Journal of
Applied
Crystallography
ISSN 0021-8898

ASSA, a Program for Three-Dimensional Rendering in Solution Scattering from Biopolymers

M. B. Kozin, V. V. Volkov and D. I. Svergun

J. Appl. Cryst. (1997). **30**, 811–815

J. Appl. Cryst. (1997). **30**, 811–815

ASSA, a Program for Three-Dimensional Rendering in Solution Scattering from Biopolymers

M. B. Kozin,[a]* V. V. Volkov[a] and D. I. Svergun[a,b]

[a]*Institute of Crystallography, Russian Academy of Sciences, Leninsky prospekt 59, 117333 Moscow, Russia, and* [b]*European Molecular Biology Laboratory, Hamburg Outstation, EMBL, c/o DESY, Notkestrasse 85, D-22603 Hamburg, Germany. E-mail: saxslab@incr.msk.su*

(*Received 23 July 1996; accepted 30 January 1997*)

Abstract

A program for three-dimensional rendering and interactive model refinement, *ASSA*, is presented. The program permits the simultaneous display and manipulation of low-resolution envelope models obtained from solution-scattering data and the crystallographic or electron microscopic models in the Brookhaven Protein Data Bank format. *ASSA* is coupled with the shape-determination program of Svergun, Volkov, Kozin, Stuhrmann, Barberato & Koch [*J. Appl. Cryst.* (1997). **30**, 000–000], thus allowing one to monitor the shape determination process in real time. The program *ASSA* is written in C using the PHIGS+ standard for the three-dimensional solid rendering and the XView and XLib libraries for the menu-driven user interface. The system runs on a Sun SPARC ZX-20 workstation; a simplified IBM-PC version is also available.

1. Introduction

Recent developments in data-interpretation methods allow the effective analysis of X-ray and neutron scattering data from biopolymers in solution in terms of three-dimensional (3D) models. Thus, given the solution scattering curve, a low-resolution particle shape can be restored (Svergun, Volkov, Kozin & Stuhrmann, 1996; Svergun *et al.*, 1997). Combining solution-scattering and crystallographic data, positions of domains in complex particles can be established (Svergun, 1997, and references therein). Alternatively, electron microscopic models can also be used as initial approximations or for the validation of the results (Svergun, 1997, and references therein).

New data-analysis methods require adequate tools to represent their results. Low-resolution models are best represented as solid bodies by rendering the particle surface. It should be possible to display and manipulate these models in 3D together with the atomic (or pixel-type) models obtained from crystallographic and electron microscopic studies where these are available. Moreover, validation of the model usually requires its automatic or interactive refinement, that is, the visualization program should be able to communicate with the corresponding data-analysis programs.

The main requirements of the 3D rendering software for solution scattering are as follows. (i) It should be able to represent any configuration of 3D bodies which can freely overlap and to display simultaneously crystallographic structures or electron microscopic models. Moreover, it should provide information about the absolute position of each body. (ii) The position and orientation of each selected individual body or group of bodies should be interactively modifiable. (iii) It should be possible to make parts of the surfaces transparent in order to display the interpenetration of the objects. (iv) It should be able to evaluate scattering curves from individual objects or groups of objects and to represent the evolution of these curves in real time as the object is being changed. (v) It should automatically refine the low-resolution models based on experimental data but provide interactive control of the refinement process. (vi) It should be portable to different hardware platforms and the hardware resources should be rationally utilized by flexible configuring of the program.

Existing 3D visualization software packages used in protein crystallography and electron microscopy are not suitable for our purposes. Most of the crystallographic packages, *e.g. SHAKAL* (Keller, 1989) or *O* (Jones, Zou, Cowan & Kjeldgaard, 1991) generate either wireframe or ball-and-stick models, whereas the programs for electron microscopy use various data formats and are oriented towards the shaded surfaces. The software package *ASSA* described below was developed in the framework of the creation of an expert system for solution-scattering data analysis from biopolymers. It permits the simultaneous display of solid-type and pixel-type models and the refinement of low-resolution models using the solution-scattering data.

