

How to make X-ray simulation software working on WWW: A simple recipe based on seven years of experience

Sergey Stepanov*

Argonne National Laboratory, GM/CA CAT at Advanced Photon Source,
9700 S, Cass Av. Bldg. 436 Argonne, IL 60439

ABSTRACT

Attaching WWW interfaces to scientific software opens new opportunities to researchers by making their results available to wide scientific community in a way complimentary to publication. We have shown that this task may be much easier than many used to think: the amount of additional code is small, the Common Gateway Interface (CGI) can be written in any language, not necessarily PERL, and the software can be interfaced on any operating system it was originally written and does not have to be ported to UNIX. This paper provides some useful recipes resulted from seven years of author's experience in developing and maintaining highly successful X-ray Web server project. All these solutions are based on free public domain software (Apache, GnuPlot, and InfoZip) and applicable for multiple computer platforms. Some practical examples are provided.

Keywords: X-ray diffraction, X-ray scattering, X-ray optics, X-ray theory, web-based computations

1. INTRODUCTION

Many scientists, at least in the X-ray scattering field, often face the problem that after a research project is finished, they would like to find further applications to scientific software resulted from it. Leaving apart commercial usage, one of the most attractive solutions is attaching a web interface to the software so that the author can see how his software is used and establish links with his users. However, there is a prejudice that it might be technically difficult and time consuming, thus distracting scientist from his primary research.

In 1997, when the author of these lines decided to make his software accessible by WWW he had similar fears. However, the problems occurred to be much simpler than expected and that project started in 1997 resulted in the highly successful X-ray Server [1, 2] served to date more than 85,000 calculations requests. The primary objective of this paper is to share the X-ray Server technology. In the next two Sections the basics of attaching web interfaces to scientific software are discussed with examples drawn from the X-ray Server. Then, in Section 4 the benefits of making the WWW interfaces are demonstrated.

2. WWW INTERFACE TO X0h

Let us consider the technology of web interfacing on the example of the *X0h* program [3] operational on X-ray Server. The *X0h* software calculates dielectric susceptibilities χ_0 and χ_h for crystals and other materials in the X-ray range by applying an interpolation algorithm [4] to atomic scattering amplitudes taken from standard tables [5-7]. The code is written in FORTRAN for MS DOS and the program is called as follows:

```
X0h.exe wave=1.54 code=Silicon i1=1 i2=1 i3=1
```

Here the command line parameter *wave* stands for the wavelength of X-ray radiation, parameter *code* refers to the crystal description record in the *X0h* database, and *i1*, *i2*, *i3* correspond to the Miller indices of the Bragg reflection. In

* sstepanov@anl.gov; phone 1 630 252 0664; fax 1 630 0667; sergey.gmca.aps.anl.gov

the process of execution X0h.exe creates a text-type output file named X0h.tbl containing the results of calculations (the values of χ_0 and χ_h).

The first step in attaching a WWW interface to X0h.exe is installing web server software on the computer where the X0h program resides. Since X0h runs under Microsoft Windows, one can use either the web server supplied with Windows, or the Apache web server [8] which is free multi-platform software. The examples below refer to the Apache server, although they should work with the Microsoft web server as well.

The next step is to create the following HTML page for entering all the data needed by X0h and place it into the web server documents folder:

```
<HTML>
<BODY>
<FORM ACTION="/cgi/X0h_form.exe" METHOD=GET>
<p>X-ray wavelength: <INPUT NAME="wave" size=10 value="1.54">
<p>Crystal name: <SELECT NAME="code">
      <OPTION>Diamond
      <OPTION>GaAs
      <OPTION SELECTED>Silicon
      <OPTION>ZnS
    </SELECT>
<p>Bragg reflex: <INPUT NAME="i1" size=3 value="1">
      <INPUT NAME="i2" size=3 value="1">
      <INPUT NAME="i3" size=3 value="1">
<p><INPUT TYPE=SUBMIT VALUE="Get X0h!">
</FORM>
</BODY>
</HTML>
```

In this example, please pay attention to the names of input fields (wave, code, i1, i2, and i3) and to the value assigned to the parameter ACTION since they will be discussed below. When the above HTML page is opened in a web browser, it displays as follows:

X-ray wavelength:

Crystal name:

Bragg reflex:

