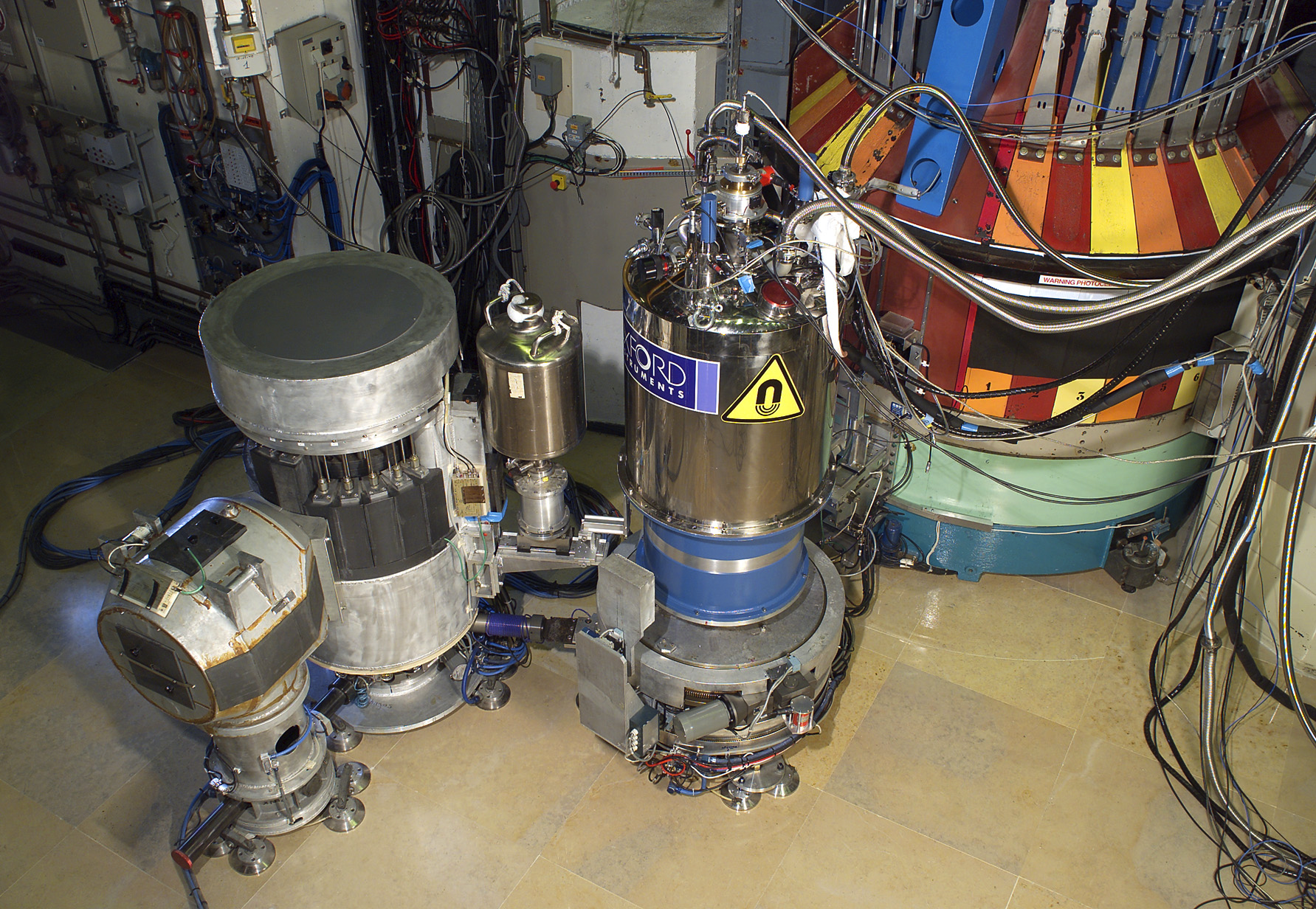


**SIMULATION D’UN SPECTROMETRE**

**TROIS-AXES**

**vTAS**

(vIRTUAL THREE AXIS SPECTROMETER)

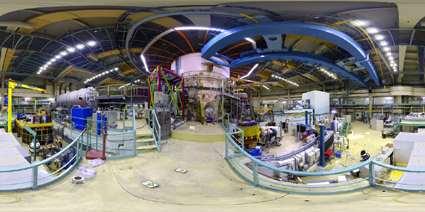


Noëlle Le Delliou – Institut Laue-Langevin - 2006

Sous le tutorat de :

Alain Filhol

Martin Boehm



L’applet **vTAS** (virtual Three Axis Spectrometer) a pour but de modéliser un spectromètre trois-axes a neutrons.

Le logiciel présenté ici s’inspire d’une première version datant de 1998 et développée par Alain Bouvet, un physicien à l’ILL à l’époque. L’interface notamment présentait de nombreux problèmes et a donc dû être entièrement revue. Le nouveau **vTAS** est opérationnel mais n’est pas complet. Diverses améliorations et extensions sont envisageables.

Ce travail a été effectué lors d’un stage de 3 mois sous la tutelle d’Alain Filhol (Divison scientifique – Groupe calcul scientifique) et avec l’aide de Martin Boehm (Physicien co-responsable de IN14, spectromètre trois-axes de l’ILL).

Ce rapport présente ce logiciel ainsi que les méthodes correspondante et donne les explications nécessaires à l’implémentation de celui-ci.

Je tiens à remercier chaleureusement Alain Filhol, mon maitre de stage pour son encadrement, Martin Boehm pour son aide précieuse et l’ensemble du personnel de l’ILL pour son accueil.

###### PLAN

1. Explications physiques
   1. La diffusion inélastique des neutrons
   2. Le cristal échantillon : réseau direct et réseau réciproque
   3. Les indices de Miller
   4. Le spectromètre trois-axes
2. Présentation de l’ancien logiciel
3. Présentation de **vTAS**
   1. Cahier des charges
   2. Présentation générale de la fenêtre
   3. Description des onglets
   4. Les paramètres de l’instrument
   5. Les paramètres de l’échantillon
4. Quelques éléments de réalisation
   1. Liens graphiques – calculs
   2. Graphique
      * 1. Dessin de la maille
        2. Représentation graphique des coordonnées du triangle
   3. Calculs
      * 1. La matrice UB
        2. Calcul des paramètres de maille réciproque
        3. Calcul des angles
        4. Calcul de ki et kf en pixels
        5. Calcul des coordonnées à partir des longueurs
        6. Le scan automatique
   4. Programmation
      * 1. Déroulement général du programme
        2. Gestion des évènements
   5. Les murs
      * 1. Dessin des murs
        2. Détermination des murs
   6. Imprécisions
5. Intégration de l’applet dans une page HTML
6. Ce qu’il reste à faire
7. Annexes
   1. Où trouver des informations
   2. Plans des instruments

# I- Explications physiques

1. La diffusion inélastique des neutrons

Pour bien comprendre le comportement des matériaux, on doit, non seulement connaître leur composition chimique, leur structure cristallographique… mais aussi la façon dont les molécules bougent, vibrent… à l’intérieur de l’échantillon.

Pour cela, on fait appel à la diffusion inélastique des neutrons qui apparaît lorsqu’une onde interagit avec un objet en mouvement. Cette technique est complémentaire avec les méthodes aux rayons X, utilisées à l’ESRF par exemple.

On observe la diffusion inélastique lorsqu’un neutron cède ou reçoit de l’énergie de l’atome qu’il percute. Lors d’une collision avec perte d’énergie pour le neutron, celui-ci a une vitesse plus basse et donc une longueur d’onde plus grande.

Atome

Flux de neutrons incident

Flux de neutrons réfléchi

Il existe deux modes de diffusion inélastique : la diffusion incohérente qui nous renseigne sur les mouvements individuels des atomes et la diffusion cohérente pour les mouvements collectifs.

1. Le cristal échantillon : réseau direct et réseau réciproque

Dans un cristal, les atomes sont donnés de façon tri périodique. Autrement dit, un cristal est constitué d’empilement dans les trois directions de l’espace d’une brique élémentaire, ou maille, contenant un certain nombre d’atomes. Le réseau tri périodique est appelé réseau direct ou cristallin.

La maille (ou brique) élémentaire est le motif géométrique d’atomes le plus simple, qui, en se répétant indéfiniment, constitue un réseau cristallin. Une maille élémentaire contient un noeud du réseau à chaque sommet, mais aucun noeud à l’intérieur de son volume ou de l’une de ces faces.

Le réseau réciproque est un réseau construit géométriquement à partir du réseau direct du cristal. On déduit ainsi des paramètres de maille du réseau direct de nouveaux paramètres de maille.

Chaque point du réseau réciproque est associé à une direction dans le réseau direct et inversement. C'est donc un outil très pratique pour le physicien car il permet de visualiser le déplacement du point de mesure. Autrement dit, chaque couple [configuration du spectromètre - orientation de l'échantillon] est associé à un point dans l'espace réciproque et chaque type de balayage est associé à un déplacement de ce point selon une trajectoire donnée dans l'espace réciproque.

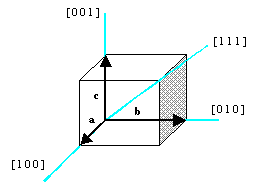
On peut distinguer 7 familles de maille, détenant chacune des propriétés particulières sur les angles et la longueur de la maille élémentaire. Le tableau ci-dessous est un récapitulatif des caractéristiques de maille :

|  |  |
| --- | --- |
| **Nom famille** | **Propriétés** |
| Cubique |  |
| Hexagonal | systeme-hexagonal |
| Tétragonal ou quadratique |  |
| Orthorombique | systeme-orthorhombique |
| Monoclinique | systeme-monoclinique |
| Triclinique | systeme-triclinique |
| Rhomboeidique | systeme-rhomboedrique |

1. Les indices de Miller

Un fois que l’on a déterminé la maille élémentaire du cristal, on peut en définir une base normée où les trois axes sont les vecteurs a, b et c. Les indices de Miller sont les coordonnées d’un vecteur dans cette base et sont notés (h,k,l).

