

Université Paul Sabatier, Toulouse

INITIATION À
L'INSTRUMENTATION
NUMÉRIQUE :

Utilisation du logiciel LabVIEW 2012
pour la mise en œuvre
de cartes multifonctions
et le pilotage d'instruments

TD d'initiation

Initiation à l'instrumentation numérique

Travaux dirigés

2 séances d'initiation
au logiciel LabVIEW.

Table des matières

1 Informations pratiques	4
1.1 Contenu du document	4
1.2 Objectifs du module	4
2 Généralités sur la mesure	4
2.1 Introduction	4
2.2 Grandeurs à mesurer	5
2.3 Types de mesures	5
2.4 Domaines d'application	6
2.5 La chaîne de mesure automatique	7
3 Mode opératoire	7
4 Introduction à LabVIEW	10
4.1 Constitution d'un VI	11
4.2 Définitions importantes	14
4.3 Exercices	18
4.4 La structure "boîte de calcul"	19
4.5 Les structures boucles et registres à décalage	19
4.6 Exercice	22
4.7 Les tableaux	22
4.8 Exercices	24
4.9 Les structures "condition" et "séquence"	24
4.10 Exercices	26
4.11 Graphes	26
4.12 Exercices	28
4.13 Transformée de Fourier	29
4.14 Exercices	29
4.15 Autocorrélation et mesure d'un retard	30
4.16 Exercices	31

1 Informations pratiques

1.1 Contenu du document

Ce document regroupe une **introduction générale aux thèmes de l'instrumentation numérique**, les textes correspondants aux **TD d'initiation** au logiciel LabVIEW 2012 permettant le pilotage d'instrumentation.

1.2 Objectifs du module

Avant de rentrer dans le vif du sujet, quelques précisions s'imposent. Tout d'abord, même si nous utiliserons un logiciel (LabVIEW) pour réaliser les différentes manipulations, il faut préciser clairement que **l'instrumentation numérique n'est pas de l'informatique**.

Il faut être **capable d'analyser un problème d'instrumentation** et de définir les **méthodes** que l'on compte mettre en œuvre pour résoudre ce problème (carte multifonction, oscilloscope programmable, etc). Ensuite, il s'agit de réaliser et de **tester le fonctionnement des montages** (au multimètre ou à l'oscilloscope par exemple), de **choisir les bons modes opératoires** (fréquence d'acquisition, nombre de voies de mesure), d'apporter un **regard critique sur les mesures et les résultats**. Tout cela relève clairement du travail de **physicien de laboratoire**. C'est sur ces aspects que l'évaluation sera faite.

2 Généralités sur la mesure

2.1 Introduction

Tout scientifique et ingénieur est confronté en permanence au problème de la mesure, de sa transmission, de son traitement et de son interprétation. Dans le processus de recherche ou de R&D, le chercheur/ingénieur est quelquefois conduit à mettre en œuvre une instrumentation complexe à but uniquement **exploratoire**, pour tester des phénomènes mal compris. Dans ce cas, le chercheur/ingénieur espère que l'interprétation des résultats fournira des indices pour bâtir sinon une théorie, du moins un début d'explication du phénomène. Les expérimentations sont souvent conduites avec pour but explicite de vérifier les **prédictions** d'une théorie scientifique. De nombreuses avancées très importantes en Physique découlent directement d'une expérience dont les résultats ne cadrent pas avec les théories disponibles. La vérification expérimentale est ainsi le socle sur lequel s'appuie tout l'édifice des sciences exactes.

En milieu industriel, la mesure et l'instrumentation sont également très répandues. C'est également dans ce milieu que la quasi totalité des capteurs est développée. Bien sûr, l'aspect "exploratoire" est ici absent, si l'on excepte toutefois la recherche "industrielle" ou "appliquée" dont les méthodes sont similaires à celles de la recherche "fondamentale". Par contre, d'autres aspects devront être pris en compte, en particulier la **fiabilité**. Des contraintes économiques peuvent également peser très lourd sur les choix et la mise en œuvre de chaînes de mesure. Quelques pourcents d'erreur sur la quantité mesurée de matières premières utilisées lors d'un processus industriel peut représenter une véritable fortune!

Le thème de l'instrumentation est ambitieux, s'il en est, et l'on voit, regroupées sous ce vocable, de si nombreuses techniques et technologies qu'il devient irréaliste de prétendre les connaître toutes. Néanmoins, dans cet enseignement, l'étudiant trouvera une introduction au thème de **l'instrumentation numérique de haut niveau**, c'est-à-dire l'instrumentation mise en œuvre grâce à des instruments (oscilloscopes numériques, générateurs programmables, cartes multifonction, ...) qui se pilotent grâce à un ordinateur et à un langage de programmation évolué.

2.2 Grandeurs à mesurer

Toutes ces mesures sont de plus applicables à un très grand nombre de paramètres physiques, qui peuvent être :

- **mécaniques** : longueur, vitesse, débit, pression, viscosité, état de surface, puissance, couple, ...
- **électriques** : tension, intensité, puissances, ...
- **thermiques** : température, résistance thermique, ...
- **chimiques** : pH, concentration, salinité, ...

Toutefois, l'élément servant à mesurer les grandeurs désirées (capteur, palpeur, sonde, ...) délivre très souvent (et de plus en plus) un **signal électrique** qu'il faudra récupérer, amplifier et adapter sans déformation (ou alors avec une déformation maîtrisée : linéarisation par exemple). On se ramène donc quasiment tout le temps à **des mesures de tension**. Il est facile ensuite de stocker les signaux, le les traiter et de les restituer à l'aide d'un ordinateur.

2.3 Types de mesures

Les mesures à effectuer en milieu industriel ou dans des laboratoires de recherche universitaire sont extrêmement variées. On peut en effet distinguer plusieurs catégories :

- **mesures simples**, exécutées à la main avec un appareil transportable : elles se font sur le site de l'élément ou du paramètre à mesurer. Exemple : mesure de tension à l'aide d'un multimètre.
- **mesures complexes**, nécessitant un appareillage lourd. Exemple : mesure de consommation spécifique d'un moteur à combustion interne sur un banc moteur. Généralement, ces mesures complexes résultent d'une combinaison de nombreuses mesures plus simples. C'est le cas de nombre de mesures faites dans des labos de recherche fondamentale.
- **mesures multiples**, nécessitant l'acquisition simultanée ou très rapide de plusieurs paramètres : on fait appel à une centrale d'acquisition portable pour des mesures sur site, ou à des appareils indépendants fédérés par un micro-ordinateur (pilotage des appareils et traitement des mesures) en labo. Exemple : mesure de plusieurs paramètres de fonctionnement d'un véhicule (vitesse, régime moteur, consommation instantanée, ...) à l'aide d'une centrale de mesure embarquée.

C'est bien sûr les deux derniers types de mesure (complexe et multiple) qui nous intéressent dans cet enseignement. Les deux sont souvent effectués sur **bancs de mesures automatiques** avec du

matériel piloté par informatique. Ces bancs de mesure informatisés permettent l'acquisition, le traitement et éventuellement le stockage d'un grand nombre de données. Par exemple, on peut vouloir enregistrer la réponse d'un système oscillant en fonction du temps (ressort amorti, balancier) pour pouvoir retrouver ses paramètres en modélisant cette réponse (constante d'amortissement, ordre du système). Dans ce cas, on enregistrera un grand nombre de positions consécutives (par exemple tous les millièmes de seconde pendant 10 secondes). Cette tâche ne peut évidemment pas être réalisée "à la main".

On peut également rencontrer le cas d'une chaîne de **mesure-action en retour**. Par exemple, on peut vouloir réaliser l'asservissement en température d'un matériau. On mesure alors la température à l'aide d'un capteur placé sur ce matériau. Si celle-ci est supérieure à une valeur fixée, on déclenche un ventilateur de chauffe. Ainsi la température est asservie dans une fourchette de valeurs jugées "acceptables".

En fait, ces types de mesures ont une bonne partie de leur mode opératoire en commun, et la mise au point d'un banc de mesure automatique passe souvent par une mise au point en mode manuel.

2.4 Domaines d'application

La finalité de l'instrumentation est très variée, on peut citer notamment :

- **les mesures en laboratoire** : indispensable à tout organisme de recherche (fondamentale ou appliquée), à tout service d'étude, ..., ceci afin de pouvoir élaborer des théories et les vérifier, de concevoir et tester des nouveaux matériaux, composants, produits, ...
- **les mesures de qualification** de produits de grande consommation : avant de lancer un produit en production et sur le marché (automobile, électroménager, Hi-Fi, vidéo, ...), on doit s'assurer qu'il va bien fonctionner sans problèmes pendant une durée de vie minimum, dans divers environnements (chaud, froid, humidité, ...), et qu'il respecte les normes en vigueur. On fait alors fonctionner un échantillon de ces produits (quelques dizaines d'unités dans les cas courants) dans des laboratoires avec du matériel susceptible de les piloter (marche/arrêt, plus/moins vite, ...), de générer des ambiances climatiques extrêmes (froid, chaud, humidité, ...), et de mesurer divers paramètres.
- **les mesures de tests en production** : à toutes les étapes de la fabrication d'un produit, ses caractéristiques sont mesurées et comparées à des valeurs mini et maxi (les tolérances). Le problème est le même que précédemment : les mesures sont faites dans une ambiance très perturbée. Il faut aussi faire face au stockage d'un très grand nombre de données et à leur exploitation, en direct ou en différé.
- **le contrôle de processus industriels** : beaucoup de processus de fabrication industriels sont asservis, c'est-à-dire contrôlés par une ou plusieurs variables, et ce, en temps réel. Il est alors nécessaire de mesurer et de traiter au fur et à mesure de leur acquisition plusieurs paramètres répartis tout au long du processus pour pouvoir agir en conséquence et garantir la qualité et la conformité du produit final. Par exemple, la synthèse d'un produit chimique

peut-être possible dans une gamme de température donnée qu'il faut contrôler et ajuster en permanence pour un déroulement correct de la synthèse.