The key part of this page are the <INPUT> and <SELECT> commands placed inside the HTML form, i.e. between the <FORM> and </FORM> tags. When a remote user clicks on the Get X0h! button associated with the input type SUBMIT, his web browser submits the content of INPUT and SELECT fields to the web server and requests it to start a program specified in the ACTION field, namely /cgi/X0h_form.exe. This program is an additional layer that has to be incorporated between X0h.exe and the web server in order to implement the Common Gateway Interface (CGI) without modifying X0h. One can write X0h_form in practically any programming language (not necessarily PERL as often believed) because the only two requirements to the language are the abilities to read environment variables and print to standard output. The X0h_form has to be placed into the /cgi folder on the web server because according to the CGI convention, any file requested from /cgi is expected to be an executable program, i.e. it is executed on the server computer instead of being sent to a remote client. Web servers intercept any screen output of CGI programs and forward it to remote web browsers.

After X0h_form starts, it can access the HTML form input by reading the environment variable QUERY_STRING:

```
query=GetEnv("QUERY_STRING")
```

In our example the result will be:

```
query="wave=1.54&code=Silicon&i1=1&i2=1&i3=1"
```

In the next steps X0h_form.exe has to:

1. Parse the query string,
2. Execute X0h.exe with the help of either exec or system functions and pass it the parameters wave, code, i1, i2, and i3 as command line arguments,
3. Open and read the X0h.tbl file,
4. Write the output page in the HTML format to the standard output; -- this page will be directed to remote user.

As one can see, X0h_form is a relatively simple program that can be composed in a few hours. In the case of X-ray server it is written in Fortran. Schematically the whole chain is presented in Fig. 1.

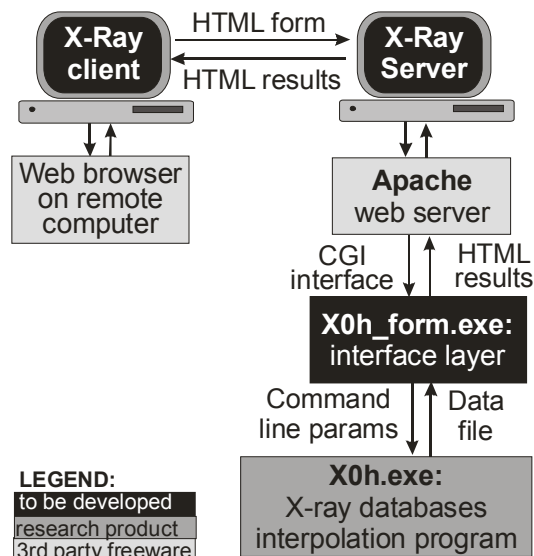


Fig. 1. Providing WWW interface to the X0h program that calculates X-ray scattering amplitudes.

3. WWW INTERFACE TO GID_sl

GID_sl is another program operational on the X-ray Server that simulates dynamical X-ray diffraction from strained crystals, multilayers, and superlattices for arbitrary Bragg diffraction geometries [9]. It is sophisticated scientific software resulted from a series of publications [10-15]. Respectively, the WWW interface to this software requires much more parameters, see Fig.2. Different sections of HTML form in Fig.2 request X-ray wavelength, crystal parameters, surface orientation, geometry of diffraction, scan data, and etc. Besides, the form includes a special scripting language for defining layers for crystals with surface films, multilayers, or strain profiles. However, in spite of all this complexity, the WWW interface to *GID_sl* is not very different from that in the previous example.

When *GID_sl* starts, it expects input data in the form of a file with the name passed as a command line argument:

```
GID_sl.exe filename.inp
```

Therefore, the CGI interface to *GID_sl* has to parse the data from the HTML form on Fig.2 and convert it into filename.inp. By analogy with the previous example we shall call this interface software *GID_form.exe*.

At the end of calculations *GID_sl* produces two files: a data file containing the dependence of X-ray reflection coefficient on a scan angle and a log file of the calculations. Here the tasks of *GID_form* are to output an HTML page presenting the results, to produce a plot of the calculated curve and to pack all the output files into one compressed archive for easy downloading. The latter two tasks are performed by *GID_form* using freeware software GnuPlot [16] and InfoZIP [17], respectively.

The complete scheme of providing WWW interface to *GID_sl* is shown in Fig. 3. The interface structure is similar to that on Fig. 1. Same approach is used with the other five programs at X-ray Server [1]. Those programs calculate X-ray specular reflection from multilayers, X-ray diffuse scattering from interface roughness, X-ray resonant magnetic reflectivity and X-ray multiple Bragg diffraction.