Voici un exemple de cette notion pour une maille cubique:



Dans cet exemple, les trois angles sont égaux à 90º et les trois vecteurs sont égaux, le repère est donc orthonormé, ce qui n’est pas toujours le cas, selon le type de maille.

On peut remarquer qu’en utilisant cette notion on peut déterminer un vecteur du cristal avec ces indices de Miller, donc en (r.l.u) ou avec sa longueur en Å-1 et son orientation. Les indices de Miller sont donc très pratiques pour se repérer dans le référentiel cristallographique.

1. Le spectromètre trois-axes

Il existe plusieurs types de spectromètres de diffusion inélastique des neutrons. Chaque instrument est optimisé pour une utilisation particulière, selon ce que l'on cherche à observer.

Le spectromètre neutron à trois axes, appelé aussi TAS (Triple Axis Spectrometer) a été inventé en 1955 par B. Brockhouse, et ce travail lui a valu le prix Nobel de physique en 1994. C'est en effet l'instrument emblématique de l'étude des dynamiques atomiques, moléculaire et de spin dans les cristaux, dynamiques qui sont au coeur d'un grand nombre de leurs propriétés.

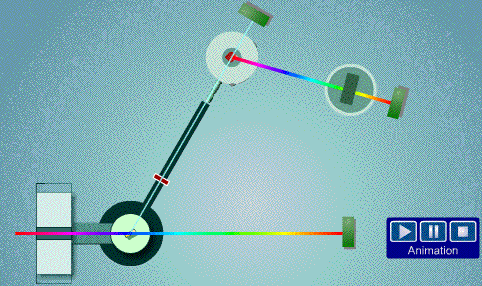
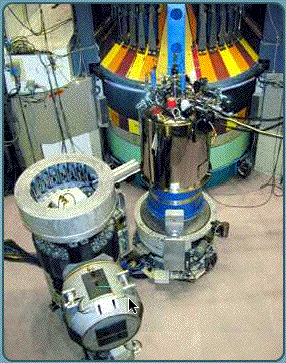
Le spectromètre à trois axes effectue la mesure quantitative de la distribution en énergie des neutrons diffusés par l'échantillon dans la direction d'observation choisie et pour l'énergie (la longueur d'onde) de neutrons incidents choisie. Il est donc potentiellement capable de tout mesurer dans le domaine d'énergie et de distances accessibles aux neutrons.

En pratique, l'instrument se compose :

- d'un cristal monochromateur qui sélectionne une longueur d'onde à partir du faisceau polychromatique de neutrons provenant du réacteur. L'axe de rotation de ce monochromateur (choix de l'énergie de mesure) est le 1er axe de l'instrument.

- le second axe est celui de la rotation du porte-échantillon qui permet d'orienter le réseau cristallin dans la direction d'observation voulue. C'est aussi l'axe de rotation pour le positionnement du cristal analyseur dans une direction de diffusion des neutrons donnée.

- le troisième axe est celui de la rotation du cristal analyseur qui analyse l'énergie des neutrons pour la direction de diffusion choisie. C'est aussi l'axe de rotation du détecteur de neutrons qui mesure ainsi le spectre d'énergie des neutrons pour chacune des directions de diffusion et d'orientation de l'échantillon choisies.



Analyseur

Détecteur

Echantillon

Monochromateur

1. Le scan

Effectuer un scan revient à fixer une des variables et à effectuer un balayage avec le spectromètre entre deux valeurs limites.

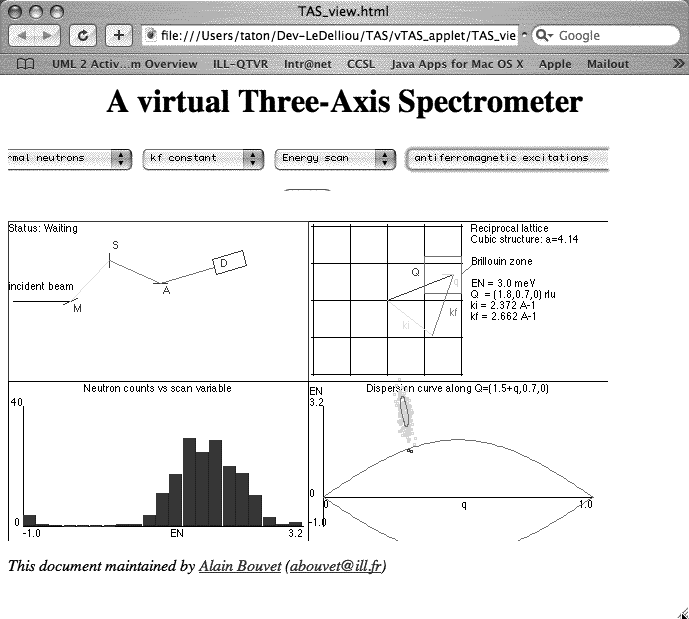
On obtient ainsi un ensemble de spectres de diffusion inélastique des neutrons dont les pics sont l'information cherchée. Leur observation pour un nombre suffisant de points de l'espace réciproque permet ensuite au physicien de modéliser la dynamique du matériau.

Au delà de la modélisation graphique du spectromètre, c’est ce qui va nous intéresser dans ce projet.

# II - Présentation de l’ancien logiciel

Alain Bouvet, un physicien de l'ILL, est l'auteur en 1998 d'une première version de Virtual-TAS, implémentée en JAVA 1.0.

Ce logiciel, très prometteur, avait malheureusement été développé trop vite et a donc été suspendu pendant de longues années.



Présentation de la fenêtre :

La barre de menu située dans la partie supérieure de la fenêtre permet à l’utilisateur de choisir la température des neutrons incidents (c’est-à-dire le type de source sur lequel l’instrument est installé), le type de balayage que l’on va exécuter et le type d’excitation. Sous ce menu se trouve le bouton “scan” qui permet de lancer le balayage.

La fenêtre est composée de quatre panneaux :

Panneaux du haut

- à gauche : la configuration de l'instrument pour le point de mesure considéré

- à droite : l'espace de mesure, c'est-à-dire, le point et l'énergie de la mesure et la direction de son déplacement dans le réseau réciproque du cristal échantillon.

Panneaux du bas

- à droite : plusieurs configurations instrumentales permettent d'atteindre le même point de mesure dans l'espace réciproque, mais l'orientation de la sonde instrumentale est différente. La résolution instrumentale n'étant pas une petite sphère, mais un ellipsoïde allongé, le choix de son orientation lors d'un balayage est crucial pour l'obtention de bons résultats. Le logiciel figure donc la forme de la sonde instrumentale et la façon dont elle va traverser la courbe de dispersion d'un phonon acoustique (vibration particulière de réseau cristallin).

- à gauche : la forme du pic de diffusion que l'on peut s'attendre à observer selon que l'on a bien ou mal choisi les paramètres de balayage.

L’image ci-dessus est une copie d’écran de la fenêtre à un moment donné mais en réalité, les dessins des quatre panneaux sont sans cesse animés.

# III – Présentation du vTAS

# Cahier des charges

Les utilisateurs visés :

* Débutants dans le milieu du trois-axes mais ayant des notions de cristallographie (paramètres de maille, réseau réciproque, etc) et sur le principe de base d’un spectromètre. Ces utilisateurs veulent simplement jouer avec le logiciel;
* Physiciens experts dans le domaine. Ils veulent rapidement vérifier certaines configurations de leur expérience, sans avoir à utiliser le logiciel Mad (logiciel qui contrôle les troics-axes de l’ILL). En effet, comme le nombre de paramètres instrumentaux est élevé, il n'est pas simple de passer mentalement de la configuration instrumentale à l'espace de mesure (espace réciproque), même pour un physicien entraîné.

Caractéristiques du logiciel :

* Interactivité : l’utilisateur doit avoir la main mise sur tous les paramètres d’expérience. Entre autre, il doit pouvoir saisir des valeurs numériques au clavier et toutes les connexions entre ordinateur et utilisateur doivent être utlisées (clavier, souris);
* Facile à utiliser : les actions de l’utilisateur doit être guidées et rapidement, il doit savoir utiliser le logiciel. Le déroulement doit donc être logique

Le logiciel a été développé sous Mac OSX en utilisant la plateforme Netbeans 4.1. Il était également testé en parallèle sous PC avec Windows XP et Netbeans 5.0, associé à JAVA 1.4.2.1.

# Recommandations d’usage :

Afin de garantir une utilisation optimale, il est recommandé de :

- Avoir la JVM installée sur la machine;

- Avoir la dernière version de JAVA (J2SE 5.0);

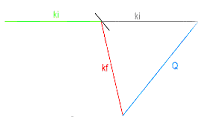
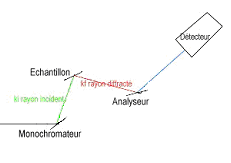
- Avoir Firefox ou Safari pour ouvrir le projet sous frome HTML (ne marche pas avec Internet Explorer);

- Avoir un Mac pour ouvrir ce projet sous forme d’applet (.app).