2.5 La chaîne de mesure automatique

La **chaîne de mesure** est l'ensemble des éléments nécessaires pour connaître la valeur ou l'évolution de paramètres d'un système physique. Le matériel de base sera le même que pour les mesures manuelles, auquel on va rajouter un ou plusieurs de ces éléments :

- un **ordinateur** qui commande les instruments, récupère les données mesurées et affiche les valeurs mesurées (courbes, indicateurs, ...). Il permet le traitement des données **en différé** (stocker ces données, traitement mathématique, affichage de courbes, ...), ou **en temps réel** dans le cas d'un contrôle de processus ou d'asservissement.
- des **instruments de mesure (oscilloscopes, générateurs de fonctions) programmables**, via une liaison série (RS232) ou un bus (GPIB ou autre).
- une **carte multifonctions insérée dans l'ordinateur**. Ces cartes sont très utilisées quand on a beaucoup de mesures à faire en parallèle, car elles possèdent un grand nombre d'entrées configurables par logiciel depuis l'ordinateur (réglage de la sensibilité, du type de mesure simple ou différentielle, ...). Elles peuvent avantageusement remplacer les instruments de mesure programmables.

C'est ce matériel que l'on va utiliser pour les travaux pratiques.

3 Mode opératoire

Avant de mettre en œuvre l'instrumentation proprement dite, et surtout avant de commencer à faire le programme de pilotage de celle-ci, il est indispensable de **définir dans le détail ce que l'on désire réaliser**, en répondant à des questions simples :

- quel matériel dois-je utiliser (oscilloscope, générateur, cartes, etc)
- combien de voies de mesure dois-je utiliser ?
- combien de points de mesure dois-je acquérir ? un seul ? un grand nombre consécutif ? si oui, avec quelle fréquence d'échantillonnage ? plusieurs fois d'affilée ? dois-je faire une mesure "en continu" ?
- que faut-il faire lorsque la mesure est réalisée ? dois-je traiter en temps réel les données pour extraire l'information recherchée ? faut-il sauver les données sur le disque du PC pour un traitement ultérieur ?
- le signal est-il adapté à l'étendue de mesure de ma carte ?
- à quelle fréquence dois-je créer le signal de sortie ?
- et ainsi de suite...

C'est uniquement lorsque l'on aura répondu à ces questions et établi un **schéma**, un **organigramme** de la tâche à réaliser que l'on pourra débiter sa mise en œuvre.

D'autre part, en mesure, il existe un **principe de base** : il faut **rester critique** vis-à-vis de son montage et des résultats observés. Il ne faut **jamais se fier** aveuglément aux indications d'un **appareillage même (et surtout) sophistiqué**. Dans la mesure du possible, on devra essayer

plusieurs types de manipulations permettant de corréler les résultats (par exemple en vérifiant au multimètre le résultat d'une mesure), et comparer autant que possible ces résultats ainsi que les protocoles d'expérimentation avec des collègues ayant fait une expérience similaire. **On ne peut se déclarer satisfait du fonctionnement d'une instrumentation que lorsque l'on aura effectué une série de tests critiques et raisonnés.**

Travaux dirigés (2 séances) :

**Initiation au logiciel d'instrumentation
et de traitement des données**

LabVIEW

Les exercices encadrés sont à réaliser.

Les exemples fournis sont à analyser,
à comprendre et à tester.

4 Introduction à LabVIEW

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) est un logiciel de développement d'application d'un genre nouveau. En effet, contrairement aux autres langages de programmation, LabVIEW n'utilise pas un langage textuel, mais un langage de programmation graphique : le langage G. Les différentes fonctions du programme sont repérées par des icônes reliées entre elles par des fils. LabVIEW permet en outre : (1) de piloter des cartes multifonctions et des instruments munis d'interfaces séries ou GPIB ; (2) d'analyser, de présenter, de traiter et de stocker les données. Ce langage de programmation permet de développer des applications dans le domaine de l'instrumentation numérique sans connaissances approfondies de l'informatique. Les programmes LabVIEW sont appelés “**VI**” (pour “Virtual Instrument”) car il s'agit réellement de “transformer” son ordinateur en un “instrument virtuel” qui regroupe les capacités d'un ou plusieurs oscilloscopes, d'un ou plusieurs générateurs de fonction, ... Un VI utilisé à l'intérieur d'un autre VI est appelé “**sous-VI**”. Chaque sous-VI peut être exécuté de façon autonome.

Les VI sont organisés de la manière suivante :

- Une **face avant** sur laquelle le programmeur va créer l'interface graphique visible (tracé de courbes, indicateurs de valeurs, témoins de mise en service, boutons de commande, commutateurs rotatifs, des LED, ...)
- Un **diagramme** dans lequel le programmeur va dessiner le code de l'application. Pour cela, le programmeur dispose d'une grande quantité de bibliothèques contenant des sous-programmes (ou sous-VI), déjà écrits, qui permettent d'effectuer une grande partie des fonctions les plus utilisées dans le test et la mesure ainsi que le contrôle de processus (acquisition de données, contrôle d'instruments...). Il existe également des bibliothèques contenant les fonctions classiques des langages usuels (boucles FOR, WHILE, manipulation de tableaux, de fichiers, ...).

Lorsque vous démarrez LabVIEW, vous obtenez une fenêtre similaire à celle de la figure 1.



Figure 1 – Fenêtre de démarrage LabVIEW.

Depuis la version 8.0, LabVIEW offre la possibilité de créer et gérer des projets via l'Explorateur de projet. Celui-ci offre aux utilisateurs une visualisation, dans l'environnement de développement, des fichiers dont ils ont besoin pour une application.

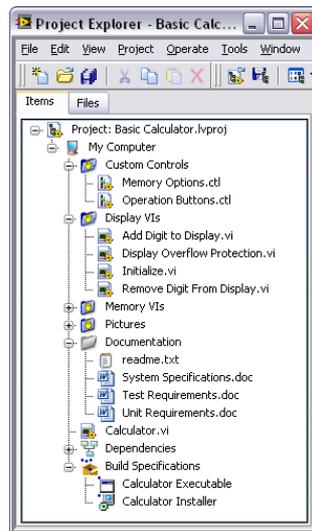


Figure 2 – Exemple de projet LabVIEW.

Pour accéder à l'explorateur de projet, cliquer sur "Create project" dans la fenêtre de démarrage, puis sur "Blank Project".

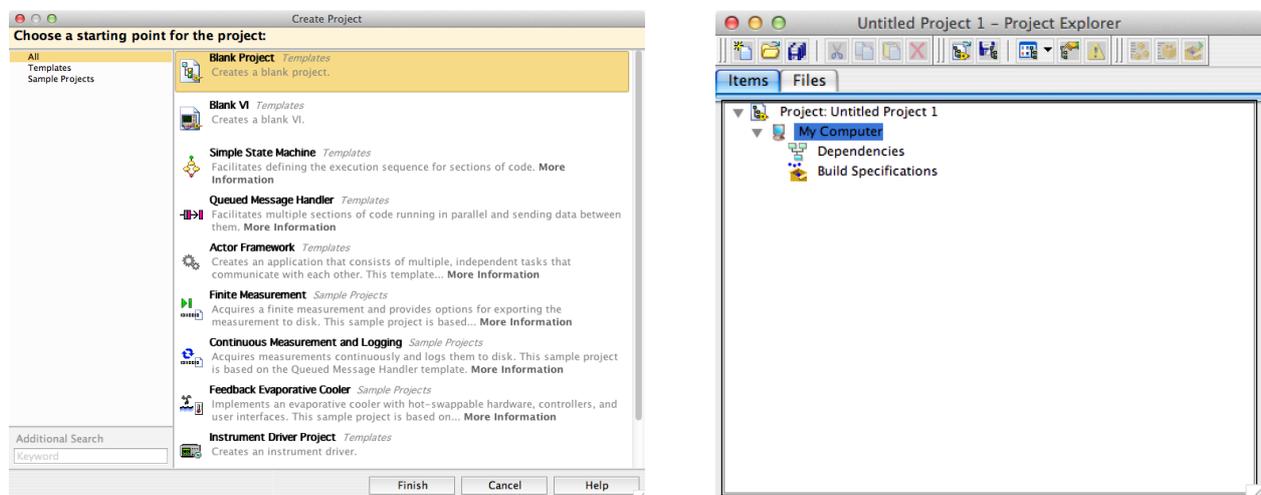


Figure 3 – Accès à l'explorateur de projet LabVIEW.

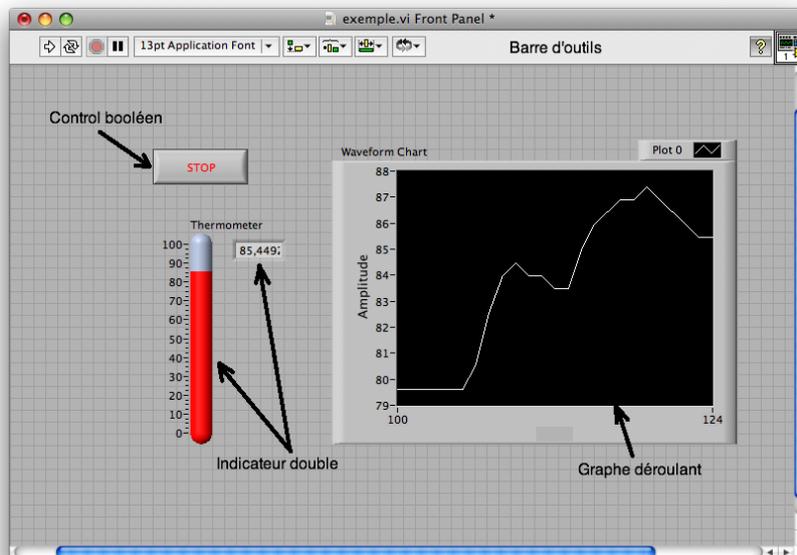
Pour les séances de TD, la création de projet n'est pas nécessaire, nous nous contenterons d'un simple "Blank VI".

4.1 Constitution d'un VI

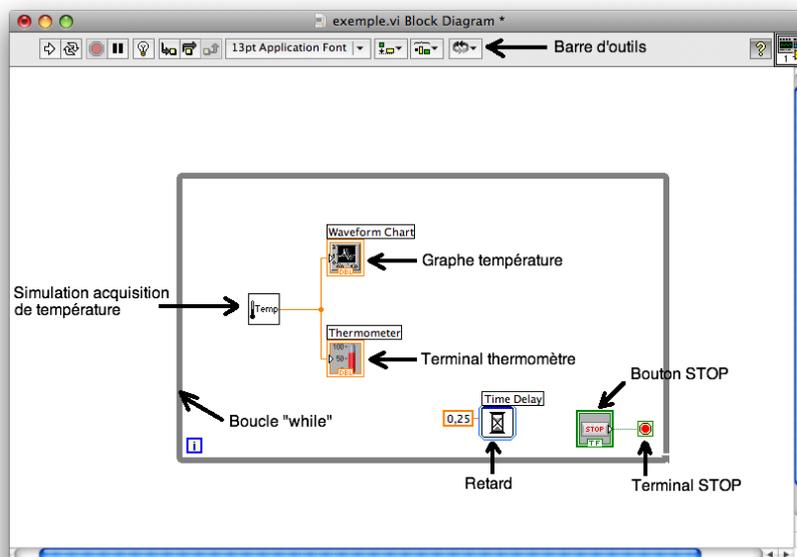
Un VI possède donc 2 fenêtres essentielles :

- La **face avant** sur laquelle on dispose les commandes, les indicateurs, les graphes... (similaire à la face avant d'un instrument). Les différents éléments utiles pour la face avant sont regroupés dans la **palette de commande**.