2. Particle shape and the scattering intensity

The algorithms used to represent the particle envelope and to evaluate the solution-scattering intensity will first be briefly discussed. The shape of a wide variety of homogeneous particles at low resolution can be conveniently described with the help of the angular envelope function $r = F(\omega)$, where $(r,\omega) = (r,\varphi,\theta)$ are spherical coordinates (Stuhrmann, 1970). This class of low-resolution models proved to be especially useful for the interpretation of solution scattering data (Svergun, 1997).

Generally, one needs $3N$ coordinates to describe the model of a 3D surface consisting of N points. Taking advantage of the surface representation by the angular shape function, a set of points on the surface of the unit sphere is generated which yields $2N$ angular coordinates ω_i. Triangulation of the set of points provides the information on how these points should be combined in order to get the model ribs and facets (Rodgers, 1985). Every shape can then be described by only $N+3$ parameters: the coordinates of the centre point (usually the centre of mass) and the N distances $r_i = F(\omega_i)$ from the centre to the particle surface. This significantly improves the performance of the shape-representation procedure.

The unit sphere is triangulated using a quasi-uniform grid generated by the following algorithm (Svergun, 1994)

$$\left.\begin{array}{ll}\theta_i & = \arccos[1-2(i-1)/f_k] \\ \varphi_i & = 2\pi \operatorname{mod}[(i-1)+f_{k-1},f_k]/f_k\end{array}\right\}, i = 1,\ldots,f_k+1 \tag{1}$$

where f_k is the kth Fibonacci number, defined as $f_k = f_{k-1} + f_{k-2}, f_0 = f_1 = 1$. Depending on the desired resolution, grids with $6 \leq k \leq 18$ (f_k = 14 to 2585) can be taken.

The first program of the graphics package, *TRIANG*, triangulates the unit sphere given the order of the Fibonacci grid and saves the relevant information into auxiliary files. Fig. 1 shows the triangulation of a sphere with Fibonacci grid at k = 14 (f_k = 611). Note that for each Fibonacci number, the triangulation has to be performed only once as it depends only on the directions ω_i and not on the particle shape.

For the purpose of shape determination the particle envelope is parameterized as

$$F(\omega) \cong F_L(\omega) = \sum_{l=0}^{L} \sum_{m=-l}^{l} f_{lm} Y_{lm}(\omega) \tag{2}$$

where $Y_{lm}(\omega)$ are spherical harmonics, f_{lm} are the (complex) multipole coefficients and the truncation value L determines the resolution of the shape representation. For the given set of the multipole coefficients (*i.e.* the given shape), the solution-scattering intensity is readily evaluated as

$$I(s) = 2\pi^2 \sum_{l=0}^{\infty} \sum_{m=-l}^{l} |A_{lm}(s)|^2, \tag{3}$$

where s is the modulus of the momentum transfer vector **s** [$s = (4\pi/\lambda)\sin\theta$, λ is the wavelength and 2θ is the scattering angle]. The partial amplitudes $A_{lm}(s)$ are expressed as a power series of the shape coefficients and the intensities are evaluated by the program module *FLM2IS* (see Svergun, 1997, for description of the algorithm). A more general program, *FOURBODY*, evaluates the scattering intensity from a complex particle consisting of up to four arbitrarily positioned subunits, each represented by its own shape function, as described by Svergun (1994, 1997).

The program *FLM2SLD* evaluates the coordinates of the surface points from the set of coefficients f_{lm} for the given order of the Fibonacci grid and saves them into a binary file together with the name of the file containing the appropriate triangulation information.

