L'option *enregistrer avec option/distribution pour développement* permet de créer une bibliothèque autonome permettant de sauver à la fois le programme principal et tous les sous VIs. Il est souhaitable d'effectuer cette opération toutes les fois que le programme est remanié en profondeur, pour pouvoir revenir en arrière si besoin est...



Fig 37 Enregistrement avec option

4.6 Mémento sur le tracé de courbes

LabVIEW offre de nombreuses possibilités pour tracer différents types de courbes. On trouve les différents afficheurs de courbes dans le menu *Graphe* de la fenêtre *Contrôle*. Leur utilisation est résumée ci-après.

4.6.1 Graphe déroulant

L'indicateur *Graphe déroulant* permet de tracer des courbes en fonction du numéro du point (donc sans référence au temps). Les données (de type scalaire, indiqué par un fil fin) sont envoyées au fur et à mesure sur ce type de graphe. On peut également constituer d'abord le bloc de données (vecteur de numériques) et l'envoyer vers le graphe :

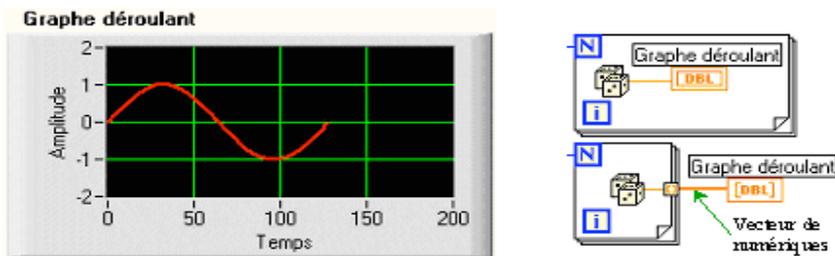


Fig.38 Exemple de graphe déroulant

Pour avoir plusieurs courbes sur un même graphe, il faut les assembler à l'aide de la fonction *Assembler* (menu *Cluster* de la fenêtre *Fonctions*):

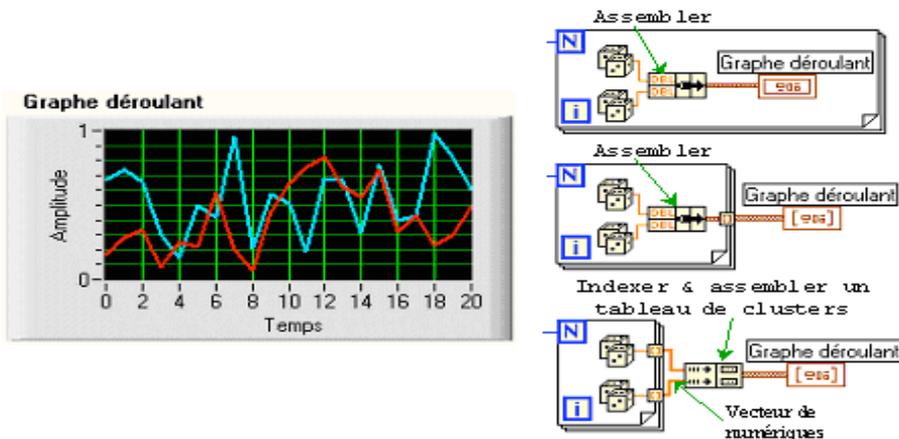


Fig 39 a
Tracé en temps réel

Fig 39 b
Tracés en temps différé

4.6.2 Graphe « oscilloscope »

L'indicateur *Graphe* permet de tracer des courbes à la manière d'un oscilloscope. Il accepte trois types de données différents :

- Vecteur de numériques représentant les ordonnées des points à tracer (fig.40a) ci-dessous);
- Cluster de deux numériques (t_0 et Δt) et un vecteur de numériques où les deux numériques représentent dans l'ordre t_0 , l'instant de début de la mesure, et Δt , l'écart de temps entre deux mesures (fig. 40b ci-dessous). Ce cluster est obtenu à partir de la fonction *Assembler* de la sous-palette *Cluster* ;
- Vecteur de données de type *Waveform*. Ce type de données intègre de façon implicite les caractéristiques du cluster précédent, c'est-à-dire un vecteur de numériques représentant les coordonnées des points de mesures, l'instant initial t_0 et le temps entre deux mesures Δt (fig.40c) ci-dessous). Cette structure de données spécifique "Waveform" peut être visualisée à l'aide de la fonction *Composantes* d'une waveform de la sous-palette *Waveform* de la palette *Fonctions*.