- Le **diagramme** qui regroupe les différents éléments de programmation G décrivant l'action du VI. Les différents éléments utiles pour la face avant sont regroupés dans la **palette de fonctions**.



(a) La face avant du VI

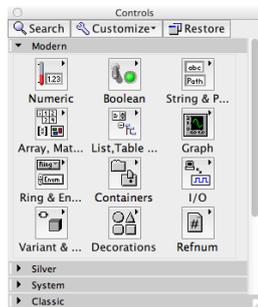


(b) Diagramme du VI

Figure 4 – Exemples de face avant et de diagramme d'un VI LabVIEW.

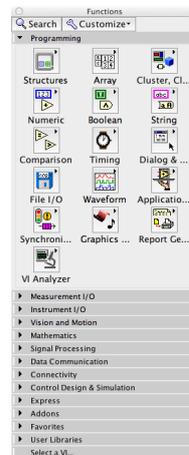
Les outils nécessaires à la programmation et l'exécution des VIs sont regroupés dans la **palette d'outils**. Pour changer d'outils, on peut soit cliquer dans la barre d'outils, soit taper sur la touche "ESPACE". Si on active la petite icône verte en haut de la palette d'outils, la sélection de l'outil se fait automatiquement (par exemple si on approche d'un connecteur, la bobine de liaison apparaît). Une illustration de ces différents éléments est donnée dans les figures 5–7.

Palette de commandes
(associée à la face avant)



Utilisée pour disposer des commandes ou des indicateurs sur la face avant du VI.

Palette de fonctions
(associée au diagramme)



Utilisée pour disposer des opérateurs mathématiques, des outils statistiques, des sous programmes ou des constantes, sur le diagramme du VI.

Figure 5 – Palettes de commandes et de fonctions.



Sélection automatique

- | | | | |
|--|------------------------------|--|-------------------------|
| | Changer un paramètre | | Scrolling tool |
| | Déplacer / changer la taille | | Mettre un point d'arrêt |
| | Ecrire | | Mettre une sonde |
| | Connecter | | Changer la couleur |
| | Raccourci | | Choisir la couleur |

Figure 6 – Palettes d'outils : outils nécessaires à la programmation ou à l'exécution d'un VI

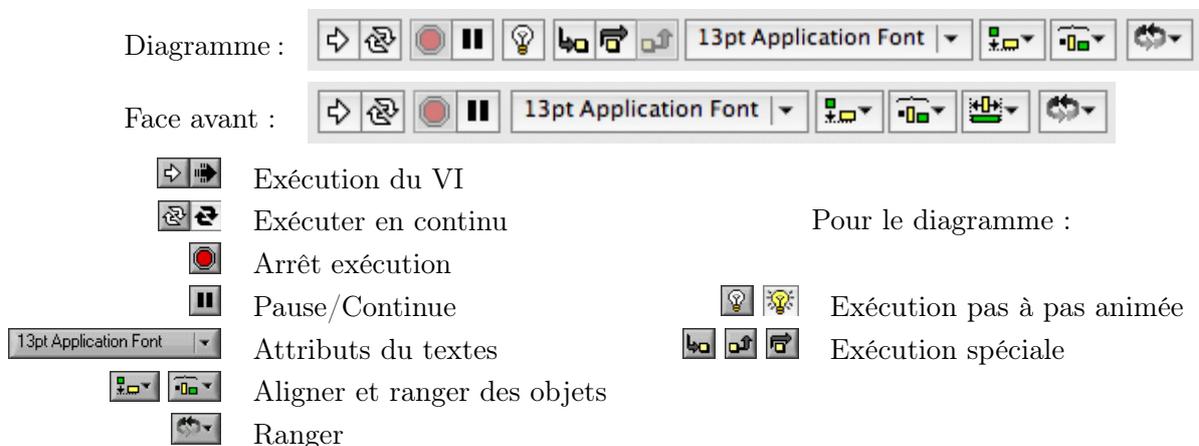


Figure 7 – La barre de status

LabVIEW propose également de nombreux programmes (VIs) qui “simulent” le comportement et les performances d'instruments disponibles dans le commerce, grâce à une carte multifonctions connectée dans l'ordinateur. Dans le cas présenté, le VI correspond à un générateur de fonction Hewlett Packard 34401A. On appelle ces programmes des VI “Instruments”.

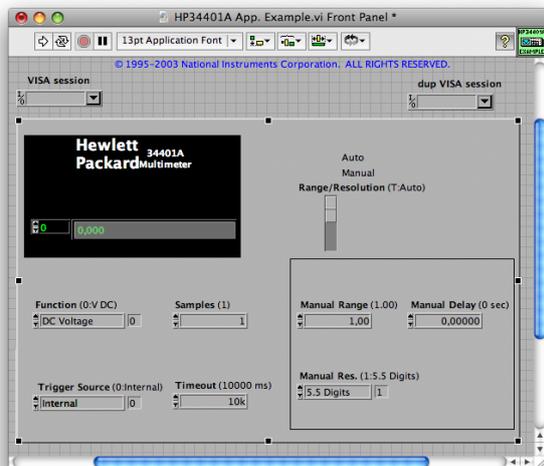


Figure 8 – Un exemple de VI “instrument”. Les VI Instruments servent à : (i) faire l'interface homme-machine (ii) simuler un instrument existant.

4.2 Définitions importantes

1. **Commande** : paramètre d'entrée du VI ou terminal d'entrée du VI. Une commande apparaît à la fois sur la face avant (sa valeur est modifiée en interactif avec l'outil “doigt” ou “caractère” A) et sur le diagramme où elle sert de terminal d'entrée au VI (voir figure 9). A l'exécution du VI, la commande prend la valeur indiquée.
2. **Indicateur** : paramètre de sortie du VI ou terminal de sortie du VI. Un indicateur apparaît à la fois sur la face avant et sur le diagramme (voir figure 10). Un indicateur permet de visualiser le résultat de l'exécution d'un VI ou le contenu d'une variable en cours d'exécution. Il existe des indicateurs de toutes sortes : numériques, tableaux, caractères, graphes, ...
3. **Constantes** : valeurs fixes qui servent lors de l'exécution du VI. Elles n'apparaissent que sur le diagramme (voir figure 11).
4. **Fonctions** : sous-VI prédéfinis, disponible dans la palette de fonctions. Exemple : fonction > numérique > multiplier. Voir figure 12.

Il est possible de faire apparaître le **menu local** d'une fonction, d'une commande ou d'un indicateur en cliquant sur le bouton droit de la souris tout en maintenant sur l'icône correspondant. Ce menu très important permet d'effectuer de nombreuses opérations :

- **aide en ligne** : description des terminaux et de la fonction
- **description** : rentrer sa propre description de la fonction
- **visualiser** : les étiquettes, les terminaux
- **remplacer** : substituer une fonction par une autre
- **créer** : une constante, un indicateur, une commande avec un câblage automatique sur un terminal

Ce dernier point est très important : tout objet peut être câblé automatiquement grâce à son menu local (créer un indicateur ou une commande selon le cas). Ceci permet d'éviter toute une série d'erreur (erreur de type, confusion commande-indicateur, etc). Faire plusieurs tests de ce principe.



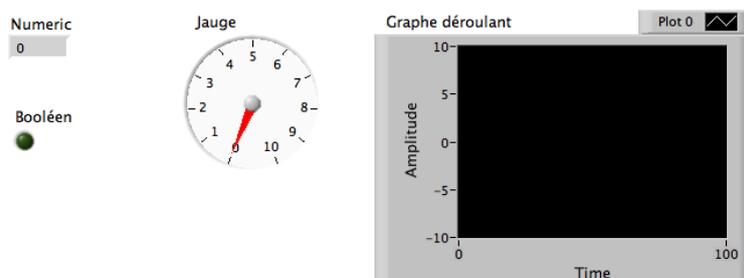
(a) Exemples de commandes sur la **face avant**. On peut redéfinir les bornes d'une commande numérique avec l'outil "doigt" ou "A" : sur le cadran, on a changé 0,10 en -50,50.



(b) Exemples de terminaux des commandes sur le **diagramme**. Les commandes sont encadrées en traits **gras**. L'icône permet de repérer la commande, la couleur traduit le type :

- orange : réels (DBL : double précision)
- vert : logique (TF = True-False ou Vrai-Faux)
- bleu : entiers (I8 : entiers de 8 bits)
- rose : chaînes de caractères (abc)

Figure 9 – Exemples de commandes (dans "palette de commandes").



(a) Exemples d'indicateurs sur la **face avant**. On peut redéfinir les bornes d'un indicateur numérique avec l'outil "doigt" ou "A".

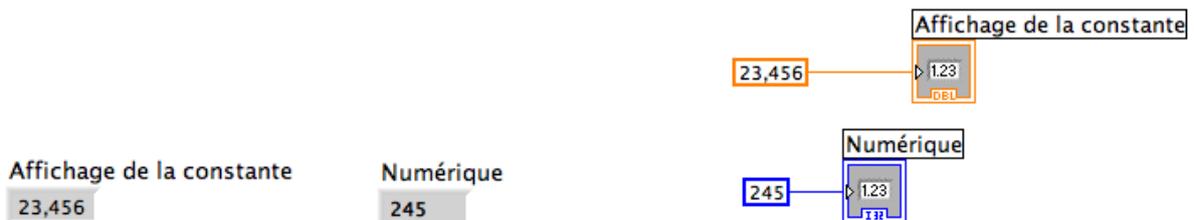


(b) Exemples de terminaux d'indicateurs sur le **diagramme**. Les commandes sont encadrées en traits **fins**. L'icône permet de repérer la commande, la couleur traduit le type :

- orange : réels (DBL : double précision)
- vert : logique (TF = True-False ou Vrai-Faux)
- bleu : entiers (I8 : entiers de 8 bits)
- rose : chaînes de caractères (abc)

Figure 10 – Exemples d'indicateurs (dans "palette de commandes").