3. 3D rendering

3D representation is performed in the program *ASSA*, the main program of the package. The name is an abbreviation of 'Analyse Solution Scattering Automatically', as the program is designed to be a central part of an automated system for the solution-scattering data analysis. *ASSA* is written in C and runs on a Sun SPARC ZX-20 workstation, but is designed in principle for any Unix workstation. It uses the PHIGS (programmer's hierarchical interactive graphical system) standard for 3D solid rendering and the XView and XLib routines for the menu-driven user interface. *ASSA* can represent up to 50 different graphic entries simultaneously, each entry being provided with its own drawing features such as colour, transformation, transparency, surface and edge representation mode *etc.*

3.1. *Input formats*

There are two different formats available for *ASSA* input files and, correspondingly, two different types of graphic entries. The first is the envelope format, either as an ASCII file containing the f_{lm} coefficients (*.flm, file with an extension 'flm'), or as a binary file (*.sld) containing the surface representation. In this case, the graphic entry has the form of a solid body and can be

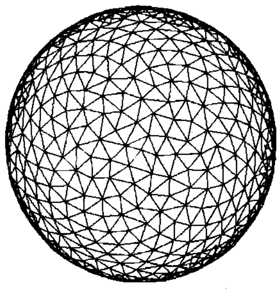


Fig. 1. Triangulation of the unit sphere evaluated by the program *TRIANG* on the Fibonacci grid with k = 14 (f_k = 611).

represented in different modes: wireframe, lighted surface and Gouraud shaded surface.

The second is the standard Protein Data Bank (PDB) format (Bernstein *et al.*, 1977). The graphic entry is the atomic model represented either by the entire set of atoms, or by its C_α backbone. The PDB format (file *.pdb) can also be used to read and display electron microscopic models. Fig. 2 illustrates different modes of shape/atomic structure representation using the structure of lysozyme (Diamond, 1974). The atomic coordinates of lysozyme were taken from the Protein Data Bank, entry 6lyz. The f_{lm} coefficients describing the protein envelope were evaluated and stored in the *.flm file using the program *CRYSOL* (Svergun, Barberato & Koch, 1995).

3.2. *Grouping and transformation*

Every graphic entry (shape or structure) is organized as a separate object with its own graphic features and local transformation matrix. Any rotation, shift or scaling changes this matrix and the result is displayed immediately. Transformations can be made either with the menu sliders, or directly through the mouse or keyboard. The information on the current location of the object (*e.g.*