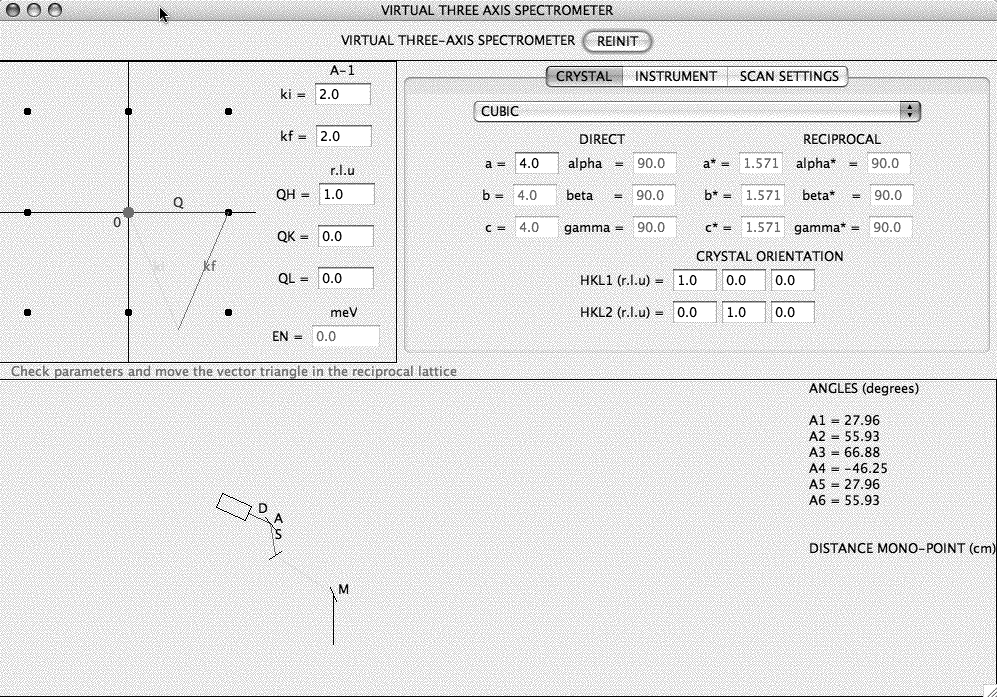
1. Présentation générale de la fenêtre

Remarque préalable : représentation graphique du spectromètre

Dans l’espace direct : Dans l’espace réciproque :



Les angles et les longueurs étant corrélés, à chaque fois que l’on modifie la longueur de ki ou kf (par la souris ou directement dans le champ texte), on doit recalculer les angles correspondants afin de redessiner la configuration correcte de l’instrument dans l’espace réciproque.



Nous avons choisi de créer une fenêtre de taille (950,630) pour que celle-ci prenne toute la place du plus petit écran.

L’écran est divisé en trois parties :

* En haut à gauche se trouve dessin du triangle représentant le trois-axes en configuration de mesure, dans le réseau réciproque. Ki est le vecteur incident, kf le vecteur diffracté et Q le vecteur de diffusion. Leurs valeurs sont données à droite de ce panneau dans les unités correspondantes. QH, QK et QL sont les composantes en r.l.u du vecteur Q. On trouve également la valeur de l’énergie qui est la différence entre les deux vecterus ki et kf et est donnée en meV. Les champs textes contenant tous ces paramètres sont grisés ou non, selon les choix effectués par le biais des onglets;
* En haut à droite se trouvent les onglets permettant de régler les paramètres de l’instrument et des conditiosn de l’expérience;
* En bas se trouvent le dessin trois-axes en configuration instrumentale, dans le réseau direct et les mesures de angles de l’instrument. A droite de ce panel se trouvent les angles de l’instrument, ceux-ci étant décrits plus précisément par la suite.

La classe dessinRec (resp. dessinTas) contient les informations respectives de l’instrument dans le réseau réciproque (resp.direct) et dérive d’un *Japplet*.

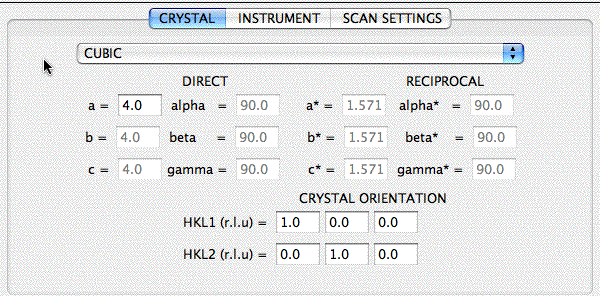
L’interface a été codée entièrement à la main, sans utiliser l’éditeur d’interface de Netbeans.

# Description des onglets

# 

L’utilisateur accède rapidement au réglage et au choix d’un certain nombre de paramètre grâce aux 4 onglets situés en haut à droite. Nous allons détailler ci-après chacun de ces onglets.

* + - * l’onglet CRYSTAL



Il permet à l’utilisateur de choisir les paramètres du cristal échantillon, c’est-à-dire les paramètres de maille et l’orientation de celui-ci. Avec le menu déroulant (JcomboBox), on commence donc par sélectionner un type de maille parmi les 7 proposées (cubique, hexagonale, trigonale, orthorhombique, tétragonale, triclinique et monoclinique). Ensuite, on rentre les longueurs (a, b et c) et les angles (alpha, beta, gamma) définissant la brique élémentaire dans l’espace direct dans les champs de texte correspondants. Pour information, les paramètres cités précédemment sont calculés dans l’espace réciproque et indiqués à côté. Pour le calcul de ces paramètres, voir la partie « calcul de la maille réciproque ».

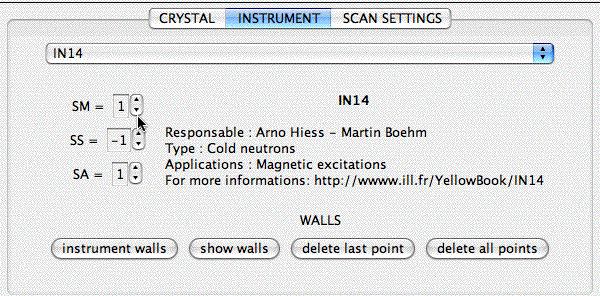
On peut constater que, selon la famille de maille sélectionnée, certains champs de texte sont rentrés automatiquement et deviennent non modifiables et que l’utilisateur est guidé dans le choix de ces paramètres, selon le type de maille.

Par exemple, si on choisit un échantillon de maille cubique, les trois angles sont bloqués à 90 degrés et, comme dans ce cas, a = b = c, les champs correspondant à b et c sont bloqués. Il suffit alors uniquement de rentrer la valeur de a, de taper sur “ ENTRÉE” et a et b prennent la même valeur que celle choisie pour a.

Une fois que la maille du cristal est réglée, on choisit les vecteurs HKL1 et HKL2 définissant l’orientation du cristal.

La modification du type de cristal et des paramètres de celui-ci entraîne la modification du dessin des points de la maille (points noirs sur le graphique de la partie gauche). Par exemple, si on choisit une maille hexagonale, ces points formeront un motif hexagonal dans la représentation réciproque. Si on change les longueurs de la maille avec les paramètres a, b, c, les points seront plus ou moins rapprochés.

* + - * l’onglet INSTRUMENT



Il permet de choisir l’instrument qui va servir à faire l’expérience.

Tout d’abord, on sélectionne un spectromètre dans une liste. Ce sont des trois-axes présents à l’ILL (IN14, IN8, IN1, IN20).

Une fois l’instrument choisi, on peut encore changer le sens de diffusion de chaque composant du trois-axes (SM pour le monochromateur, SS pour l’échantillon et SA pour l’analyseur). On change la valeur de ces variables (-1 ou 1) en appuyant sur la flèche montante ou descendante correspondant à chacune de ces variables.

On peut ensuite dessiner des murs autour du trois-axes dans le réseau direct. Ceci peut se révéler utile lors d’une rapide vérification de ceratines configurations possibles par exemple.

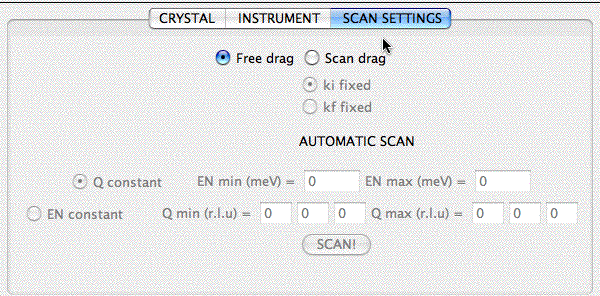
Il y a alors deux possiblités pour les faire apparaître :

On peut les dessiner à la main avec la souris, en cliquant sur les points correspondant aux coins des murs. Dans ce cas, pour les dessiner, on appuie sur le bouton « show walls » ce qui aura pour effet de relier ces points entre eux.

On peut également faire dessiner les murs prédéfinis dépendant de l’instrument choisi en appuyant sur le bouton « instrument walls ». les murs dessinés sont aussi proches que possible de la réalité et ont été déterminés à partir des plans des instruments de l’ILL. La détermination et le dessin des murs sont expliqués plus précisément par la suite.

Si l’utilisateur veut modifier le dessin, il peut soit effacer tous les murs (dessinés à la main ou automatiquement) en cliquant sur le bouton « delete all points ». Il peut également effacer le dernier mur affiché en cliquant sur « delete last point ».

* + - * l’onglet SCAN SETTINGS



Il existe quatre façons de bouger le triangle :

* Le mode de positionnement manuel avec la souris;
* Le mode de positionnement à partir de valeurs rentrées au clavier;
* Le mode de scan manuel avec la souris;
* Le mode de scan automatique avec des valeurs limites rentrées au clavier.

La représentation du spectromètre dans la configuration instrumentale ne peut pas être bougée. Celle-ci suit les mouvements effectués sur le spectromètre dans la configuration de mesure (dans le réseau réciproque). Le point du triangle noté 0 restera fixe, quelque soit la configuration du triangle.Son emplacement dépend uniquement des dimensions de la fenêtre (il est situé au milieu du panneau).