5. **Liaisons** : elles permettent le transfert de l'information entre les sous-VI. Elles s'effectuent à l'aide de l'outil "bobine" (voir figure 13). Les caractéristiques des liaisons (forme, épaisseur, couleur) reflètent le type d'information transportée (entier, réel, logique, vecteur, tableau, caractère, ...). Une liaison incorrecte apparaît en tirets (voir figure 14 — attention de ne pas confondre avec la liaison en pointillés qui correspond à une liaison logique, cf. figure 13), barrés d'une croix rouge. On peut détruire tous les mauvais câblages avec **CTRL-B**.



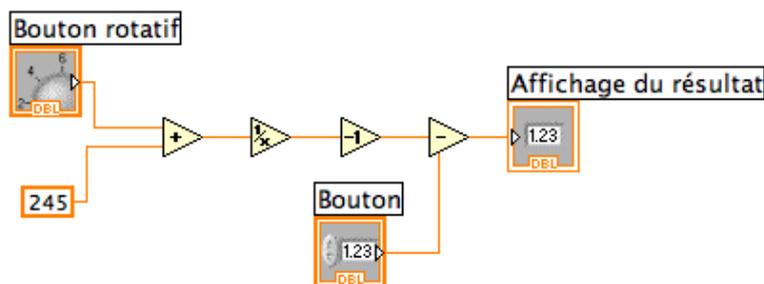
(a) Face avant : les constantes n'y apparaissent pas (seuls les indicateurs de ces commande apparaissent).

(b) Diagramme (code des couleurs identique; pour changer le type et créer un indicateur par "clic-droit", menu local)

Figure 11 – Exemples de constantes, ici, par exemple, les valeurs 23,456 et 245 (dans "palette de fonctions").



(a) Face avant. A l'exécution, l'indicateur affiche le résultat du calcul.



(b) Diagramme

Figure 12 – Exemples de fonctions (Fonctions > Programmation > Numérique).

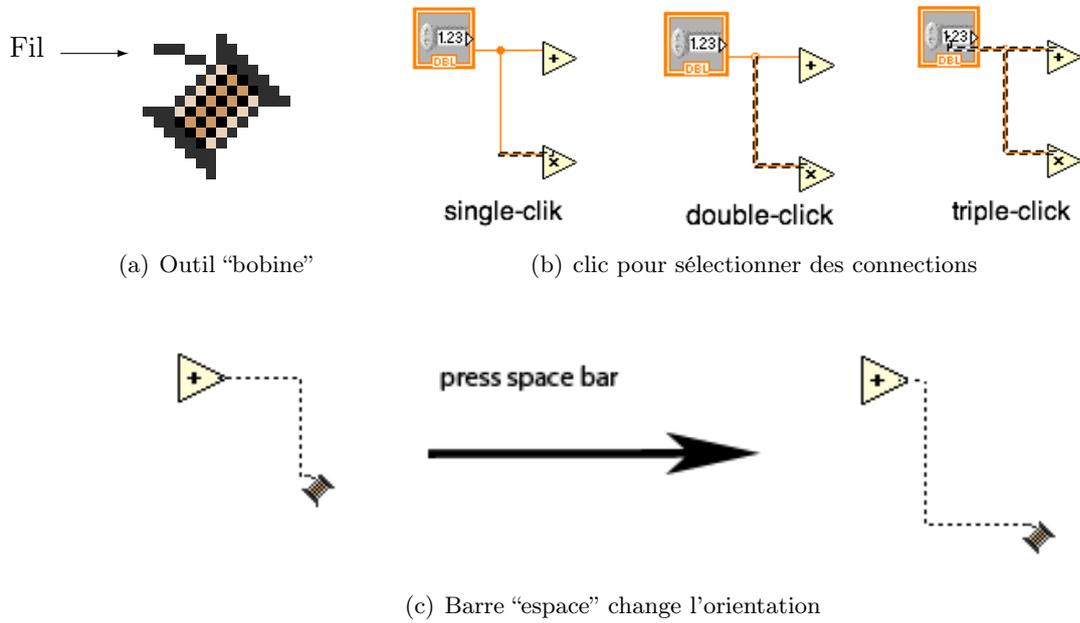


Figure 13 – Liaisons : outil bobine et astuces sur les connections

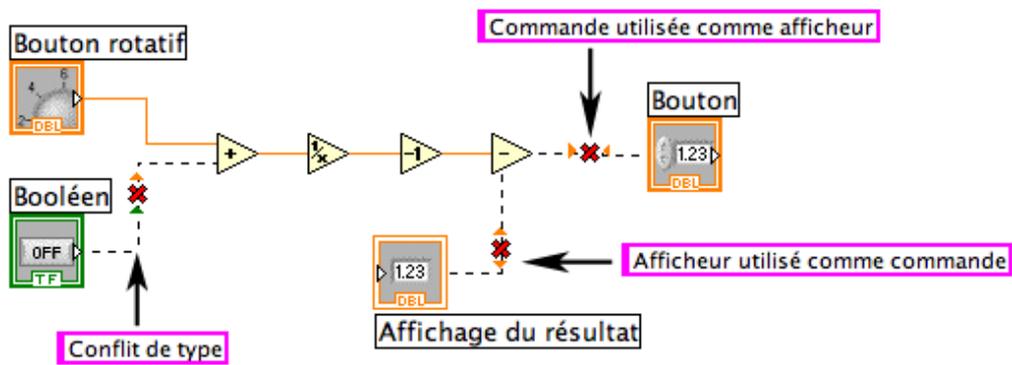


Figure 14 – Exemples de câblages incorrects sur le diagramme

4.3 Exercices

1. Ecrire un VI qui calcule le rapport de deux nombres réels x et y (division réelle) que l'on spécifie par des commandes (**numériques**). On spécifiera $x=76.2$ et $y = 4.5$

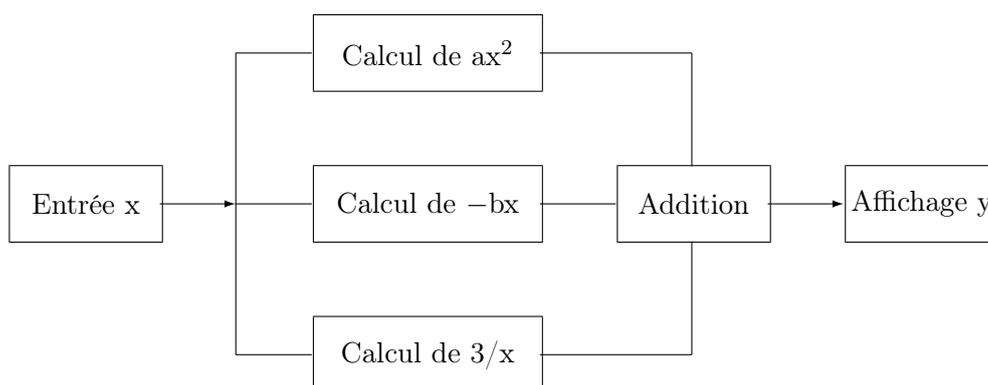
Mode opératoire :

- Disposer deux commandes numériques que l'on renomme " x " et " y ". Disposer un opérateur "diviser". Cabler ces trois éléments. Faire apparaître le menu local de la sortie de l'opérateur "diviser". Actionner : **créer ... un indicateur**. Cette opération crée un indicateur (par défaut un afficheur numérique) qui a automatiquement le type correct et qui est déjà câblé à l'opérateur. Exécuter votre programme.
- Faire apparaître le menu local de l'indicateur qui affiche le résultat. Actionner : **remplacer ...** pour remplacer celui-ci par n afficheur à aiguille. Modifier les bornes de l'indicateur à aiguille pour que celui-ci affiche correctement le résultat. Faire apparaître le menu local de la commande x . Actionner : **représentation ... I8** pour transformer cette commande en une commande de type entier de 8 bits. Vérifier que cette commande est bien entière. Exécuter votre programme.
- Exécuter votre programme "en continu" (les deux flèches). Modifier les valeurs des paramètres en cours d'exécution. Observer le résultat.

2. Faire un VI qui calcule $y = ax^2 - bx + 3/x$ avec x, a et b des réels double précision

- x est une commande de type numérique
- a est une commande de type glissière
- b est une commande de type bouton
- y sera représenté sur un vu-mètre (cadran à aiguille)

Mode opératoire (organigramme simplifié) :



Respecter la disposition des blocs de calcul ci-dessus afin d'obtenir un VI clair, symétrique et harmonieux (et facile à déboguer ...). Exécuter ce programme en continu en modifiant les paramètres.

4.4 La structure “boîte de calcul”

Elle permet d'implémenter une équation mathématique sans utiliser les fonctions numériques. La figure 15 montre un exemple d'utilisation de cette structure. On entre les équations avec l'outil A (éditeur de texte). Les entrées et les sorties (les terminaux) sont ajoutées avec le **menu local de la structure** (clic-droit de la souris placée sur le cadre de la structure).

Chaque ligne de l'équation doit se terminer par “;” .

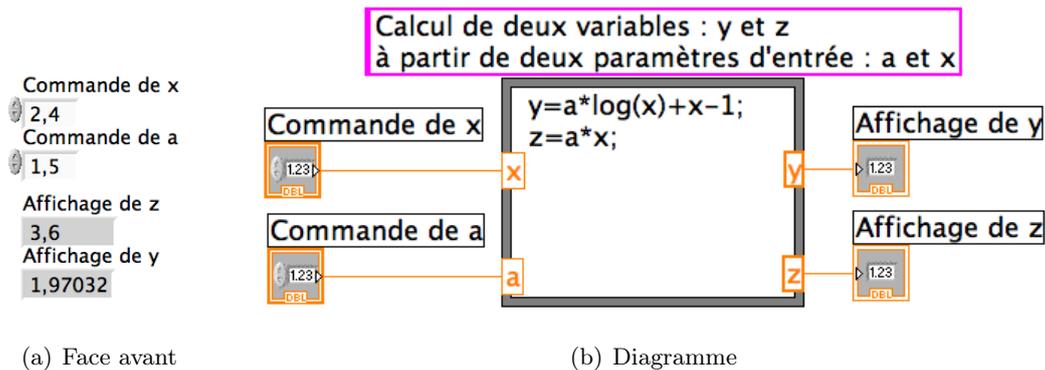


Figure 15 – Calculs avec une boîte de calcul.