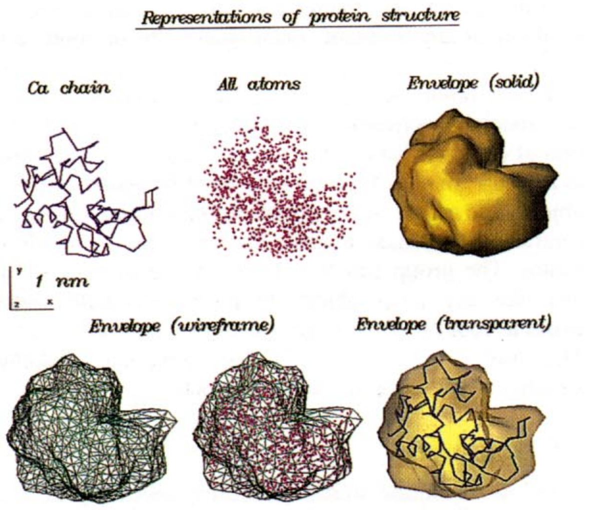
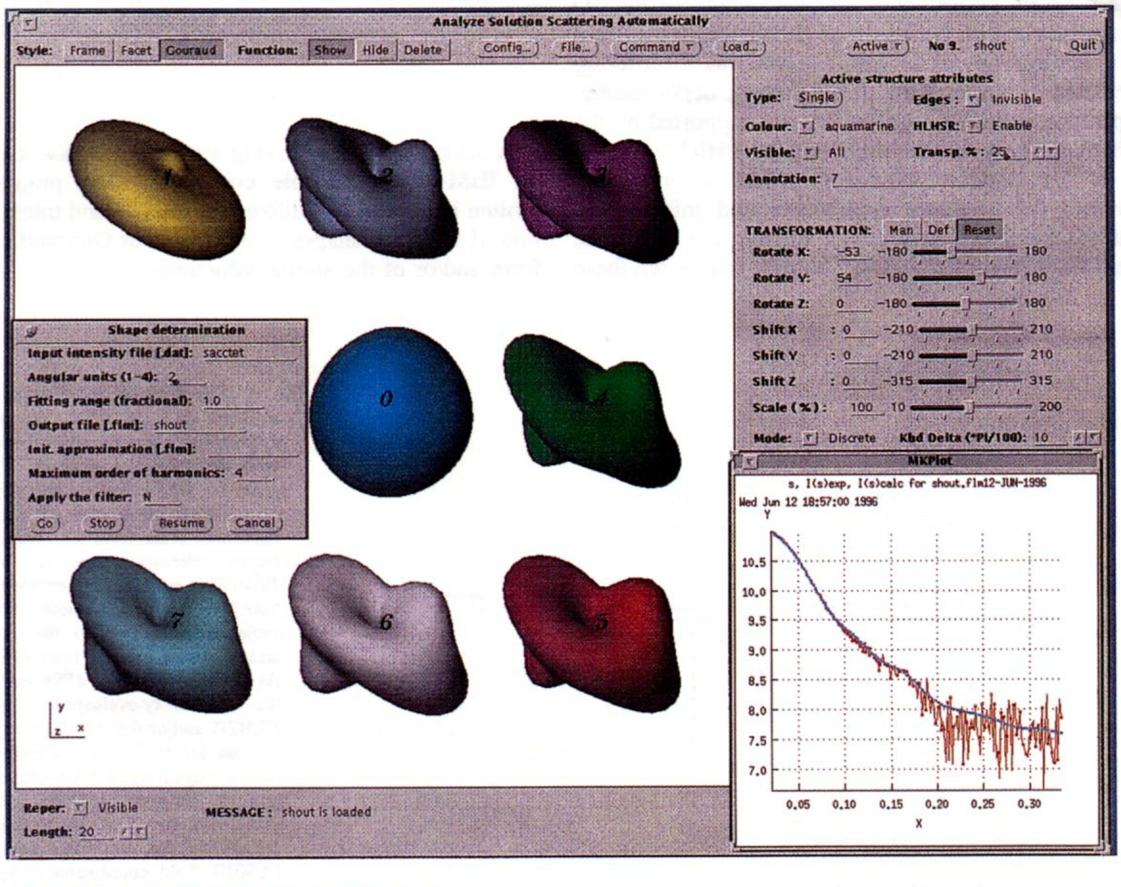


Fig. 3. Real-time shape determination by *ASSA*. The initial parameters are specified and the process is controlled with the help of the command window entitled 'Shape determination'. The spherical initial approximation (shape no. 0) is represented in the middle of the screen. Seven successful iterations have been made and the current fit to the experimental intensity curve is displayed in the window entitled 'MKPlot'.


Analyze Solution Scattering Automatically
Style: Frame Facet Gouraud
Function: Show Hide Delete
Config... File... Command Load...
Active
Quit
Man Def Reset
1
2
3
0
4
7
6
5
y
z x
Go Stop Resume Cancel
MKPlot
Y
10.5
10.0
9.5
9.0
8.5
8.0
7.5
7.0
0.05 0.10 0.15 0.20 0.25 0.30
X

No 9. shout

Active structure attributes

Type: Single
Edges : Invisible
Colour: aquamarine
HLHSR: Enable
Visible: All
Transp. % : 25
Annotation: 7

TRANSFORMATION:
Rotate X: -53 -180 180
Rotate Y: 54 -180 180
Rotate Z: 0 -180 180
Shift X : 0 -210 210
Shift Y : 0 -210 210
Shift Z : 0 -315 315
Scale (%) : 100 10 200
Mode: Discrete
Kbd Delta (*PI/100): 10

Shape determination
Input intensity file [.dat]: sacctet
Angular units (1-4): 2
Fitting range (fractional): 1.0
Output file [.flm]: shout
Init. approximation [.flm]:
Maximum order of harmonics: 4
Apply the filter: N

s, I(s)exp, I(s)calc for shout.flm12-JUN-1996
Wed Jun 12 18:57:00 1996

Reper: Visible
Length: 20
MESSAGE : shout is loaded

Fig. 2. Types of structure representation available in *ASSA*.

coordinates of its centre and its rotation angles) is available at any moment. Each object can be reset to its initial position.