The input data validation and error handling are managed as follows. The interface layers (*X0h_form*, *GID_form*) do not provide any error checking except for the presence of all required keywords in QUERY_STRING. The input is validated at the scientific software level. Then if, for example, *GID_sl* finds an inconsistency in user input, it writes the error message into the file named filename.err and quits. Once *GID_form* finds such a file after executing *GID_sl*, the content of it is printed to the output HTML page.

X-rays: Wavelength(A) / Energy(keV) = Line=Cu-Ka1 Polarization=Sigma

Crystal: Sigma= A W0= Wh=

Bragg Reflection: Substrate da/a=

Geometry specified by:

-- Geometry parameter ([1,7]=incidence angle, [2,8]=exit angle, [6]=Bragg planes angle, [9]=g0/gh):

-- Surface plane ([1-5]): Miscut direction: Miscut angle:

Scan axis: Indices, if other scan axis: Invert scan axis

Scan limits: from to Scan points= Plot argument=

watch progress (single click, please!)

Top layer profile (optional):

```

period=20
t=100 code=GaAs sigma=2
t=70 code=AlAs sigma=2 da/a=a
end period
    
```

(same "Submit" action as above; single click, please!)

Available codes:

[?] Crystals:

AlAs

AlFe3

[?] Non-crystals:

Al2O3

B4C

[?] Elements:

Ac

Ag

Fig. 2. HTML form for WWW-accessed calculations of X-ray Bragg diffraction curves.

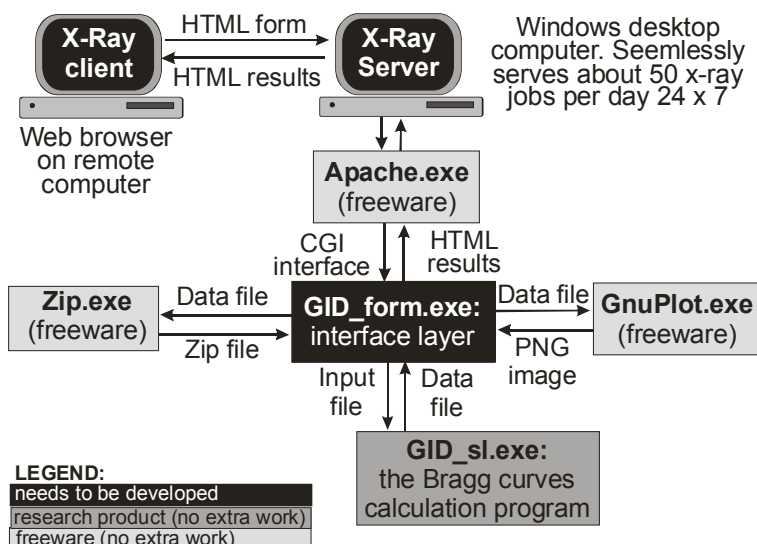


Fig. 3. Illustration of adding WWW interface to GID_sl program calculating Bragg diffraction curves. The only pieces of code to be developed are the HTML form and the CGI data conversion layer GID_form.exe.

4. DISCUSSION AND CONCLUSIONS

As one can see the technology of making scientific software accessible by the World Wide Web is relatively simple. On the other hand, the benefits of interfacing scientific software are hard to overestimate. They are clearly proven by author's experience in authoring the X-ray Server that uses the CGI technology for providing access to X-ray diffraction and scattering software. Fig.4 illustrates some statistics of using X-ray Server by various groups. The server attracts a wide network of users that geographically covers the whole globe. It helped the author to establish new collaborations, to greatly refine and expand the scientific software and find new ideas for advancing X-ray scattering theory..