4.5 Les structures boucles et registres à décalage

Comme pour les langages de programmation classique (Matlab, C, ...), il est possible de faire des boucles avec LabVIEW. Les objets graphiques permettant de réaliser les boucles sont dans la palette fonction > structures. Les deux principales boucles qui seront utilisées sont la boucle “FOR” et la boucle “WHILE”. Ces boucles ainsi que les registres à décalage sont décrits ci-dessous.

1. La structure “**WHILE**” : implémente le pseudo code “Exécuter ce qui est à l’intérieur de la boucle tant que la condition est vraie”. Cette structure est munie d’un terminal d’itération qui indique le nombre de passage dans la boucle. Un “booléen” (bouton “stop”) est automatiquement connecté au terminal conditionnel. Il peut bien sûr être remplacé par un test sur une valeur. Voir les figures 16 et 17.
2. La structure “**FOR**” : implémente le pseudo code “Pour i valant de 0 à N par pas de 1, exécuter ce qui est à l’intérieur de la boucle”. Le nombre d’itération (N) est défini à l’extérieur de la boucle. Voir les figures 18 et 19.
3. **Les registres à décalage** : ces registres permettent de stocker des informations (scalaires, vecteurs, tableaux, chaînes de caractères) qui circulent et/ou sont modifiées dans des boucles au cours des itérations (voir figures 20, 21 et 22). La mise en place d’un registre à décalage s’effectue en plaçant le curseur de la souris sur le bord de l’objet boucle et en cliquant sur le bouton droit (**menu local de la structure**).

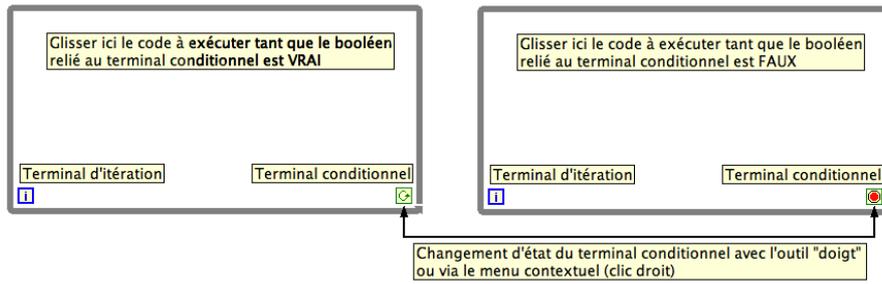


Figure 16 – La structure WHILE.

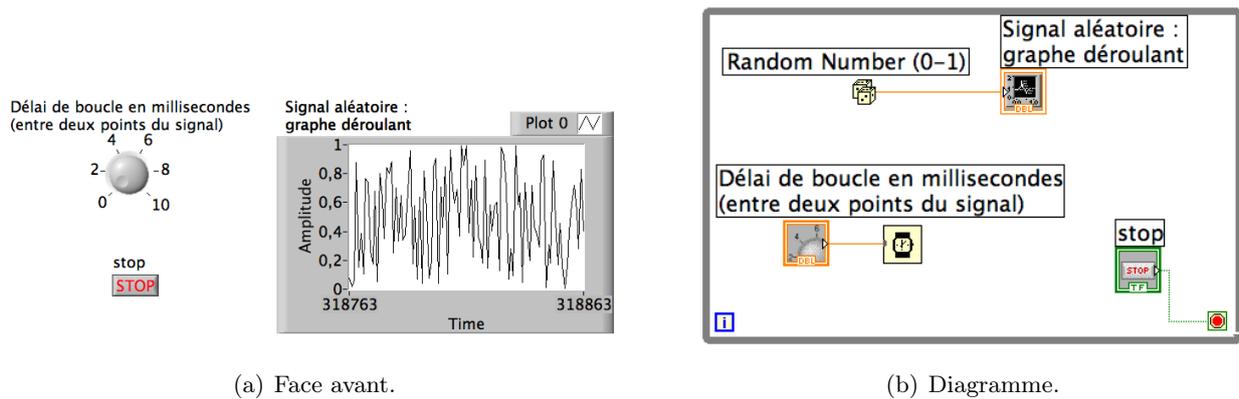


Figure 17 – Exemple d'utilisation de la boucle WHILE.

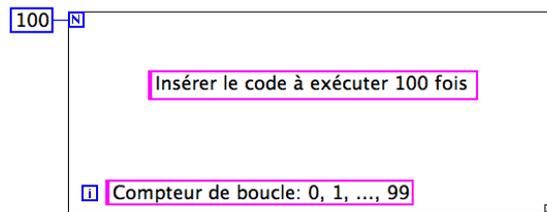


Figure 18 – La boucle FOR.

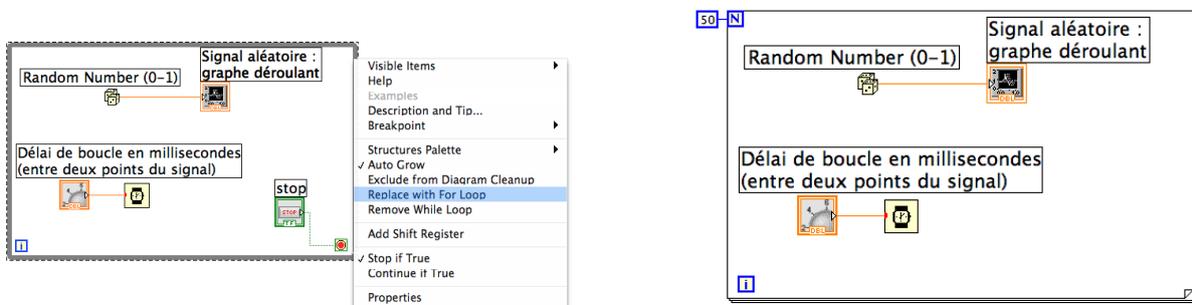


Figure 19 – Équivalent de l'exemple de la figure 17 en remplaçant la boucle WHILE par une boucle FOR.

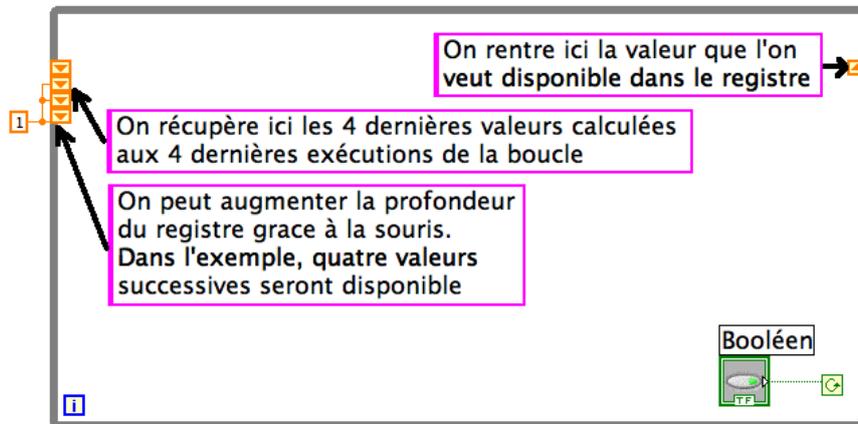


Figure 20 – Registre à décalage

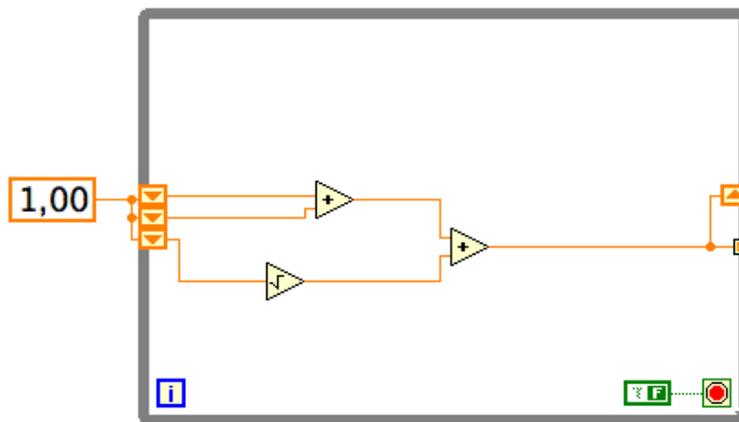
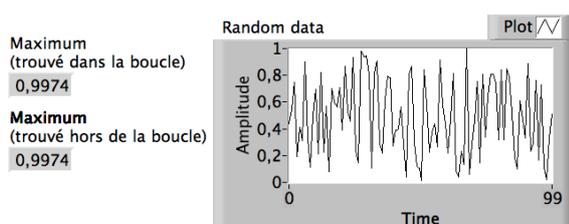
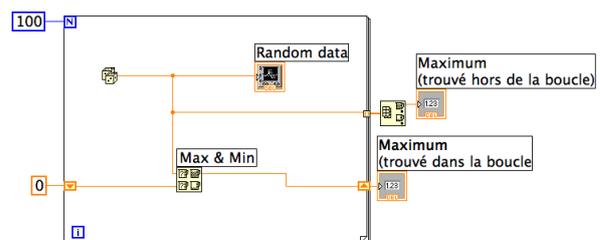


Figure 21 – Exemple d'utilisation de registre à décalage. **Exercice : que fait ce programme ?**



(a) Face avant



(b) Diagramme

Figure 22 – Exemple d'utilisation de la boucle FOR avec des registres à décalage.

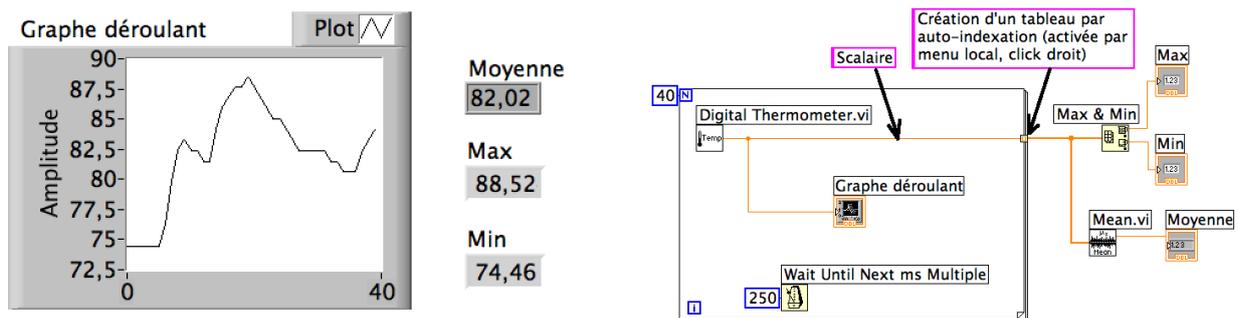
4.6 Exercice

On considère les éléments de la suite $x(n) = x(n-1) + x(n-2)$ avec $x(0) = 1$ et $x(1) = 1$. Cette suite est appelée "suite de Fibonacci".

