(see Appendix C, Important Scan Types) is designed to facilitate this.

The program is also responsible for four standard corrections to the data:

1. area correction, due to the changes in the active area of the sample as the diffractometer angles change, introduced by Robinson (1990b);
2. Lorentz factor correction, to convert an angular scan into an integral in reciprocal space, as in Warren (1969);
3. polarization correction, as in James (1950); and
4. monitor and stepsize normalization to account for different counting times and scanning rates that may have been used for different parts of the data.

Under the *c(olumn)* menu, it is necessary that all the diffractometer angles be identified by their column sequence number in the scan file. These must not change from one scan to the next.

The remaining parameters are entered at the command line. The *le(ft)* and *ri(ght)* numbers of points are to be considered as background; all points in between are integrated numerically to a total intensity. The *s(cantype)* can be *r(ocking)* or *i(ndex)* and will affect the Lorentz factor; all other scans found in the batch input will be ignored. For the area correction, the incident beam is taken to have dimensions *wi* vertically by *hi* horizontally, the sample has width *ws*, and the vertical exit beam slit is *we*.

The *r(un)* command followed by starting and ending scan numbers will perform the integration. Two output files are generated. The *.inf file has a line of information about each scan found, such as the date, background, and peak levels. The *.pk file contains the scan number hkl, the structure factor, and the structure factor error as the first three columns.

Averaging Procedure AVE

This procedure uses the *.pk output of PEAK to locate and compare all the symmetry equivalents. It determines an overall estimate of the average systematic error, ε, assumed to be a constant fraction of each individual structure factor value. The value of ε is taken to be a quality factor (R value on structure factors) in assessing the data, as discussed in Robinson (1990b). The program then passes through the data a second time to generate a weighted average value (file *.wgt) of each inequivalent structure factor with an enlarged error bar that combines its statistical errors input with its overall systematic error estimated in the first pass.

The *.ave output listing is very useful for finding bad data. It lists together all reflections that are equivalent according to the specified symmetry. Any that fall out of line are flagged with warnings indicating they should be checked. They are identified by scan number in the listing. This is also a useful way of testing the symmetry of the data, if this is unknown from context, since the program can be run with different preassigned choices of assumed symmetry.

The *s(et)* menu has submenus for defining *c(olumns)*, *s(ymmetry)* from the list of all 17 possible plane groups from p1 to p6mm, and *a(veraging)* parameters and modes. Under *a(veraging)* is the *c(utoff)* parameter, which is the number of times that a structure factor must be larger than its statistical error for it to be used in the estimation of ε. The default value (number of standard deviations) is 2σ, and this should be increased to 10σ if the data are not strongly affected by counting statistics.

Fitting Procedure ROD

This is a large refinement program for fitting a structural model to crystallographic data. It is specific to surface diffraction in that it calculates CTR rather than bulk diffraction structure factors. It differentiates the “bulk” and “surface” parts of the structure and includes a simple description of surface roughness that is usually necessary to get a good fit. The bulk cell is periodic and invariant; it only contributes to the CTRs. The atoms in the surface cell can be selectively refined in position, occupancy, and Debye-Waller factor. It is important that the z coordinates in the surface be a smooth continuation from those of the bulk.

The input files needed for ROD are as follows. All are expected to have a header line that is ignored on input. The second line of the first two files must contain the six unit cell parameters in real space, a, b, and c (in angstroms) and α, β, and γ (in degrees). The program checks that both are the same:

1. bulk unit cell coordinates, *.bul;
2. surface unit cell coordinates, *.sur (fixed), or *.fit (parameterized);
3. structure factor data, *.dat (renamed from *.wgt above); and
4. model parameters, *.par (optional).

In order that small data sets can be compatible with large models, ROD allows a very flexible parameterization scheme. The same displacement, occupancy, and DW parameters can be assigned to more than one atom. This is very convenient for structures that contain repeated motifs or different symmetries of components within the whole unit cell. In fact, the program does not handle symmetry at all, since this can always be built into the model; all structures are assumed to have triclinic, 1 × 1 unit cells.

APPENDIX C:
SUPER COMMAND FORMAT

Commands have a variable number of arguments, depending upon the specific application. Arguments follow on the same line as the command and are separated by spaces or commas. Most commands will prompt the user for the required arguments if no arguments are given. Multiple commands may be placed on one line provided they are separated by a semi-colon.

Values of important control parameters are stored as “variables.” To obtain the current value of a variable, type its name followed by =. To change the value stored in a variable, type the new value after the =, e.g., as = 1.65. Rudimentary algebra is also allowed, e.g., as = bs * 3. Space characters are required as delimiters around the mathematical operation.