Le mode de positionnement manuel :

En général, on utilise ce mode en début d’expérience pour positionner le triangle en cochant le bouton radio “free drag”.

L’utilisateur fixe manuellement et librement la position des deux extrémités libres du triangle. Alors, on clique sur l’une de celles-ci et on amène le point en restant appuyé avec la souris. Les trois vecteurs Q, ki et kf bougent en conséquence, sans contraintes.

Les valeurs des vecteurs dans leurs unités respectives (Å-1 ou r.l.u) sont indiquées à droite du triangle.

## Le mode de positionnement à partir de valeurs numériques :

Ici, l’utilisateur va positionner le triangle, non plus avec la souris mais grâce à des valeurs numériques qu’il rentre dans les champs textes prévus à la droite du triangle.

Par exemple, s’il veut commencer son expérience avec une valeur de ki précise, il rentrera celle-ci dans le champ texte prévu pour ki. Alors, après validation (touche “ENTRÉE”), le triangle prendra la configuration voulue (Q ne change pas et kf est calculé en fonction).

Remarque : l’utilisateur dit valider chaque valeur rentrée. Il ne peut pas, par exemple, rentrer ki, puis kf, puis valider en tapant sur “ENTRÉE”. Dans ce cas, la valeur de ki ne serait pas prise en compte.

## Le mode de scan manuel

Ce mode est utilisé, en général après l’un des deux (ou les deux) modes décrits ci-dessus et est activé lorsque le bouton radio “scan drag” est sélectionné.

Une fois que le triangle est correctement positionné, on peut commencer l’expérience et effectuer le balayage. Le balayage manuel se fait avec la souris. Comme pour le mode de positionnement manuel, on bouge l’une des deux extrémités libres du triangle.

Cependant, on impose en plus quelques contraintes dûes au choix du scan.

* Le scan en Q avec ki constant

Ce mode de scan est activé lorsque les boutons radio “Q constant “ et “ki fixed” sont sélectionnés.

Dans ce cas, le vecteur Q reste fixe (en orientation, sens et module) et la valeur du vecteur ki reste fixe, mais son orientation varie. Alors, on clique sur l’extrémité commune entre ki et kf et on amène la souris sur le point désiré, avec les contraintes énoncées précédemment. Le pointeur de la souris doit rester sur le cercle de rayon ki et de centre 0, donc en fait, celui-ci détermine uniquement l’orientation du vecteur ki.

La valeur de kf s’affiche à droite du panneau.

Calcul de kf :

L’extrémité du vecteur devant rester sur le cercle de centre 0 et de rayon ki, on détermine le rayon de ce cercle en pixel. Ensuite, on replace le pointeur de la souris sur ce cercle, par une simple règle de trois.

* Le scan en Q avec kf constant

Ce mode de scan est activé lorsque les boutons radio “Q constant “ et “kf fixed” sont sélectionnés.

Il fonctionne de la même façon que le mode prédédent mais comme ici, c’est la valeur de kf qui est fixé, le curseur de la souris reste sur le cercle de rayon kf et de centre le point commun entre Q et kf.

De même, la valeur de ki est affichée à droite du panneau.

Remarque: ici, le fait que kf reste constant est moins visible à cause du changement d’échelle. Celui-ci étant calculé en fonction de ki et puisque celui-ci varie, le facteur d’échelle est modifié et on peut donc moins bien observer la contrainte.

Calcul de ki :

Le calcul de ki utlise exactement la même méthode que pour kf mais en changeant la valeur du rayon, puisque maintanant, le cercle est de rayon kf et de centre (xQ, yQ).

* Le scan en énergie

Ce mode de scan est activé lorsque le bouton radio “EN constant “ est sélectionné.

Dans ce cas, les valeurs des vecteurs ki et kf restent constantes, mais leurs orientations varient. On bouge avec la souris le point commun entre Q et kf. La valeur et l’orientation de Q s’adapte à celles de ki et kf pour fermer le triangle.

Les composantes de ce vecteur sont affichées à droite du panneau.

Calcul de QH et QK et QL :

Les valeurs des composantes de Q en Angstroms inverses sont uniquement la différence entre les coordonnées de l’origine et le point cliqué. Ensuite, pour avoir leurs valeurs en r.l.u, on multilplie ce vecteur par *Ubinv* (voir la partie Matrice UB pour plus de précisions).

Le mode de scan automatique

Le troisième mode de manipulation correspond à un balayage automatique dans lequel l’utilisateur, en plus de choisir le vecteur fixé et mode de balayage (comme dans le balayage manuel), rentre les valeurs limites de la plage de balayage. Pour valider ces valeurs, il faut appuyer sur « ENTREE ».

Pour lancer le balayage, l’utilisateur active le bouton « SCAN ».

Alors, les dessins de l’instruments dans les deux panels bougent en même temps pour balayer la zone indiquée.

L’activation de ce bouton fait en fait appel à la routine *scanAuto* qui sera décrite par la suite.

L’onglet SCAN SETTINGS comporte ainsi l’ensemble de boutons radio, bouton et champs de texte permettant de régler et de choisir le jeu de paramètres.

Selon le mode choisi, certains champs sont inaccessibles.

# Les paramètres de l’instrument

L’utilisateur a le choix entre quatre trois-axes de l’ILL : IN14, IN1, IN8 et IN20.

Chacun de ces instruments a des particularités, dépendant, entre autres, de la mesure pour laquelle il est destiné.

Liste des paramètres spécifiques à chaque spectromètre:

* Longueur entre les composants (en cm) : lg\_MonoSample, lg\_SampleAna, lg\_AnaDet. La variation de ces valeurs selon les instruments n’est visible quand dans la configuration instrumentale située dans el panneau inférieur;
* Distances entre les plans réticulaires (en Å-1) : DM pour le monochromateur, DA pour l’analyseur;
* Sens de diffusion de chaque composant : SM (monochromateur), SS (échantillon), SA (analyseur). Ces variables ont pour valeur 1 ou –1.
* Des valeurs de base de ki, kf (en Å-1) et QH, QK, QL (en r.l.u). Ces variables sont les seuls paramètres instrumentaux qui vont varier au cours de l’expérience, lors du positionnement ou du scan. Ce sont elles qui vont déterminer la configuration initiale du triangle et du spectromètre dans le panneau du bas, lorsque l’utilisateur va changer d’instrument. La différence de ces valeurs entre les instruments provient de la différence de température des neutrons incidents et donc du type de mesure que le trois-axes va effectuer.
* Les coordonnées des murs, sous forme de tableau à deux colonnes

Remarque :

Par défaut, on choisit de représenter le spectromètre IN14.

# Les paramètres de l’échantillon

Le programme précédent ne manipulait que des échantillons cubiques, don’t on ne pouvait pas changer la taille.

Ici, on peut choisir le type de cristal, parmi les 7 familles de maille présentées, ainsi que la taille de la maille élémentaire et l’orientation du cristal. On distingue alors trois types de paramètres pour le cristal :

* Les longueurs des mailles (en Å) : AS, BS, CS dans le programme et a,b,c dans l’interface;
* Les angles entre les trois vecteurs constituant le référentiel cristallographique (en degrés) : AA, BB, CC dans le programme et alpha, beta, gamma dans l’interface;
* L’orientation du cristal est défini par deux vecteurs (donnés en r.l.u.) déterminant le plan d’observation : AX, AY, AZ dans le programme, HKL1 dans l’interface définit le premier vecteur et BX, BY, BZ (HKL2) définit le deuxième.

Note : dans la routine *matrixUB,* HKL1 est appelé vect\_A et HKL2 vect\_B.

Ce sont des tableaux de trois cases contenant les composantes de ces vecteur.

Remarques :

Par défaut, le cristal échantillon est choisi cubique de longueur 4 Å et le plan est défini par les vecteurs (1,0,0) et (0,1,0).

Les paramètres définissant la maille élémentaire sont soumis aux contraintes imposées par le type de maille.

# IV- Quelques éléments de réalisation

1. Liens graphique-calculs

Dans ce projet, on peut voir se détacher deux parties bien distinctes : la partie graphique et la partie physique.

La partie physique comprend les calculs qu’effectue le physicien, en dehors de l’application (calcul des angles et de la matrice UB par exemple).

Ces calculs sont donc effectués avec les unités telles que l’Å-1 ou les unités r.l.u.

La partie graphique comprend tous les dessins (dans les deux réseaux) et les calculs géométriques (calculs des coordonnées par exemple). Toutes ces manipulations sont effectuées avec l’unité de l’écran : le pixel.

On doit sans cesse naviguer entre ces deux parties.

Par exemple, lorsque l’on rentre une valeur de kf en Å-1, on doit de suite adapter le dessin du triangle, et donc, ses coordonnées en pixels. Ensuite, lorsqu’on bouge le triangle à la souris, on change les valeurs des coordonnées en pixels et on doit immédiatement calculer la valeur des vecteurs, puis les nouveaux angles (le calcul des angles dépend des valeurs des vecteurs en unités physiques).

Ces deux parties étaient extrèment liées dans le code initial et j’ai donc créé un coefficient permettant de lier les deux parties.

Ainsi, *coeffpixel,* permet de passer des unités physiquesaux pixels.

Il dépend de la taille de la fenêtre et de la longueur de certains vecteurs (pour plus de détails sur le calcul, voir le code dans la routine *calcMaille*).

Ainsi, il est modifié à chaque fois qu’on diminue ou agrandit la fenêtre ou qu’on la longueur de ces vecteurs (à chaque appel de *monDessinRec.repaint()*).