1. Utilisation d'une boucle "WHILE" :
 - calculer un élément toutes les 0.1s
 - arrêter le calcul quand l'utilisateur le souhaite, à partir d'une commande STOP
 - afficher la suite x dans un graphe déroulant
2. Modifier votre VI pour utiliser une boucle "FOR" :
 - calculer les 40 premiers éléments de la suite
 - afficher la valeur de $x(n)/x(n-1)$. En déduire la limite de cette valeur lorsque n tend vers l'infini

4.7 Les tableaux

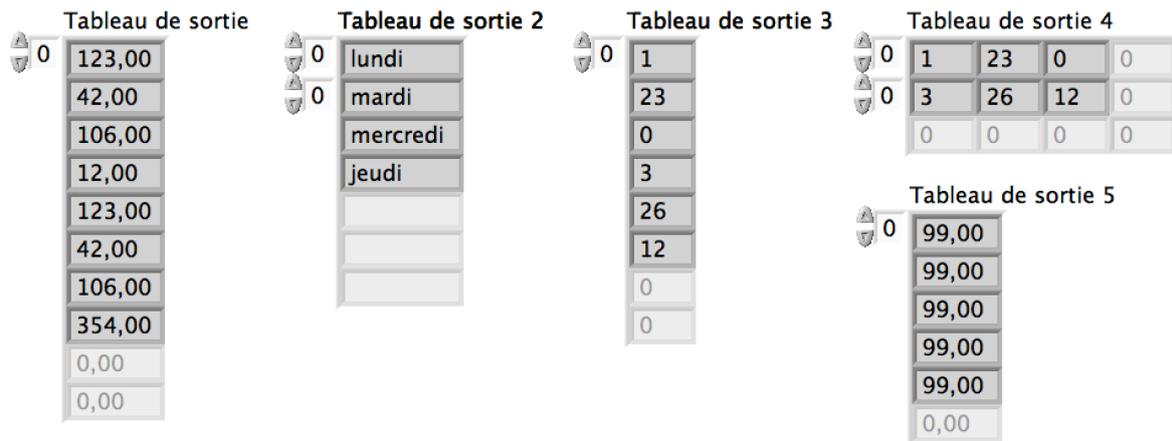
Plusieurs méthodes peuvent être utilisées pour définir un tableau de valeurs. Certaines fonctions (ex. : sinus, impulsion, ...) fournissent directement un tableau de N valeurs. Les valeurs calculées dans une boucle (FOR en particulier) peuvent être placées dans un tableau si l'auto-indexation est activée (voir figure 23). Un scalaire dans une boucle qui est sorti de cette boucle par un fil devient automatiquement un tableau contenant toutes les valeurs successives de ce scalaire. Ce principe s'appelle : **auto-indexation**. Pour obtenir uniquement le dernier élément en sortie de boucle, il est nécessaire de désactiver cette auto-indexation par le **menu local du terminal** de sortie de boucle. Ces méthodes sont illustrées dans les figures ci-dessous (figure 24) de même que quelques remarques sur le polymorphisme des fonctions (figure 25) et quelques outils spécifiques aux tableaux, par exemple sous-ensemble d'un tableau, taille d'un tableau (figure 26).



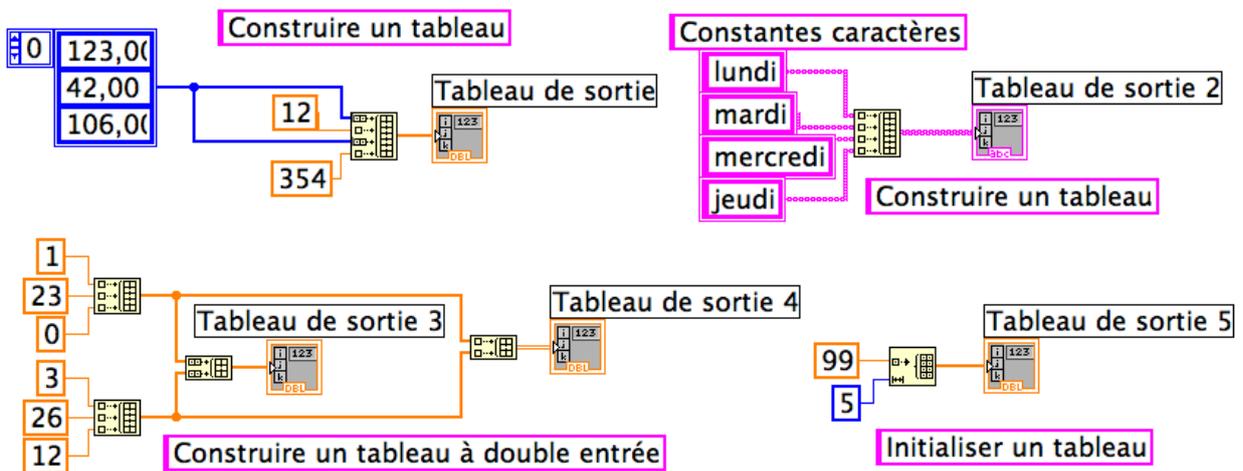
(a) Face avant.

(b) Diagramme.

Figure 23 – Exemple de création d'un tableau avec la boucle FOR.

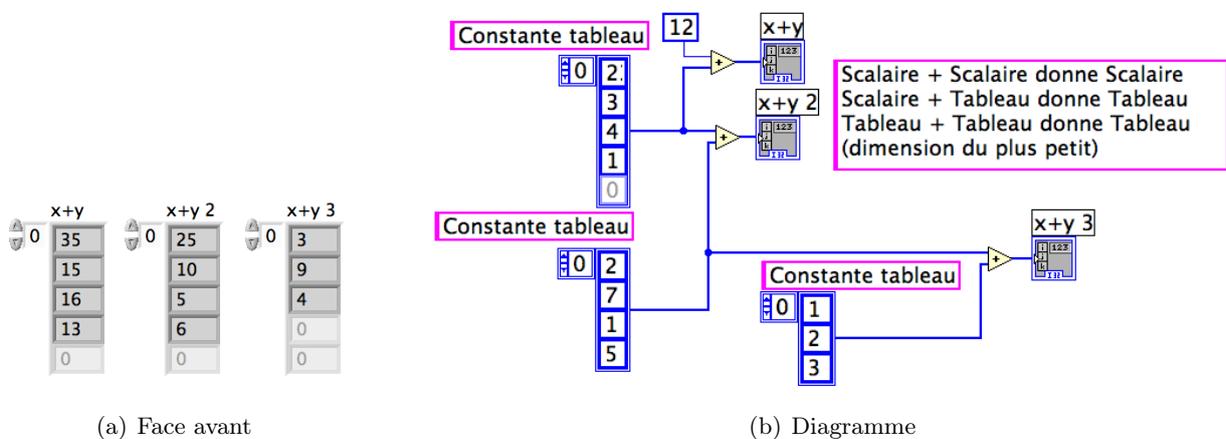


(a) Face avant.



(b) Diagramme.

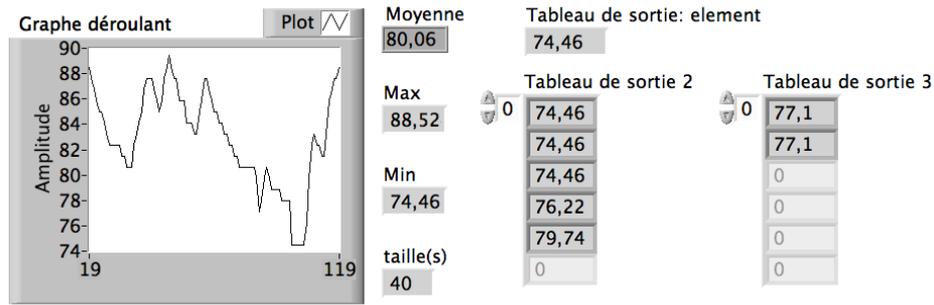
Figure 24 – Exemple de définition d'un tableau.



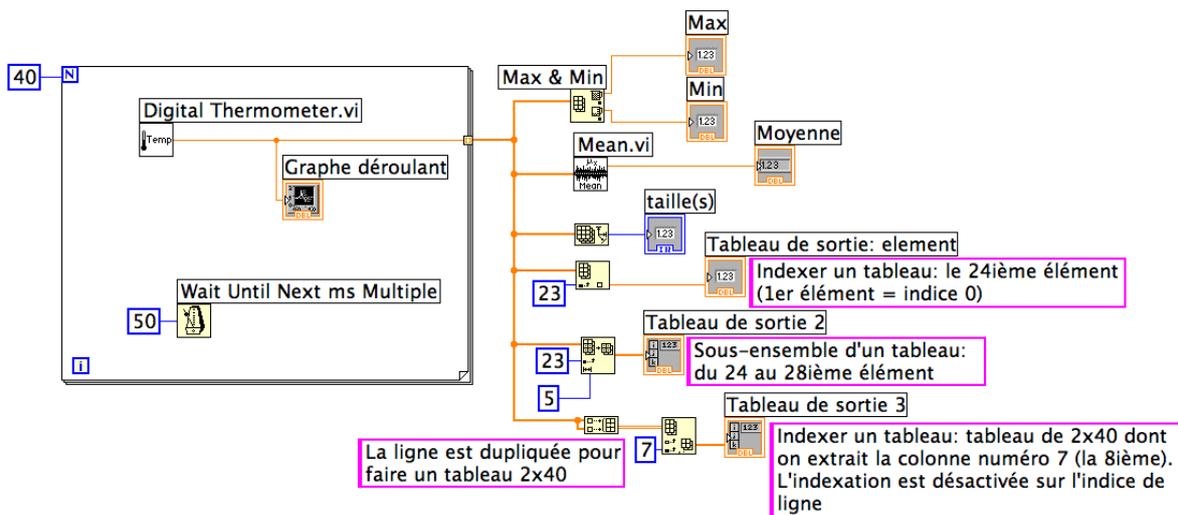
(a) Face avant

(b) Diagramme

Figure 25 – Remarque sur le polymorphisme des fonctions.



(a) Face avant



(b) Diagramme

Figure 26 – Outils de manipulation des tableaux.