A manual of allowed commands will be available at the beamline. A list of all valid "super" commands can be obtained by typing "help" at the "super" prompt or using the F8 key. Pressing F5 lists the orientation matrix and the values of string variables. The values of all other variables are listed using F6.

Basic Super Commands

mv	Move motors (four arguments: TTH, TH, PHI, CHI).
ct	Count (one optional argument = preset).
br	Go to position defined by H, K, and L (three arguments).
wh	Print current H, K, and L and values of the motor angles (no arguments).
ca	Calculate motor angles from H, K, and L (three arguments).
ci	Calculate H, K, and L from motor angles (five arguments).
end	Exit the program.

Basic Super Variables

as	Crystal reciprocal lattice parameter a^*
bs	Crystal reciprocal lattice parameter b^*
cs	Crystal reciprocal lattice parameter c^*
al	Crystal reciprocal lattice angle α^*
be	Crystal reciprocal lattice angle β^*
ga	Crystal reciprocal lattice angle γ^*
wv	Incident wave vector ($2\pi/\lambda$)
nobs	Number of alignment reflections used to calculate OM
h1, k1, l1, h2, k2, . . .	HKLs of orientation reflections
t1, u1, p1, c1, a1, u2, p2, . . .	2-Theta, theta, phi, chi, and alpha of orientation reflections
B	Target value of incidence angle, α_i
B2	Target value of exit angle, α_f
alm	Mode of calculating ALP for five-circle geometry (alm = 1 means ALP is frozen at its current value; alm = 2 means ALP is calculated)
bem	Mode of constraining incidence angle for five-circle geometry (bem = 1 means *incidence* angle α_i is fixed at target value B; (bem = 2 means *exit* angle α_f is fixed at target value B2; bem = 3 means incidence and exit angles are made equal)
fchi	CHI angle of optical surface orientation (laser alignment)
fphi	PHI angle of optical surface orientation (laser alignment)

Orientation Matrix

There are two ways of achieving the sample orientation, depending on whether the reciprocal lattice parameters are known or unknown. In the more commonly used "lattice parameters known" mode, the six reciprocal lattice parameters are entered manually using syntax name = value. The names of the six reciprocal lattice constants are as, bs, cs, al, be, and ga. Two orientation reflections are required with h, k, l, TTH, TH, PHI, CHI, and ALP specified for each reflection. The meaning of the angles is slightly different in different modes of calculating the orientation matrix. We refer here only to the five-circle mode, which is set by the command *frz 5*.

Individual variables in the orientation matrix can be changed using syntax name = value. The variable names in this case consist of a single letter and a number (e.g., t1, t2, t3, . . .). The number refers to the number of the orientation reflection. The letters are assigned as follows for the five-circle mode: t = 2-theta, u = theta, p = phi, c = chi, a = alpha, h,k,l = Miller indices, g = gamma (fixed out-of-plane detector position), and z = zeta (fixed sample offset angle).

The orientation reflections may also be entered using the *or* (orient) command. To enter the parameters for the current diffractometer position, type *or* followed by the number of the reflection (in this case 1 or 2) and the hkl. One may also manually enter all variables on one line by typing

or # h k l TTH TH PHI CHI ALP

The orientation matrix is a general 3 × 3 matrix with 9 degrees of freedom described by Busing and Levy (1967). Six degrees of freedom are constrained by the reciprocal lattice parameters, 2 more come from the first orientation reflection (the two polar angles), and the final degree of freedom comes from the second orientation reflection (the second reflection specifies the azimuthal rotation about the first vector). The direction of the first orientation reflection, called the primary reflection, will always be reproduced exactly. The angles of the second reflection may not necessarily be consistent with the lattice parameters, and consequently the program may not reproduce the angles of the secondary reflection exactly. The *sor n m* command allows reflections n and m to be swapped.

The "lattice parameters unknown" mode is set by toggling the *om* command. Then the six reciprocal lattice parameters are determined directly from the five angles specified for each of the orientation reflections (TTH, TH, PHI, CHI, and ALP). To define the nine matrix elements of the orientation matrix, at least three non-coplanar reflections must be used, so nobs = 3. If nobs3, the best matrix is constructed from the full list of nobs reflections using a least-squares fit and will usually improve as more reflections are added. Reciprocal lattice parameters that are consistent with the three observed reflections will then be calculated. In general, the reciprocal lattice parameters will correspond to a triclinic cell with interfacial angles nearly equal to those expected from the symmetry. As guide to selecting suitable reflections, an error is calculated for each, which is the distance (in reciprocal angstroms) of the reflection from the reciprocal lattice point derived from the orientation matrix. The orientation parameters may be entered manually or by using the *or* command.