Il suffira alors de multiplier ou de diviser la valeur d’un vecteur par ce coefficient pour retrouver la longueur dans l’unité voulue.

1. Graphique

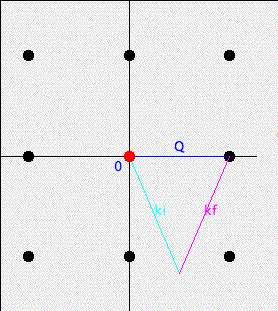
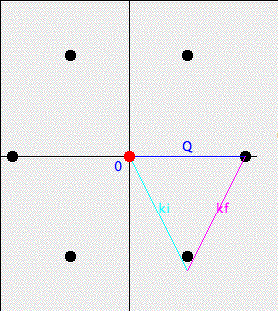
# Le dessin de la maille

Contrairement à la version précédente, nous avons décidé d’avoir la possibilité de choisir le type de maille, de représenter différemment chaque famille de maille et d’adapter le dessin en fonction des paramètres.

Ainsi, à chaque nouveau paramètre de maille choisi par l’utilisateur, nous devons déterminer une nouvelle fois les points la constituant que nous allons dessiner dans le panneau réservé au réseau réciproque.

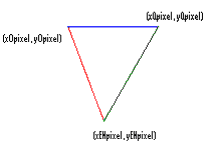
On détermine d’abord *coeffpixel* en fonction de la largeur du panneau et du maximum de quatre vecteurs (pour plus de précisions, voir le code donné en annexe dans la routine *calcMaille*), puis le nombre de cellules à dessiner, et enfin, on construit un tableau contenant les coordonnées en pixel de ces points. On peut noter que le dessin de ces points dépend des vecteurs HKL1 et HKL2, de la matrice UB et donc des paramètres de maille, ainsi selon le type de maille choisi, la forme du dessin sera différente :

Maille cubique : Maille hexagonale :

* 1. Représentation graphique des coordonnees

Le schéma ci-dessous représent le triangle avec les noms de variables donnés aux coordonnées des sommets de triangles. A noter également que le point d’origine (de coordonnées x0pixel et y0pixel reste fise et est déterminé en fonction de la taille du panneau, donc de la fenêtre).



1. Calculs

# La matrice UB

La matrice UB permet de passer du référentiel cristallographique réseau de l’échantillon (pas forcément orthormé) à un référentiel orthonormé tel que n’importe quel couple de vecteurs soit dans le plan de diffusion.

A partir des trois axes du référentiel cristallographique, la matrice B crée un référentiel orthornormé. Ensuite, si on prend deux vecteurs, la matrice U se charge d’effectuer trois rotations successives afin de repositionner ces deux vecteurs dans le plan de diffusion.

Soit (h,k,l) les coordonnées d’un vecteur dans le référentiel cristallographique, en (r.l.u) et (X,Y,Z) les coordonnées de ce même vecteur dans le référentiel orthonormé de l’instrument. Alors, on a : (X,Y,Z) = UB(h,k,l)

Dans le logiciel, ces deux vecteurs qui vont défnir l’orientation sont HKL1 et HKL2 et la matrice est calculée dans la routine *matrixUB.* L’essentiel de cette méthode avait été codé par Alain Bouvet (en fait traduit en JAVA à partir de Fortran) et j’ai donc, après avoir vérifié les calculs avec M. Boehm, inséré cette fonction dans mon code.

En pratique, la matrice UB permet de passer des Angstroms inverses aux coordonnées r.l.u.

Pour passer des r.l.u aux Angstroms inverses, on utilisera la matrice inverse, calculée dans la routine *matrixUBinv*. Le code de cette routine est inspiré d’une fonction en Fortran du code du logiciel TASMAD.

Remarque : si le cristal est cubique et son orientation selon les vecteurs (1,0,0) et (0,1,0), la matrice UB sera diagonale de même valeur.

# Le calcul des paramètres de maille réciproque

A chaque nouveau paramètre de maille rentré dans les champs texte, on calcule les paramètres de maille correspondants dans le réseau réciproque, que l’on affiche à côté, dans l’onglet “CRYSTAL”.

Comme le lien entre ces deux référentiel est uniquement géométrique, on calcule facilement ces nouveaux paramètres.

On calcule d’abord le volume de la maille directe (en Å3)

**V = abc(1 -cos2()-cos2()-cos2() + 2cos()cos()cos())1/2**

Avec :

a,b,c,****,****,****: maille directe (Å et °)

a\*,b\*,c\*,\*,\*,\* : maille réciproque (Å-1 et °)

Remarque : on a V\*= 1/V où V\* est le volume réciproque (en Å-3)

Ensuite, on exprime les nouveaux paramètres de maille en fonction des anciens et du volume :

**a\* = bc sin()/V et permutations circulaires**

**cos(\*) = (cos()cos()-cos())/sin()sin()**

Il pourrait être envisageable de pouvoir rentrer, au choix, les paramètres de maille directs ou réciproques, en effectuant les calculs inverses.

Remarque :

On peut également calculer ces paramètres de la façon suivante :

a\* = [G\*](0,0), b\* = [G\*](1,1), c\* = [G\*](2,2)

cos(\*) = [G\*](1,2)/bc,

cos(\*) = [G\*](0,2)/ac,

cos(\*) = [G\*](0,1)/ab

où [G\*] = 1/[G] avec

| a2 ab cos() ac cos()|

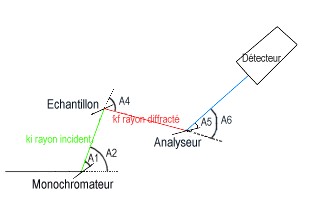
[G] =|ab cos() b2 bc cos()|

|ac cos() bc cos() c2 |

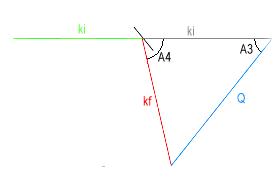
G est appelé tenseur métrique et G\* tenseur métrique réciproque.

# Le calcul des angles

Le spectromètre dans la configuration instrumentale :



Le spectromètre dans la configuration de mesure :



Les angles sont calculés à chaque nouvelle valeur d’un des trois vecteurs dans la routine *calcTASAngles.*

* Calcul de A1 et A2

La loi de Bragg donne la relation suivante, pour le monochromateur :

λ = 2.DM.sin(A1)

avec DM la distance inter-réticualaire du monochromateur et lambda la longueur d’onde sélectionnée.

On a également la relation suivante :

ki = 2.π/ λ

On peut donc donner l’expression de A1 :

Sin(A1) = π/(ki\*SM)

Donc **A1 = asin(π/(ki\*DM))**

On a également l’expression de A2 puisque **A2=2\*A1**

* Calcul de A5 et A6 :

De même, on applique la loi de Bragg sur l’analyseur :

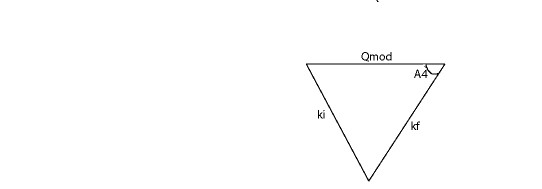
λ = 2.DA.sin(A5)

Et kf = 2.π/ λ

D’où **A5 = asin(π/(kf\*DA))**

Et on a également **A6 = 2\*A5**

* Calcul de A4 :

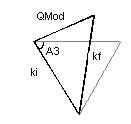


où Qmod est le module du vecteur Q

Le triangle étant quelconque, on peut appliquer la relation suivante (relation trigonométrique dans un triangle quelconque)

**cos(A4) = (ki2+kf2-Qmod2)/(2\*ki\*kf)**

* Calcul de A3 :



On a

**cos(A3) = (ki2+kf2-Qmod2)/(2\*ki\*QMod))-atan(YYC,XXC)**

Avec XXC et Yyc les composantes de Q en Angstroms iverses.

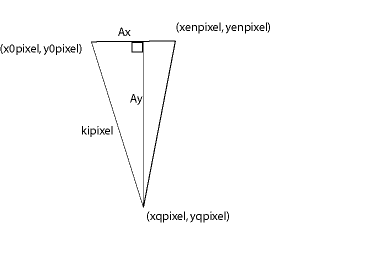
Les angles sont calculés en radians et sont affichés en degrés avec 2 chiffres significatifs à droite du panneau inférieur.