4.8 Exercices

On considère à nouveau la suite de Fibonacci : $x(n) = x(n - 1) + x(n - 2)$ avec $x(0) = 1$ et $x(1) = 1$. Ecrire un VI de manière à obtenir :

- un tableau qui contient les 15 premiers éléments de la suite (par auto-indexation)
- un tableau qui contient les 5 premiers éléments de la suite (par extraction du tableau précédent)
- un indicateur qui contient l'élément numéro 8
- un indicateur qui contient la taille du tableau
- un indicateur qui contient la moyenne des éléments du tableau

4.9 Les structures “condition” et “séquence”

1. La structure “condition”

Dans le cas le plus simple, cette structure possède deux volets exécutés selon une condition.

L'entrée de la condition ou terminal de sélection est repéré par le symbole "?". Cette entrée reçoit soit un booléen pouvant résulter d'une opération logique (voir l'exemple présenté figure 27), soit un entier.

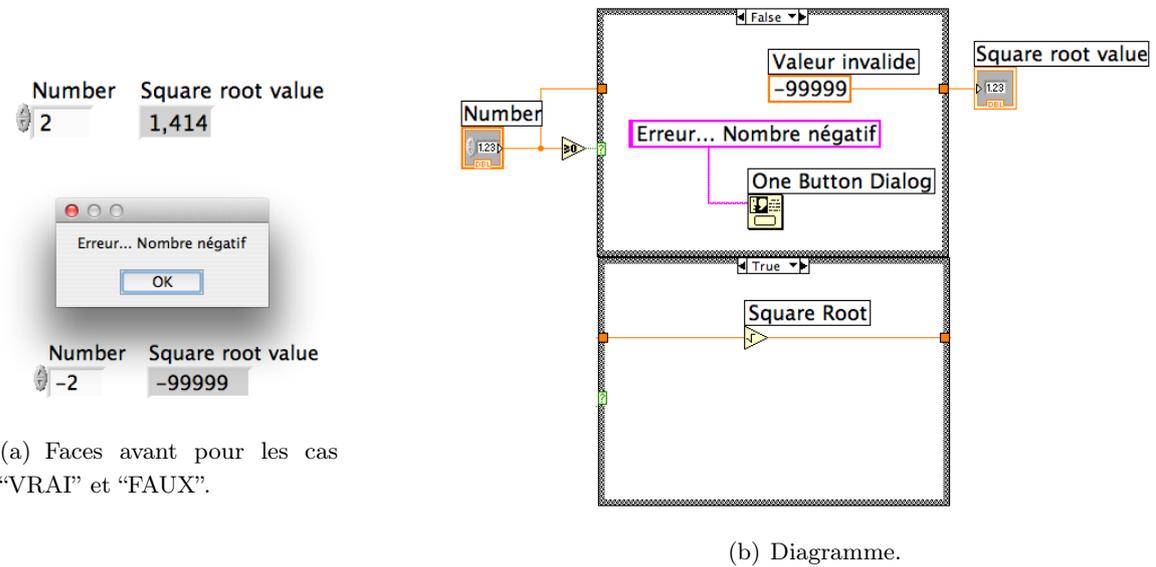


Figure 27 – Exemple d'utilisation de la structure "condition".

2. La structure "séquence"

Certaines opérations doivent être effectuées l'une après l'autre – séquentiellement. La structure "séquence" permet de définir les différentes étapes des opérations sur différents volets. La figure 28 présente un exemple d'utilisation de cette structure.

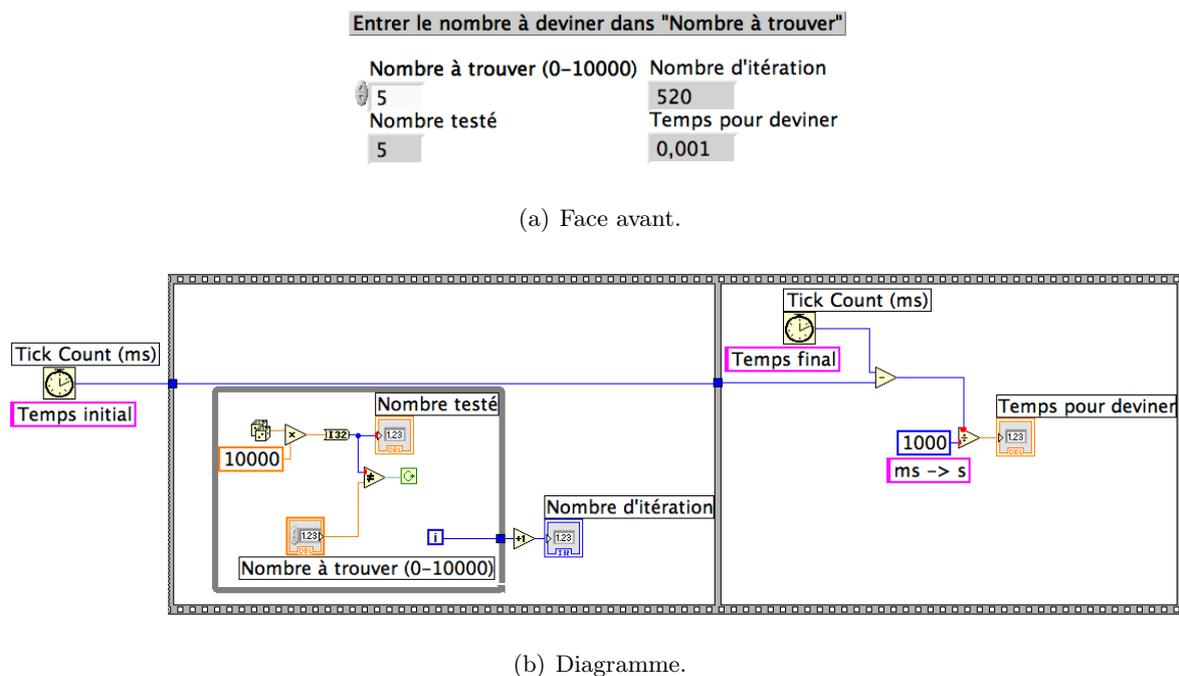


Figure 28 – Exemple d'utilisation de la structure "séquence" en mode déroulé.

4.10 Exercices

1. Calculer le temps mis par un programme pour évaluer la fonction $f(x) = \sin^2(x) + \sin(x) - 1$, en 10^5 point couvrant uniformément l'intervalle $[-2; +2]$.
2. Ecrire un VI qui calcule :

$$\begin{cases} y = -x & \text{si } -0.5 < x < 0.5 \\ y = x - 1 & \text{sinon} \end{cases}$$
 pour x variant de -2 à 2 par pas de 0.01 .

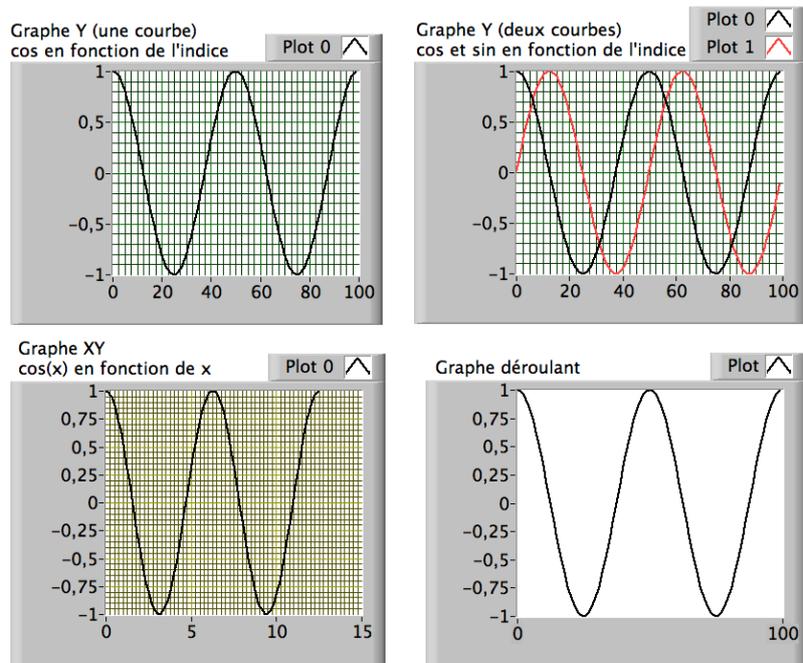
4.11 Graphes

Les graphes sont des indicateurs accessibles par la palette de commande. Il en existe plusieurs types : graphe déroulant, graphe, graphe X-Y, graphe 3D, ... La figure 29 donne un exemple des trois premiers types, qui seront les plus utilisés :

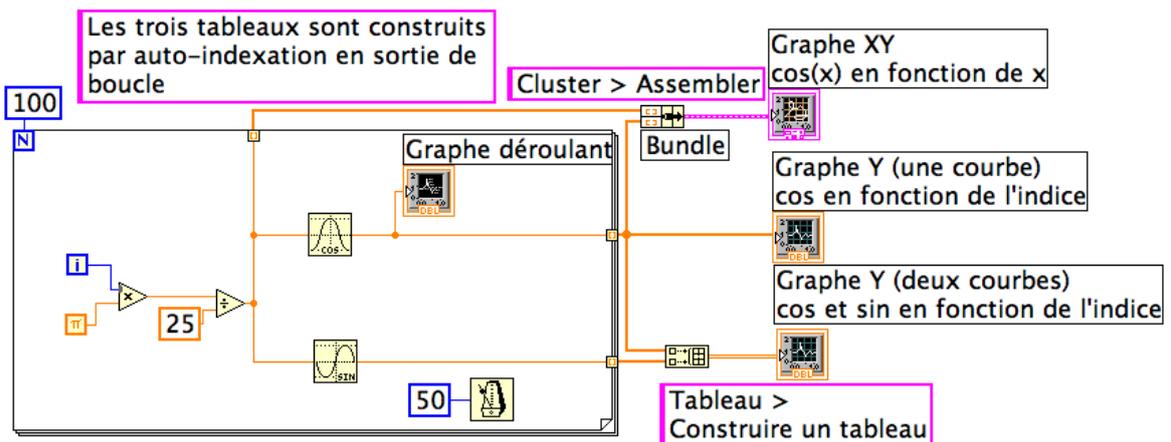
- le **graphe déroulant** (“waveform chart”) trace les valeurs d’un élément au fur et à mesure, c’est-à-dire à chaque itération ou bien à chaque exécution du VI principal ;
- le **graphe** (“waveform graph”), nommé **graphe Y** par la suite, trace les valeurs d’un vecteur en fonction de son indice. Si le graphe Y est branché sur un tableau (un ensemble de vecteurs), les différents vecteurs du tableau sont représentés indépendamment ;
- le **graphe X-Y** permet de tracer un vecteur en fonction d’un autre vecteur de même dimension. Pour l’utiliser, on assemble (“bundle”) un cluster contenant les deux vecteurs des deux axes X et Y (dans cet ordre). Le cluster est alors envoyé au “graphe X-Y”.