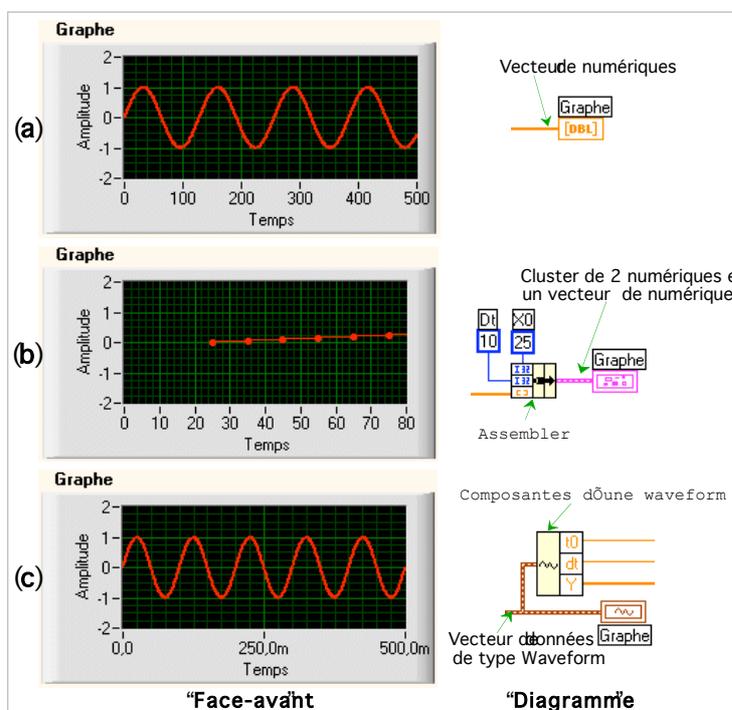


Fig.40 Exemple de graphe de type oscilloscope

A noter que par défaut (sans l'objet *Assembler* ou sans des données de type *waveform*), la courbe est tracée pour $X_0 = 0$ ($t_0=0$) et l'incrément ΔX (Δt) entre chaque point de la courbe est de 1.

Pour avoir plusieurs courbes sur un même graphe, il faut les assembler à l'aide de la fonction *Construire un tableau* (Menu *Tableau* de la fenêtre *Fonctions*). Plusieurs programmations sont possibles (figure (a) ci dessous) :

- Tableau 2D de numériques, réalisé avec la fonction Construire un tableau de la sous-palette Cluster, correspondant aux n courbes à tracer ;
- Vecteur de n clusters d'un vecteur de N numériques où n représente le nombre de courbes et N le nombre de points composant chaque courbe.

Si on désire définir l'instant initial t_0 et le temps entre deux mesures Δt , il faut utiliser la même fonction que précédemment, c'est-à-dire Assembler de la sous-palette Cluster. Au niveau de cette fonction d'assemblage, il est possible d'utiliser les deux structures de données précédentes ; cela conduit aux deux solutions de câblage présentées sur la fig.41b ci dessous. Comme le montre la fig41 c ci dessous, il est aussi possible d'avoir une représentation graphique avec des courbes possédant des instants de début t_0 et des intervalles entre points Δt différents.