# Le calcul de kipixel et kfpixel

* kipixel :

Pour calculer kipixel, on construit un triangle rectangle afin d’y appliquer le théorème

de Pythagore :

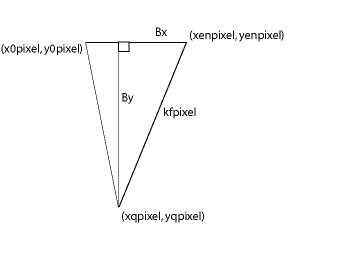


double Ax = xenpixel-x0pixel;

double Ay =yenpixel-y0pixel;

kipixel = (int)Math.sqrt(Ax\*Ax+Ay\*Ay

* kfpixel :



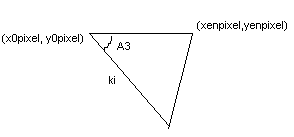
double Bx = xqpixel-xenpixel;

double By = yqpixel - yenpixel;

kfpixel = (int)Math.sqrt(Bx\*Bx+By\*By);

# Le calcul des coordonnées à partir des longueurs

* Calcul des coordonnées de l’extrémité commune entre ki et kf : routine *calcCoordENpixel*



On applique simplement dans le triangle la relation du cosinus :

cos(A3) = adj/hyp= (xenpixel-x0pixel)/ki

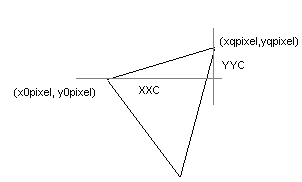
sin(A3) = opp/hyp = (yenpixel-y0pixel)/ki

Comme on veut la coordonnée en pixel, on multiplie par *coeffpixel* et on a

xenpixel= x0pixel + (int)(ki\*Math.cos(A3)\*(double)coeffpixel);

yenpixel= y0pixel + (int)(ki\*Math.sin(A3)\*(double)coeffpixel);

* Calcul des coordonnées de l’extrémité commune entre kf et Q : routine *calcCoordQpixel*



Avec XXC et YYC les composantes du vecteur Q en Å-1 .On a donc XXC\**coeffpixel* et YYC\**coeffpixel* les longueurs en pixels du vecteur Q, soit la distance entre l’origine et ce point,on a :

xqpixel = x0pixel + (int) (XXC\*(double)coeffpixel);

yqpixel = y0pixel - (int) (YYC\*(double)coeffpixel);

* 1. Le scan automatique

Lorsque l’utilisateur choisit le mode de scan automatique, il choisit dans les champs de texte correspondants dans l’onglets « SCAN SETTINGS » les valeurs limites de son scan. Comme il a déjà été détaillé, il peut choisir trois modes de scan : le scan en EN avec ki ou kf fixé et le scan en Q. La routine *scanAuto* calcule les nouvelles valeurs des vecteurs afin de faire bouger simultanément le triangle et le spectromètre.

Pour donner une impression de fluidité, nous avons choisi de calculer 100 positions. Pour changer cette donnée, il suffit de changer la variable *np* (nombre de points).

Voici dans le détail l’algorithme utilisé :

Si on scanne en Q alors

Pour i allant de 0 à np faire

Calcul des nouvelles composantes de Q en r.l.u

Calcul des nouveaux angles : appel à *calcTASAngles*

Calcul des nouvelles coordonnées des sommets du triangle : appel à *calcCoordENpixel* et *calcCoordQpixel*

On redessine les deux panneaux, en attendant 5 secondes

Sinon si on scanne en EN alors

Si ki fixé alors

Pour i allant de 0 à np faire

Calcul de kf en Angstroms inverses en fonction de EN et ki

Calcul des angles : apple à *calcTASAngles*

Calcul de la nouvelle coordonnée de EN : *calcCoordENpixel*

On redessine les deux panneaux, en attendant 5 secondes

Sinon si kf fixé alors

Pour i allant de 0 à np faire

Calcul de ki en Angstroms inverses en fonction de EN et kf

Calcul des angles : apple à *calcTASAngles*

Calcul de la nouvelle coordonnée de EN : *calcCoordENpixel*

On redessine les deux panneaux, en attendant 5 secondes

Fin\_si

Fin\_si

Fin

Explications de lignes en rouge :

* Scan en Q, calcul des nouvelles composantes de Q

Tous les calculs sont effectués en r.l.u. L’utilisateur rentre les composantes des valeurs minimales et maximales du scan. On a donc en données QHmin,QKmin,QLmin,QHmax, QKmax,QLmax.

On va alors calculer np-2 valeurs intermédiaires de QH, QK et QL qui vont relier Qmin et Qmax selon une droite.

On a alors facilement la formule

**QH = Qhmin+(Qhmax-Qhmin)\*i/np où i est la variable de boucle.**

On peut vérifier que pour i = 0, on a bien QH= Qhmin et pour i = np on a QH= Qhmax. On calcule de même QK et QL.

* Scan en EN, calcul de la nouvelle valeur de ki ou kf selon le choix du vecteur fixé

Pour cette explication, on va choisir ki fixé mais le principe est le même pour kf fixé.

Ici, l’utilisateur rentre une valeur de EN minimale et maximale et on a déjà la valeur de ki, puisqu’elle est fixée.

On rappelle que **EN = (ki²-kf²)\*ENF** où ENF permet juste de convertir en meV.

On a alors kf = √(ki²-EN) \*ENF

Ici, le EN intermédiaire est (ENmin+(ENmax-ENmin)\*i/np

On a donc alors la formule suivante :

**kf = √ (ki\*ki-(ENmin+(ENmax-ENmin)\*i/np)\*ENF);**

Pour le cas kf fixé, on utilise la même méthode, sauf qu’ici, on a ki = √(kf²+EN) \*ENF et donc **ki = √ (kf\*kf+(ENmin+(ENmax-ENmin)\*i/np)\*ENF);**

1. Programmation

# Déroulement général du programme

La difficulté de ce projet, au-delà des notions physiques et cristallographiques, était d’adapter sans cesse les dessins aux paramètres modifiés par l’utilisateur.

On peut résumer le déroulement du programme ainsi :

A chaque nouveau paramètre de maille : par le biais de l’onglet « *CRYSTAL »*

Calcul de la matrice UB : *matrixUB*

Calcul de la matrice inverse UB-1 : *matrixUBinv*

Calcul de *coeffpixel* et des points de la maille à dessiner : *calcMaille*

Calcul de *coeffang*

Dessin de la maille

A chaque nouvelle valeur de ki, kf ou Q : par le choix de l’instrument ou dans le champ de texte

Calcul des angles associés : *calcTASAngles*

Calcul de *coeffpixel* et des points de la maille à dessiner : *calcMaille*

Détermination des coordonnées : *calcCoordENpixel* et *calcCoordQpixel*

Dessin de la maille et du triangle

Dessin de l’instrument dans le réseau direct

A chaque nouvelle coordonnée : avec la souris (freeDrag et scanDrag)

Calcul de ki, kf et Q en pixels et respectivement en Å-1 et en r.l.u : *calcKipixel* et *calcKfpixel*

Calcul des angles associés : *calcTASAngles*

Calcul de *coeffpixel* et des points de la maille à dessiner : *calcMaille*

Dessin de la maille et du triangle

Dessin de l’instrument dans le réseau direct

Le dessin du triangle et de la maille se trouvant dans le même panneau, il a été créé la classe *DessinRec* dérivant de JPanel dans laquelle la fonction *dessine* redessine à la fois les points de la maille et le triangle.

En pratique, à chaque appel à cette fonction, on fait les actions suivantes :

Calcul de la matrice UB : *matrixUB*

Calcul de la matrice inverse UB-1 : *matrixUBinv*

Calcul des angles : *calcTASAngles*

Calcul de l’origine du repère : *calcCoord0pixel* (dépendant uniquement de la taille de la fenêtre)

Calcul de *coeffpixel* et des points de la maille à dessiner : *calcMaille*

Si nouvelle valeur de ki, kf ou Q alors

Calcul des coordonnées : *calcCoordENpixel* et *calcCoordQpixel*  pour positionner le triangle dans le repère graphique de la fenêtre

Fin\_si

Appel aux fonctions graphiques *drawRecLat* (qui dessine les points de la maille) et *drawQkikf*  qui dessine le triangle.

On appelle la fonction *dessine* à chaque fois que la fenêtre est redimensionnée (dans ce cas, seule la valeur de *coeffpixel* change) ou à chaque modification des paramètres de maille, des vecteurs ou des coordonnées (voir ci-dessus).

Au démarrage de l’applet, la fonction *init* est appelée. Celle-ci se charge de créer et de mettre en place les composants de l’interface, d’initialiser les paramètres de maille et d’instrument (*setParamInit*), l’instrument de départ étant IN14 et dessine le spectromètre dans le réseau réciproque, la maille et le triangle.

* 1. La gestion des évènements

L’utilisateur interagit de façon continue avec le logiciel, par le biais des champs de texte, des menus, des boutons ou de la souris.

Ces différents composants de l’interface ne répondent pas de la même façon aux évènements. On peut distinguer 4 types de réactions :

\*Les mouseMotionListenner :

Gèrent tous les évènements liés à la souris. Ici, ils sont permis uniquement sur le panneau du dessin réciproque.Ceux sont eux qui modifient directement les coordonnées du triangle par exemple.

* Les KeyEvent :

Gèrent les évènements liés au clavier, et plus particulièrement aux champs de texte.

Le texte tapé dans les champs de texte accessibles (non grisés) est récupéré par simple validation (touche ENTREE).

Lors de la mise à jour des paramètres de maille (angles et longueurs), on doit gérer les propriétés liées aux familles de maille.

* Les ChangeEvent:

Gèrent les évènements liés aux menus non déroulants : les JSpinner. Ceux-ci permettent uniquement de choisir le sens de diffusion de chacun des composants du TAS. A noter qu’un JSpinner est en fait un champ texte associé à une barre de déroulement.

* Les ActionEvent:

Gèrent les évènements liés aux boutons, bouton radio et menus déroulants.

Les évènements liés aux boutons radio déterminent le mode de scannage et font donc varier une simple variable booléenne.

Les évènements liés aux menus déterminent les conditions de l’expérience (type de cristal et instrument).

Il y a enfin les évènements liés aux boutons des murs, su scan automatique ou de la réinitialisation.

1. Les murs
   1. Le dessin des murs

La notion de murs est introduite dans l’onglet “INSTRUMENT”. Pour résumer, l’utilisateur peut dessiner, à la main ou automatiquement des murs autour de l’instrument dans sa configuration directe. Il peut également modifier ces murs (tout effacer ou effacer le dernier). Ces fonctionnalités sont détaillées dans la partie correspondant à l’onglet “INSTRUMENT” du paragraphe “les onglets”.