Important Scan Types

ascan	Angle scan—Scan four diffractometer motors TTH, TH, PHI, CHI.
mscan	Motor scan—Scan up to three arbitrary motors.
iscan	Index scan—Input three starting HKLs and three delta HKLs for the increments between points.
jscan	Centered index scan—Like iscan except scan is *centered* on the given starting hkl value.
kscan	Centered two-dimensional index scan—Like jscan it is *centered* on the given starting hkl value. Two sets of delta HKL step sizes are given.
rscan	Rocking scan—Input HKL value, motor number (or hkl) and one delta value for the increment. The scan moves either one motor or one hkl component and is centered at the given HKL value.
lup	Line-up scan—Input motor number and delta angle. Similar to a rocking scan, but the scan is centered at the current position of the motors. At the end of the scan, the motors move to the peak position, calculated as the center of gravity of the counts measured.
fpk	Find peak scan—Like lup, except fewer para meters are required and the data are not saved on disk. After the scan, the motors moved to the peak calculated from the first moment.
vscan	Variable scan—Input HKL, variable name, and starting value. The scan increments the value of the variable and moves the motors to the given HKL.
vlup	Centered vscan—Input HKL, variable name, delta, and npts. The scan increments the value of the variable and moves the motors to the given HKL. The scan is centered on the current value of the variable beforehand, which is reset to the center of gravity afterward.
pkup	Peakup reflection—The scan does a series of lup scans centered at each orientation reflection. At the end of the series, the angles of the orientation reflection are updated to reflect the new peak position.

The scan types *rscan* and *jscan* embody automatic “stretching”’ of the background, whereby the step size at the beginning and end of the scan is increased by a factor of 3. The *lup* and *vlup* scans are very useful during the alignment stages of the experiment because of their autocentering feature; they can also be used to make automatic realignments during an unattended batch procedure. All scans generate a video display of the data as they accumulate, which can be viewed at any time using the function keys: F1 (F2) to see the spectrum of the multichannel analyzer as it accumulates and F3 (F4) to see a histogram of the recorded counts on a linear (log) scale.

A scan may be executed by typing the command name followed by a list of arguments. To obtain help on the command format, type the name of the scan without arguments and the program will prompt you. Any scan can be interrupted during execution by ^C and continued by typing *co*. A series of commands (including scans) can be executed by using either the *sc* command or the *ex* command.

To use the *sc* command, one first has to enter a list of commands into the scan table. The list can contain almost any command executable from the command line. A scan list can be created in three ways. The most direct method is to use the *ds* (define scans) command. Alternatively, one can use the *es* (edit scans) command to edit the scan table directly. The command *gs* filename (get scans) will read a new scan table file. Once the scan table has been entered, a list of commands or scans can be executed with the *sc* command, which accepts a series of numbers in arbitrary order corresponding to the position of the command in the scan table. Before execution begins, the program does a “dry run” to check for setting calculation errors and limit problems. If the execution is halted, the *co* command will resume execution.

A series of commands or scans can also be executed with the *ex* command. This command diverts program input to a script file and executes the commands in order. There is no limit to the number of commands in the file. In contrast to *sc*, the *ex* command does not do a dry run before execution begins. If a problem is encountered during execution, the program skips to the next command in the list.

I. K. Robinson
University of Illinois
Urbana, Illinois

X-RAY DIFFRACTION TECHNIQUES FOR LIQUID SURFACES AND MONOMOLECULAR LAYERS

INTRODUCTION

X-ray and neutron scattering techniques are probably the most effective tools employed to determine the structure of liquid interfaces on molecular-length scales. These are not different in principle from conventional x-ray diffraction techniques that are usually applied to three-dimensional crystals, liquids, solid surfaces etc. However, special diffractometers that enable scattering from fixed horizontal surfaces are required to carry out the experiments. Indeed, systematic studies of liquid surfaces had not begun until the introduction of the first liquid surface reflectometer (Als-Nielsen and Pershan, 1983).

A basic property of a liquid-gas interface is the length scale over which the molecular density changes from the bulk value to that of the homogeneous gaseous medium. Molecular size and capillary waves, which depend on surface tension and gravity, are among the most important factors that shape the density profile across the interface and the planar correlations (Evans, 1979; Braslau et al., 1988; Sinha et al., 1988). In some instances the topmost layers of liquids are packed differently than in the bulk, giving rise to layering phenomena at the interface. Monolayers of compounds different than the liquid can be spread at the gas-liquid interface, and are termed Langmuir monolayers (Gaines, 1966; Swalen et al., 1987). The spread compound might “wet” the liquid surface to form a film of homogeneous thickness or cluster to form an inhomogeneous rough surface. The x-ray reflectivity (XR)