The hierarchical structure of PHIGS objects proved to be especially convenient for rendering and transformation of complex 3D configurations. Objects can be joined in a group which is then considered to be another (group) object. The centre of the group can be chosen among the centres of the joined objects, their average or the global centre. The group can then be transformed as a whole, just like any other object. In particular, there always exists a 'root object' containing all the displayed objects. This tool is very useful for displaying and analysing structures consisting of several objects.

3.3. *Transparency*

In low-resolution studies it is often necessary to 'look through' the surface, for example, when studying complex structures consisting of overlapping objects. This may be achieved by hiding the appropriate envelope or by displaying it in wireframe but the most effective way is to make it transparent as illustrated in Fig. 2.

3.4. *Configuration*

The program performance and the quality of the 3D rendering depend on the hardware accelerations available on the workstation. *ASSA* features (such as Z-buffering for removal of hidden parts of the picture, depth cueing, transparency, anti-aliasing) that are not supported by the workstation hardware are simulated by PHIGS in software at the expense of speed. *ASSA* automatically determines the hardware capabilities and informs the user about the form (hardware or software) in which the graphic features are supported. The user can switch these features on or off and reach a compromise between picture quality and the program speed. The configuration window also provides options to define the location of the light sources and their brightness.

3.5. *Intensity curve evaluation*

The scattering-intensity curve can be evaluated for any configuration of up to four different objects. The programs *FLM2IS* or *FOURBODY* which evaluate the scattering intensity are invoked by *ASSA* as child processes. The intensity curve is plotted in a separate window, independently of the main program, making it possible to display several plots simultaneously. Evaluation of the scattering curves from atomic coordinates is at present performed independently using the program *CRYSOL*.

3.6. *Output*

The current graphic image can be saved into a PostScript file. The names and absolute locations of the displayed objects can also be saved into a plain ASCII file. The latter can be edited if necessary and the configuration can be retrieved later, which is convenient for representation of complex multibody structures.

3.7. *PC version*

A simplified 3D rendering version of *ASSA* also runs on IBM-PC compatible computers. The program is written in Fortran and allows the display and transformation of up to ten shapes in wireframe or Gouraud shaded form and/or of the atomic structures.

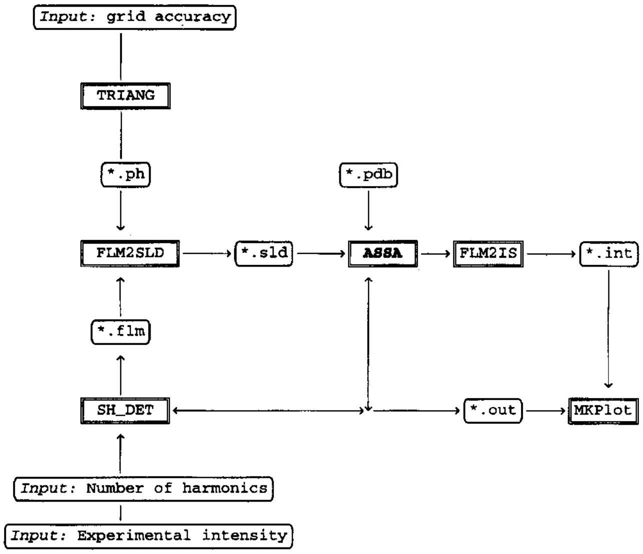


Fig. 4. The flow of information in the software package *ASSA*. The program *TRIANG* prepares the triangulation grid with the requested accuracy. The program *SH_DET* calculates the optimal set of multipole coefficients up to a given maximum order from the given experimental intensity curve. The program *FLM2SLD* evaluates the envelope function from the f_{lm} coefficients. The main program *ASSA* displays the envelopes and/or atomic models from PDB files. At the user's request, *ASSA* can invoke the intensity-evaluation program *FLM2IS* and/or the shape-determination process *SH_DET*. The scattering curves are displayed by the two-dimensional plotting program *MKPlot*. The file extensions are: *.ph, triangulation file (binary); *.flm, multipole coefficients (ASCII); *.sld, coordinates of points on the particle surface (binary); *.pdb, PDB