Les options de représentation des courbes (couleur, style de tracé, grille...) et axes (format des labels, échelle linéaire ou logarithmique, ...) sont accessibles par différents menus locaux. L’échelle des axes peut être modifiée en changeant une des valeurs avec l’outil A. Un des menus locaux permet de créer un curseur pour visualiser les valeurs du graphe avec la souris.

Une façon alternative de spécifier l’axe horizontal quand celui-ci est régulièrement espacé est de bâtir un cluster contenant les informations : valeur initiale sur x , incrément sur x et vecteur y (dans cet ordre) et de fournir ce cluster à un graphe comme dans l’exemple suivant (figure 30). Cette méthode est très pratique pour éviter de devoir créer des vecteurs inutiles.

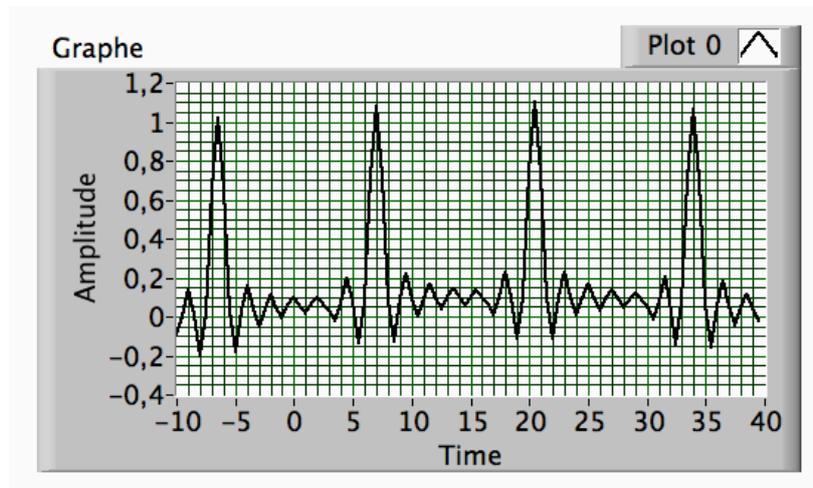


(a) Face avant.

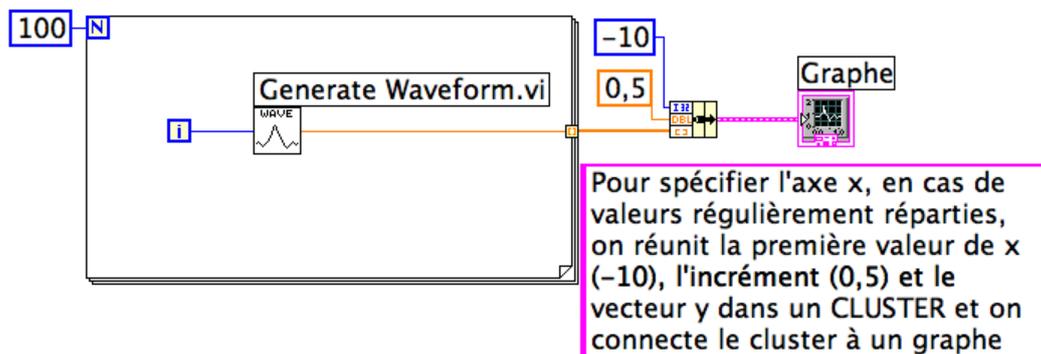


(b) Diagramme.

Figure 29 – Exemple de réalisation de graphes Y, XY et déroulant.



(a) Face avant.



(b) Diagramme.

Figure 30 – Exemple de réalisation de graphes $y = y(x)$ par cluster.

4.12 Exercices

1. Réaliser les exemples de la figure 29 et 30 et analyser leur fonctionnement
2. Représenter la fonction $y = 1/(1 - x^2)$ pour x variant de -20 à 20 par pas de $0,1$ de deux méthodes différentes :
 - en calculant les deux vecteurs y et x et en utilisant un graphe XY
 - en calculant y uniquement et en utilisant un graphe

4.13 Transformée de Fourier

La transformée de Fourier se trouve dans le menu Traitement du signal > Transformée de la palette de Fonctions. N'oubliez pas que la transformée de Fourier d'un signal donne des coefficients complexes. Pour obtenir le module et la phase de ces coefficients, vous devez utiliser la fonction qui extrait ces valeurs d'un nombre complexe, située dans le menu Numérique de la palette Fonctions.

Il existe également plusieurs fonctions dans le menu Traitement du signal > Analyse Spectrale de la palette de Fonctions qui permettent l'analyse spectrale de signaux (FFT de signaux réels, FFT de signaux complexes, FFT power spectrum, ...).

4.14 Exercices

Soit la fonction :

$$f(t) = f_1(t) + f_2(t), \text{ avec } f_1(t) = \sin\left(2\pi\nu_1 t + \frac{\pi}{2}\right) \text{ et } f_2(t) = \sin(2\pi\nu_2 t)$$

1. Calculer, en utilisant une boucle FOR, le tableau des valeurs de t , $f_1(t)$, $f_2(t)$, et $f(t)$ pour t variant de 0 à 100 secondes par pas de 0,01 seconde. On impose les contraintes suivantes :
 - une commande pour chaque fréquence ν_1 et ν_2
 - une commande permettant de fixer le pas de temps, dt
 - une commande fixant le temps d'acquisition T_{acq}
 - un indicateur donnant la valeur de la fréquence d'échantillonnage f_e
2. Tracer les graphes suivants :
 - graphes superposés de f , f_1 , et f_2 en fonction de l'indice
 - graphe de $f(t)$, **en utilisant le tableau des temps t**
 - (*Facultatif*) f_1 en fonction de f_2 (figure dite de Lissajoux, utilisée pour déterminer le rapport de fréquence)
3. Réaliser l'analyse spectrale de f : spectre de phase, d'amplitude et de puissance en fonction de la fréquence
4. Faire apparaître, en cliquant sur le menu local du graphe, la palette graphe et la palette curseurs. Utiliser ces outils pour caractériser les structures : domaine affiché (dépend de quoi? — le vérifier); symétries (par rapport à quoi? valeur?); position des raies (dues à quoi?)
5. Que se passe-t-il si f_e diminue (à T_{acq} constant = 100)? Comparer chaque fois fréquence signal réel et positions en fréquence des raies. Conclusion?

4.15 Autocorrélation et mesure d'un retard

L'autocorrélation d'un signal continu est donnée par :

$$C_{xx}(\tau) = \frac{1}{T} \int_0^T x(t)x(t - \tau)dt$$

Cette fonction a des propriétés que nous rappellerons sans les démontrer :

- fonction paire
- fonction maximale pour un décalage nul ($t = 0$)
- $\lim_{t \rightarrow \infty} C_{xx} = 0$
- si le signal x est à moyenne nulle ($\langle x \rangle = 0$), alors $\langle C_{xx} \rangle = 0$

Un exemple typique est donné par la figure 31.

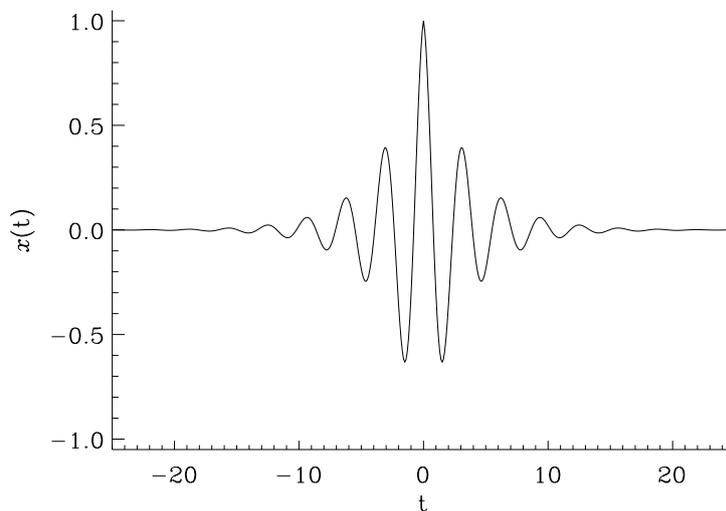


Figure 31 –

Envisageons maintenant le cas d'un signal $x(t)$ et une version de ce signal retardé de t_0 et multipliée par une constante k . On nomme ce nouveau signal $y(t)$: $y(t) = k \times x(t + t_0)$

On peut former une nouvelle fonction, la fonction d'intercorrrelation entre x et y qui s'écrit :

$$C_{xy}(\tau) = \frac{1}{T} \int_0^T x(t)y(t - \tau)dt = \frac{k}{T} \int_0^T x(t)x(t - \tau + t_0)dt = kC_{xx}(\tau'), \quad \text{avec } \tau' = \tau - t_0$$

L'autocorrélation C_{xx} est maximale (et donc l'intercorrrelation C_{xy} l'est également) pour $\tau' = 0$, c'est-à-dire lorsque $\tau = t_0$. La représentation graphique de C_{xy} est similaire à celle de C_{xx} , mais décalée le long de l'axe horizontal. Cette propriété peut nous fournir une méthode pour mesurer le décalage de deux signaux similaires et qui présentent un retard :

1. on calcule la fonction d'intercorrrelation C_{xy} de ces deux signaux
2. on cherche pour quel décalage cette fonction C_{xy} est maximale
3. le retard cherché est égal à la valeur de ce décalage

4.16 Exercices

Soit la fonction :

$$x(t) = e^{-t^2} \times \cos(2t)$$

- Représenter $x(t)$ en échantillonnant cette fonction sur 5000 points entre $t = -10$ et $t = +10$
- Représenter $y(t) = x(t + 1.5)/2.0$
- Calculer l'intercorrélation de ces deux signaux (après soustraction de leur moyenne)
- Retrouver la valeur du décalage (1.5 s) en utilisant cette intercorrélation

L'intercorrélation se trouve dans Fonctions > Traitement du signal > Opérations sur signaux. Bien lire l'aide en ligne de cette fonction.