Les points des murs choisis par l’utilisateur lors du tracé manuel sont déterminés à chaque fois que celui-ci relâche la bouton gauche de la souris. Nous avons en effet prévu une fonctionnalité qui permet à l’utilisateur de savoir, lorsqu’il clique sur un point et garde le bouton gauche appuyé, à combien de cm se trouve le point par rapport au monochromateur. Ainsi, tant qu’il n’a pas relâché la souris, le point n’est pas tracé mais les distances en x et en y sont données à droite du panneau inférieur, sous les angles.Il peut alors se promener sur ce panneau pour choisir l’endoit exact du coin du mur qu’il souhaite tracer.

Nous avons choisi de représenter ces murs selon les coordonnés des segments les constituant.

Ces coordonnées sont rangées dans un tableau de deux colonnes (appelé *coord\_walls*), la première colonne contenant les coordonnées en x et la deuxième les coordonnées en y. Pour parcourir et déterminer le nombre d’éléments du tableau, on utilise la variable *compteur.*

La variable booléenne *show\_walls* est crée afin de déterminer si l’utilisateur a choisi de tracer les murs. En effet, le dessin proprement dit de ceux-ci se trouve dans la routine *drawWalls,* appellée à chaque *repaint* de la classe *DessinTas.* Donc, si l’utlisateur n’a encore rien tracé mais redimmensionne la fenêtre, on ne doit pas tracer les points et les murs.

Ainsi, lors de l’appel de ce *repaint*, si *show\_walls* est faux, on dessine juste l’instrument, sinon, on dessine en plus les murs.

Nous allons maintenant préciser le code correspondant à chaque bouton concernant les murs.

* Le bouton “instrument walls”

Selon l’instrument choisi

Remplissage du tableau *coord\_walls*

Mise à jour de *compteur* ( = nombre de points)

Show\_walls = vrai

Nouveau dessin de l’instrument dans le réseau direct et des murs

* Le bouton “show walls”

Ce bouton permet de relier les points tracés avec la souris, détectés dans la routine *abonnements,* dans la partie *mouseReleased* de *monDessinRec*. Dans la routine *showWalls*, on met uniquement la variable *show\_walls* à vrai et on redessine le panneau.

* Le bouton “delete last point”

Ce bouton permet d’effacer le dernier point tracé, et donc également le dernier mur.

Si ce n’est pas le dernier point à enlever (le premier tracé) alors on décrémente *compteur.*

Si *compteur* est nul, alors *show\_walls* est faux. Ainsi, tous les murs sont effacés.

On redessine le panneau contenant les murs et l’instrument dans le réseau direct.

* Le bouton “delete all points”

Ce bouton permet d’effacer tous les murs et les points tracés manuellement ou automatiquement.

On met alors uniquement *compteur* à nul et *show\_walls* à faux et on redessine.

La détermination des points des murs tracés automatiquement se fait très simplement en récupérant les coordonnées du point relâché et en remplissant le tableau *coord\_walls* avec ces coordonnées. On incrémente la variable de parcours (*compteur*) et on dessine le point.

Pour indiquer à l’utilisateur la position de son point par rapport au monochromateur lorsqu’il presse la souris (*mousePressed*) ou lorsqu’il la bouge tout en la maintenant pressée (*mouseDragged*), on récupère les coordonnées du point et on lui enle=ève les coordonnées du monochromateur. On a ainsi la distance en pixels entre ces deux points et pour la convertir en cm, on multiplie par la variable *echellepix* qui dépend de chque instrument (voir la partie Détermination des murs), puis on affiche les deux valeurs trouvées.

Remarque :

On a réduit à 20 le nombre de points possibles à tracer (par choix). Ainsi, lors du tracé manuel, on teste si l’utilisateur peut encore tracer un point et on l’en informe pas un message.

* 1. La détermination des murs

La détermination des coordonnées des murs entourant un spectromètre est basée sur les plans (voir annexe) fournis par les responsables de chaque spectromètre. On procède de la façon suivante :

1- Chaque plan comporte une échelle notée dans le programme *echelleplan.*

2- On prend comme référentiel le référentiel orthonormé déterminé à partir du faisceau de neutrons incident et ayant pour origine le monochromateur.

3- On mesure à la règle la distance entre les coins de murs et le monochromateur.

4- En multipliant cette valeur par *echelleplan* on a donc la vraie distance en cm entre les deux points.

5- On détermine ensuite, pour chaque instrument, un coefficient noté *echellepix* qui permet de dessiner de manière correcte et avec à peu près la même taille les différents instruments dans le panneau inférieur.

6- En multipliant par ce coefficient, on a donc la distance en pixels du point considéré au monochromateur.

7- Pour passer du référentiel formé par le monochromateur et le référentiel de l’écran, on ajoute ou on retranche la coordonnée du monochromateur dans ce référentiel.

Le tableau ci-dessous résume les différents coefficients, attachés aux 4 spectromètres trois-axes de l’ILL que l’on a choisi de prendre en compte.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Instrument | IN14 | IN8 | IN1 | IN20 |
| Echellepix | 1/3 | 1 | 1/3 |  |
| Echelleplan | 30.6 | 20 | 63 |  |

Les murs entourant le spectromètres trois-axes IN20 ne lui imposant pas de contraintes particulière, on ne définit ni tableau, ni coefficients pour cet instrument.

Les tableaux ci-dessous donnent, pour chaque instrument, les coordonnées des coins des murs dans le référentiel défini ci-dessus, en cm sur le plan.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | X | y |  |  | x | y |  |  | x | Y |
| IN8 | -4 | -3.3 |  | IN14 | -1.6 | -3.5 |  | IN1 | -1.1 | 1.7 |
|  | -6 | 0 |  |  | -4.1 | -7.8 |  |  | -1.3 | -3 |
|  | -8.5 | 5 |  |  | -12.7 | -9.5 |  |  | -2.7 | -4 |
|  | -4 | 7.8 |  |  | -18.8 | 3.1 |  |  | -1.8 | -7 |
|  | 0 | 8.8 |  |  | -9 | 13 |  |  | 5.8 | -6.3 |
|  | -1 | 13.2 |  |  | -4.3 | 13 |  |  | -5.8 | 0 |
|  | 2 | 13.2 |  |  | -4.3 | 9.4 |  |  | 1.5 | 2.6 |
|  | 1.4 | 0 |  |  | -3.1 | 9.4 |  |  |  |  |
|  |  |  |  |  | -3.1 | 7 |  |  |  |  |
|  |  |  |  |  | -1.5 | 3.5 |  |  |  |  |

1. Les imprécisions

La manipulation interactive du triangle avec la souris correspond au cahier des charges, cependant on s’aperçoit vite que cette manipulation doit se faire avec précaution, sinon les déplacements des divers éléments deviennent vite incompréhensibles. On peut également parfois observer que le mouvement du triangle n’est pas fluide.

Tous ces défauts sont dûs à des imprécisions numériques (cas limites) résultaant de la taille finie du pixel graphique. On peut voir dans le code les incessantes conversions entre les pixels et les unités physiques (Å-1 et r.l.u), donc entre des entiers et des réels, avec des pertes de précision inévitables. Nous avons essayé de limiter les effets désagréables associees, mais qu’on ne peut malheureusement pas tous les supprimer.

Explication des variables xmouse, ymouse :

Pour pallier à la difficulté d’être précis avec la souris, la zone de détection d’un clic sur un sommet libre du triangle est déterminée par un carré centré autour de ce point et de 10 pixels de côté. Ainsi, même si on clique un peu à côté du point visé, celui-ci est validé.

Dans ce cas, il faut gérer cette différence entre les vraies coordonnées du sommet et les coordonnées du point cliqué, sinon, le dessin risque de faire un saut.

Dans ce but, on a crée les variables *xmouse* et *ymouse,* que l’on met à jour à chaque clic de souris (*mousePressed*). Elles contiennent la différence entre la coordonnée du point cliqué et celle du point du sommet, horizontalement et verticalement.

Ensuite, lors de la gestion du mouvement du triangle (*mouseDragged*), on fait l’opération inverse.

# V- Intégration de l’applet dans une page HTML

Netbeans crée automatiquement un fichier html affichant l’applet. Ce fichier se trouve dans le répertoire “build” du projet, au même endroit que les fichiers .class.

A chaque compilation, ce fichier est mis à jour.

Nous avions d’abord prévu de lire les paramètres instrumentaux à partir d’un fichier pouvant être facilement modifié sans accéder au code. Cependant, les applets sont très sécurisées et permettent difficilement ce genre de manipulation. Après quelques recherches, nous avons découvert que ceci nécessiterait de “signer”. Faute de temps, ceci a été abandonné et nous avons préféré une solution plus triviale, mettre directement en paramètres dans la page HTML les données instrumentales. Cette solution simple ne pose pas de problèmes, sinon esthétiques, tant qu’il y a peu d’instruments et de paramètres à prendre en compte.