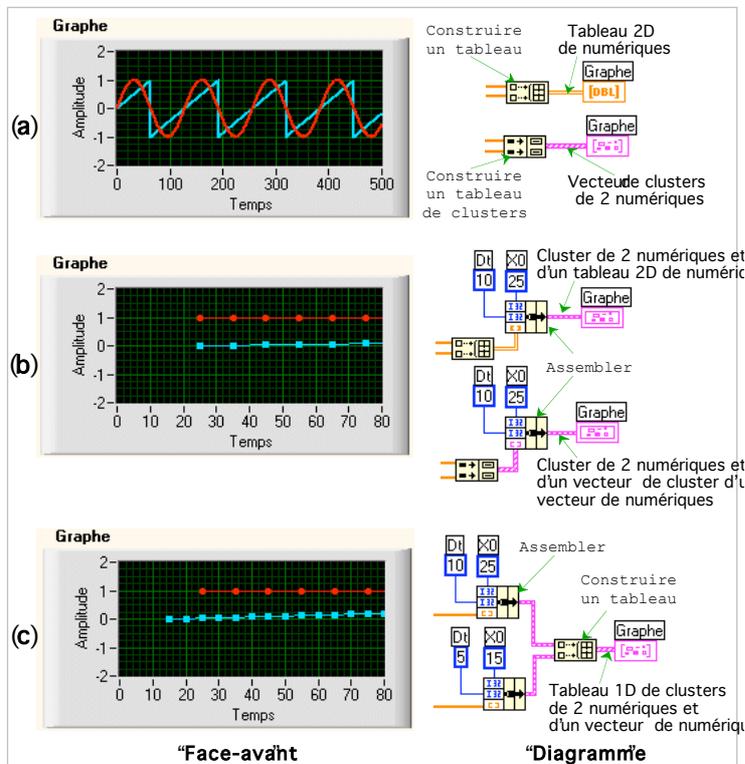


Fig.41 Exemple de graphe de type waveform

4.6.3 Graphe XY

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 (cluster de 2 vecteurs numériques) représentant les composantes X et Y de la courbe. Elle peut aussi être formatée sous la forme d'un vecteur de clusters de 2 numériques. On retrouve par ailleurs le même principe de construction de graphe multi-courbes que dans le paragraphe précédent.

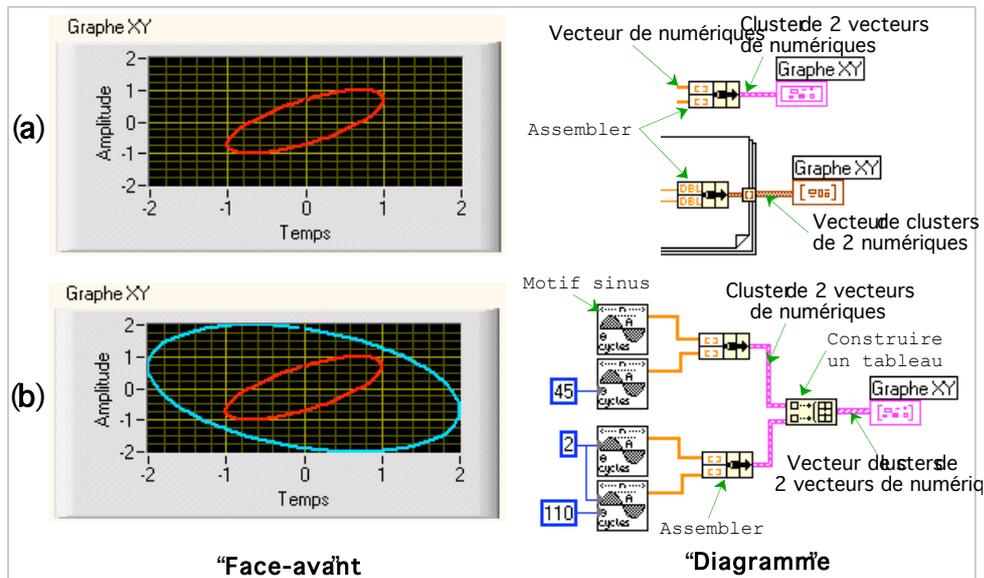


Fig.42 Exemple de graphe de type XY

5 Bibliographie

F. Cottet, « LabVIEW : Programmation et applications », Editions Dunod, ISBN 2100056670

Site National Instruments : NI.com

6 Raccourcis et suggestions

6.1 Raccourcis

Combinaison de touches	Effet produit
CTRL+H	Affichage de l'aide en ligne
CTRL+B	Suppression des « mauvais fils »
CTRL+E	Changement Face avant / diagramme
CTRL + Glisser	Duplication de l'objet
CTRL + étirement zone de sélection	Agrandissement de l'espace disponible dans un diagramme
Barre espace	Commutation bobine / sélection
Tabulation	Commutation bobine / sélection/main/texte

6.2 Suggestions

- Toujours structurer son application, vous gagnerez beaucoup de temps...
- Un VI rentre obligatoirement dans une page d'écran, pour être lisible
- Soigner le câblage de l'application augmente sa lisibilité
- Créer des sous VI, pour modulariser son application
- Sauver souvent vos applications en *mode exportation* avec des noms différents, en rapport avec la date de création
- Chercher des exemples dans la barre de menus / Aide / recherche d'exemples
- Accéder aux manuels de Labview (*.pdf) à partir du site NI.com/manuals
- Consultez sur le site de National Instruments les FAQ (frequently asked questions)
- Si vous butez sur un problème sortez de votre contexte et contactez par mail la hotline de national Instruments (support@ni.com).