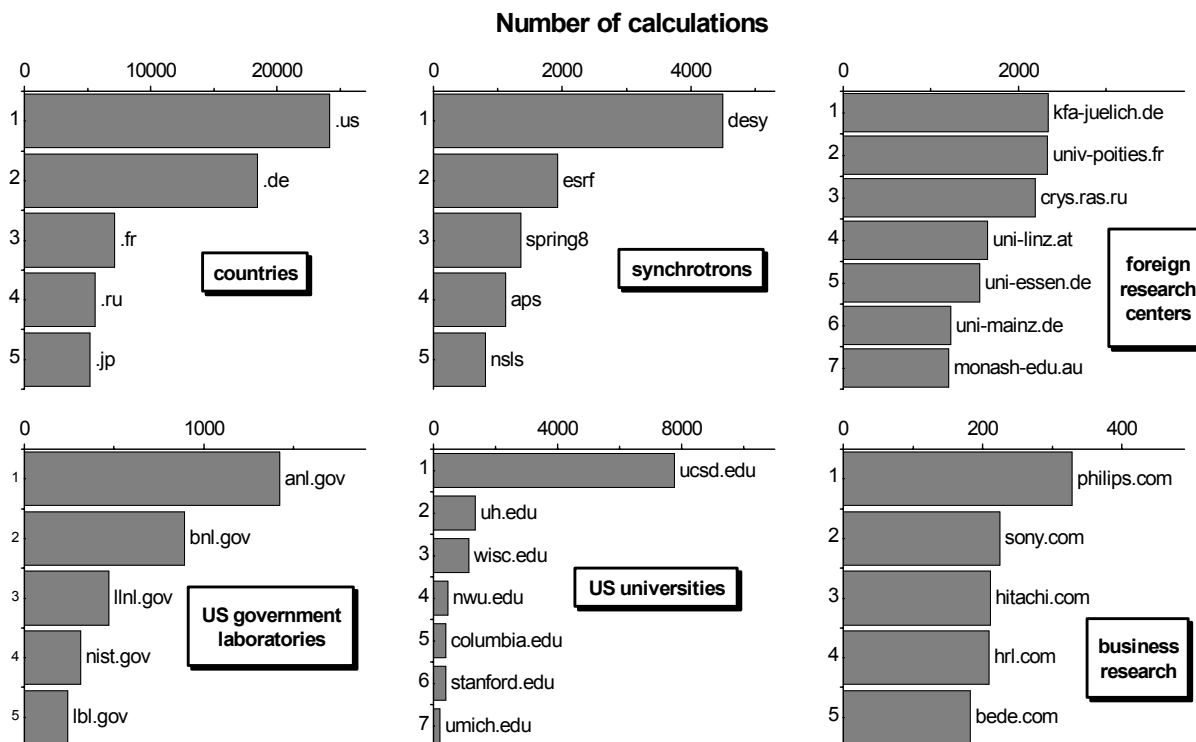


Fig. 4. Some statistics of X-ray Server usage as of May 2, 2004 (total number of calculations requests 81,372).

ACKNOWLEDGEMENTS

The permission of the GM/CA CAT and BIO CAT management to run X-ray Server on their computers is gratefully acknowledged. The GM/CA and BIO Collaborative Access Teams at the Advanced Photon Source are the research centers sponsored by the National Institutes of Health.

REFERENCES

1. X-ray Server current address: <http://sergey.gmca.aps.anl.gov>
2. S.Stepanov "X-ray server: an online resource for simulations of X-ray diffraction and scattering", SPIE 2004 Annual Meeting proceedings; to be published.
3. X0h program home page: <http://sergey.gmca.aps.anl.gov/x0h.html>
4. O.M.Lugovskaya and S.A.Stepanov -- Sov. Phys. Crystallogr., **36** (1991) 478-481.
5. International Tables for X-ray Crystallography, vol.C, Kluwer Acad. Publ., Dordrecht/Boston/London, 1992
6. B.L.Henke, E.M.Gullikson, and J.C.Davis, Atomic Data and Nuclear Data Tables, **54** (1993) 181-342.

7. S.Brennan and P.L.Cowan, Rev. Sci. Instrum., **63** (1992) 850.
8. Apache web server home page: <http://httpd.apache.org>
9. GID_sl program home page: http://sergey.gmca.aps.anl.gov/gid_sl.html
10. S.A.Stepanov, Crystallogr. Reports, **39** (1994) 182-187.
11. S.A.Stepanov, U.Pietsch and G.T.Baumbach, Z. Physik B, **96** (1995) 341-347.
12. S.A.Stepanov and R.Koehler, J. Appl. Phys., **76** (1994) 7809-7815.
13. S.A.Stepanov, E.A.Kondrashkina, M.Schmidbauer, R.Koehler, J.-U.Pfeiffer, T.Jach, and A.Yu.Souvorov, Phys. Rev. B, **54** (1996) 8150-8162.
14. S.A.Stepanov and R.Koehler, J. Phys. D: Applied Physics, **27** (1994) 1923-1928.
15. S.A.Stepanov, E.A.Kondrashkina, R.Koehler, D.V.Novikov, G.Materlik, and S.M.Durbin, Phys. Rev. B, **57**(1998) 4829-4841.
16. Gnuplot package home page: gnuplot.sourceforge.net
17. InfoZIP package home page: www.info-zip.org