Voici le code permettant de générer cette page :

<HTML>

<HEAD>

<TITLE>vTAS</TITLE>

</HEAD>

<BODY>

<H3><HR WIDTH="100%">virtual Three Axis Spectrometer<HR WIDTH="100%"></H3>

<P>

<APPLET codebase="./classes/" code="VTAS\_html.class" width=950 height=630>

<param name = IN1DM value="1.807"/>

<param name = IN1DA value="1.807"/>

<param name = IN1ki value="5.477"/>

<param name = IN1kf value="5.477"/>

<param name = IN1QH value="2.0"/>

<param name = IN1QK value="0.0"/>

<param name = IN1QL value="0.0"/>

<param name = IN8DM value="3.185"/>

<param name = IN8DA value="3.185"/>

<param name = IN8ki value="4.1"/>

<param name = IN8kf value="4.1"/>

<param name = IN8QH value="2.0"/>

<param name = IN8QK value="0.0"/>

<param name = IN8QL value="0.0"/>

<param name = IN14DM value="3.35"/>

<param name = IN14DA value="3.35"/>

<param name = IN14ki value="2.0"/>

<param name = IN14kf value="2.0"/>

<param name = IN14QH value="1.0"/>

<param name = IN14QK value="0.0"/>

<param name = IN14QL value="0.0"/>

<param name = IN20DM value="3.437"/>

<param name = IN20DA value="3.437"/>

<param name = IN20ki value="4.1"/>

<param name = IN20kf value="4.1"/>

<param name = IN20QH value="2.0"/>

<param name = IN20QK value="0.0"/>

<param name = IN20QL value="0.0"/>

</APPLET>

</P>

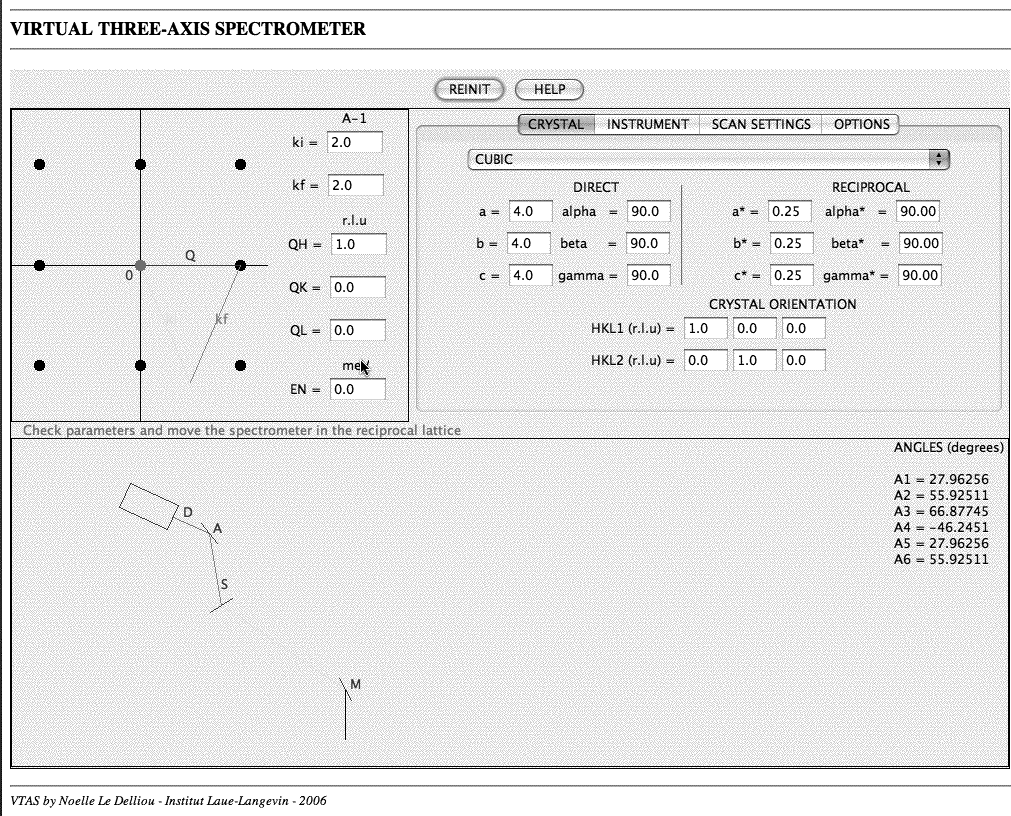
<HR WIDTH="100%"><FONT SIZE=-1><I>Institut Laue-Langevin - Noelle Le Delliou - 2006</I></FONT>

</BODY>

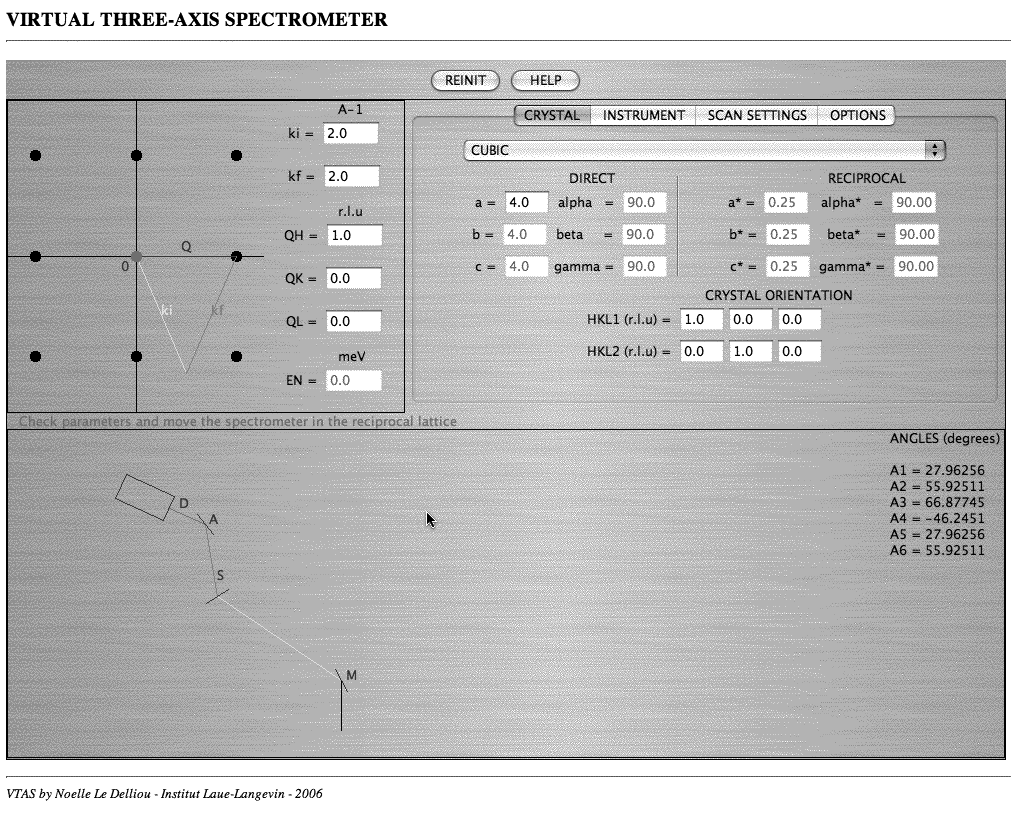
</HTML>

Cette page s’ouvre avec les navigateurs les plus fréquents à l’ILL : Firefox et Safari (navigateur apple).

Avec Firefox :



Avec Safari :



# Remarque:

Pour que la fenêtre s’affiche en plein, si possible en prenant le plus de place dans l’écran , la taille de celle-ci est fixée à 950\*630. Il faut en effet prévoir les barres de tâches du navigateur par exemple.

# VI- Ce qu’il reste à faire

Les objectifs initiaux ont été largement dépassés, puisque le but était de moderniser le code et l’interface et surtout faire marcher l’ancienne version. Diverses petites fonctionnalités ont été rajoutées, comme le dessin des murs par exemple.

Nous avons cependant imaginé des extensions possibles au logiciel actuel.

1- Dessin de la courbe de dispersion et de l’ellipsoïde de résolution ; dessin de la forme du pic correspondante

Ces points de l’ancienne version n’ont pas été repris mais sont prévus dans le code. Un panel est prévu pour accueillir ces deux graphiques ainsi que deux classes représentant ces deux dessins. Le code original concernant ces parties est réutilisable mais il reste à vérifier les calculs, les méthodes et à introduire ces parties dans le nouveau code.

2- Améliorations esthétiques

On peut également prévoir diverses améliorations que des images, des icones, des couleurs …

3- Lecture des paramètres instrumentaux

Pour l’application indépendante, les paramètres sont lus dans des fichiers.

Pour l’applet, les informations sont sous forme de paramètres dans la page HTML qui la contient. On peut prévoir de lire les paramètres dans des fichiers, comme dans le cas de l’application indépendante, mais c’est plus compliqué que la solution actuellement retenue (cf. §V).

4- Applications indépendantes exécutables sous Windows et Linux

Le .app actuel n’est exécutable que sous Mac OS et nous n’avons pas eu le temps de réfléchir à la création d’un .exe, lisible sous Windows et d’un .bin pour Linux.

5- Aide en ligne

Le présent rapport constitue la documentation du programme. La mise en œuvre de l’aide en ligne n’a pas pu être amorcée. Le code prévoit un bouton “HELP” en haut de la fenêtre qui fera alors le lien avec cette aide, programmée sous forme de JavaHelp (voir le tutorial dédié à cette partie dont le lien est donné dans la partie précédente).

6- Paramètres de maille dans le réseau direct ou réciproque

Actuellement, on peut uniquement rentrer ces paramètres directs et le programme calcule les paramètres réciproques. On pourrait donner le choix à l’utilisateur de rentrer, au choix, les valeurs des paramètres directs ou réciproques. Il faudrait alors simplement implémenter une routine appelée *calcMailleDirect*  qui ferait le calcul inverse de la routine *calcMailleRec.*

# VII- Annexes

# Où trouver des informations

\* aides sur la physique :

site de l’ILL : <http://www.ill.fr/>

Encyclopédie scientifique : <http://fr.wikipedia.org/>

Cédérom Exploring matter with neutrons

\* aides sur l’informatique

Java doc : <http://java.sun.com/j2se/1.4.2/docs/api/>

Tutoriaux et forums pour des problèmes divers : <http://java.developpez.com/>

Tutorial sur le JavaHelp : <http://cyberzoide.developpez.com/java/javahelp/PLAN>

\*aides diverses

code du logiciel TASMAD

1. Plans de IN14, IN1, IN8