4. Real-time shape determination

One of the main features of *ASSA* is the possibility of interactively controlling the shape-determination process. This is achieved by coupling the 3D rendering program with the shape-determination program *SH_DET*, described in detail by Svergun, Volkov, Kozin & Stuhrmann (1996) and Svergun *et al.* (1997). The main parameters required by the shape-determination program are the filename containing the experimental data and the maximum harmonic number L which defines the desired resolution. Its output is the particle shape expressed in terms of the multipole coefficients f_{lm}, $|m| < l$, $l \leq L$.

The shape determination is invoked from *ASSA* by a pop-up menu command. The required initial parameters are specified in a special command window (see Fig. 3). The *SH_DET* and *FLM2IS* programs are then started as separate independent child processes communicating with *ASSA* through special pipes. The procedure begins by displaying the initial approximation shape (in the centre of the screen) and the current fit to the experimental data (in a separate window). After each iteration of the shape-determination process, the current best solution is transferred to *ASSA* which displays it as a solid body and requests *FLM2IS* to plot the current fit to the experimental data (saved in the file *.out). A stack of up to eight successive solutions identified by the iteration number is displayed around the initial approximation. The entire procedure is implemented as a series of independent processes and the shape-determination procedure can be suspended at any time, allowing manipulation of the obtained shapes as usual. The process can then be resumed or stopped at the user's request. Fig. 3 illustrates the appearance of the screen during the interactively controlled shape determination.

On a Sun SPARC 20 workstation (60 MHz Super SPARC processor with 35.5 MFlops and 24-bit ZX graphic processor with 352 K 3D TriangleMesh/s), 3D manipulations with Gouraud shaded solids and with the atomic structures are performed in real time. Evaluation of the scattering intensity curve at the multipole resolution $L = 4$ usually used for the shape determination takes about 0.5 s. The procedure of the shape determination at this resolution requires fewer than 50 iterations (about 15 min CPU time).

The interconnections between the program modules are presented in Fig. 4. Further extension of the system by including the modules of experimental data processing and evaluation of the characteristic functions, real-time shape determination for dimeric particles and calculation of the intensities from the atomic models is planned.

The authors are indebted to Dr M. H. J. Koch for his continuing support and valuable comments on the manuscript. The work was supported by the NATO Linkage Grant LG 921231, INTAS Grant No. 93-645 and by the EMBL fellowship to M. Kozin.

References

Bernstein, F. C., Koetzle, T. F., Williams, G. J. B., Meyer, E. F. Jr, Brice, M. D., Rodgers, J. R., Kennard, O., Shimanouchi, T. & Tasumi, M. (1977). *J. Mol. Biol.* **112**, 535–542.

Diamond, R. (1974). *J. Mol. Biol.* **82**, 371–391.

Jones, T. A., Zou, J.-Y., Cowan, S. W. & Kjeldgaard, M. (1991). *Acta Cryst.* A**47**, 110–119.

Keller, E. (1989). *J. Appl. Cryst.* **22**, 19–22.

Rodgers, D. (1985). *Procedural Elements for Computer Graphics*. New York: McGraw-Hill.

Stuhrmann, H. B. (1970). *Z. Phys. Chem.* (*Frankfurt am Main*), **72**, 177–184, 185–198.

Svergun, D. I. (1994). *Acta Cryst.* A**50**, 391–402.

Svergun, D. I. (1997). *J. Appl. Cryst.* **30**, 792–797.

Svergun, D. I., Barberato, C. & Koch, M. H. J. (1995). *J. Appl. Cryst.* **28**, 768–773.

Svergun, D. I., Volkov, V. V., Kozin, M. B. & Stuhrmann, H. B. (1996). *Acta Cryst.* A**52**, 419–426.

Svergun, D. I., Volkov, V. V., Kozin, M. B., Stuhrmann, H. B., Barberato, C. & Koch, M. H. J. (1997). *J. Appl. Cryst.* **30**, 798–802.