<<<Back to Table of Contents

On to Part F: Familiarization with quantitative analysis>>>

Geoscience after IT: Part E

Familiarization with IT Background

T. V. Loudon
British Geological Survey, West Mains Road, Edinburgh EH9 3LA, U.K.
e-mail: v.loudon@bgs.ac.uk

Postprint of article in Computers & Geosciences, 26 (3A) April 2000, pp. A31-A39

Abstract - The geoscientist who wishes to move beyond basic techniques and day-to-day IT applications must know something of the underlying concepts and vocabulary of IT. Communication is vital, linking your desktop to the world. Generic computing tools, widely used in geoscience, can handle documents, geographic information and database management. More basic tools, including long-established programming languages like Fortran, retain an important niche. Recent developments, such as Java and a range of markup languages, bring new flexibility and precision to the geoscience record.

Key Words - Generic software, computer communications, programming languages, markup languages.

1. The need to look at the IT background

Geoscientists may have gone a long way to meeting their employers' immediate needs when they are familiar with the ways of working of a desktop computer and the software required for their projects. To move ahead, however, they must be positioned to meet future demands. This calls for a fuller understanding of some underlying concepts and some more advanced techniques that are now widely used in geoscience.

It is a big step to use a machine to help us organize our knowledge, and we should be aware of the ideas, largely from mathematics, which make this possible. One of the problems, and opportunities, of using computers, is that they manage and manipulate information in a different way from human beings. Some applications mimic earlier technology. Others, like quantitative modeling, are practicable only with a computer.

2. What computers do

Computers count. They can add two numbers together. They can compare two numbers and decide which is the larger. They can carry out simple instructions, such as: store the value of a number. They can store, retrieve and act upon a sequence of simple instructions, such as: obtain two numbers from specified locations, add them together, compare the result with a total calculated earlier, store the larger of the two totals. Because they can do such things, they can perform the full range of mathematical operations that reduce to a sequence of additions, such as subtraction, multiplication, exponentiation, converting to logarithmic or trigonometric functions.

Computers can be connected directly or through the telephone or other network, and data can be passed from one to the other. Their striking characteristic, however, is not the complexity of the underlying ideas, but their extreme simplicity. Their power stems from an ability to perform very large numbers of simple operations quickly, cheaply and accurately. They thus harness the power (and reflect the limitations) of mathematics without the need for laborious manual calculation.

Their importance in geoscience comes from the relevance of numbers and mathematics and from the ability to tie into advances in electronic engineering.

- Text characters can be coded as numbers. By ensuring that the numeric codes follow a widely accepted standard for all computers (the ASCII code), computer text can be exchanged. Because the codes get bigger numerically in alphabetic order, text can be arranged and selected alphabetically. A few more steps lead to the electronic library.
- Points in space can be coded as numbers, using coordinate geometry. They can be combined as complex images, such as photographs or satellite imagery, or as geometric objects in 2, 3 or more dimensions, such as the lines and surfaces depicted on a geological map. A few more steps lead to the computer-based spatial model.
- Numeric and graphical data, like a geochemical analysis or a downhole log, can be recorded, selectively retrieved, analyzed and graphically displayed. As standards are implemented, global integration of data can follow.
- Processes in geoscience can be modeled by mathematical operations represented by computer programs. Together with global data, they can be assembled as a more complete representation of aspects of knowledge.
- Text, imagery, spatial information, data, processes, telephone, video and audio can be linked in a hypermedia representation of the recorded geoscience knowledge base.

Electronic engineers and computer scientists have provided the tools. Progress in their application depends on experts in subject fields, such as geoscience.

3. The computing system

Some knowledge of IT methods and procedures is essential to understand the developing technology which now pervades the geoscience information life cycle. We can start with some general, obvious and basic definitions and concepts. Computing equipment, including processors, memory, disk storage, printers, display units and communication facilities, is sometimes referred to as **hardware**. This is distinct from the **software**, which includes the operating system, compilers and interpreters, and applications **programs**, which specify a sequence of computer operations to meet the needs of the end user. The program is run by the **processor**, which executes, or carries out, the instructions under the control of the **operating system**, making comparisons and performing elementary arithmetic operations as required. The necessary parts of the program, data, and final results are stored in **main memory**. Information that is too bulky to fit in main memory or will be required for a later session is held in **secondary memory**, such as disk storage.

We can for convenience think of the computing system as three subsystems: process, repository and interface. Data items are manipulated by **processes**, which follow a set of instructions supplied in the form of a computer program. The computer is designed to carry out a basic instruction set. This contains instructions for such tasks as moving an item of data from one location to another or performing simple mathematical operations on the data - add A to B, move A to B, compare A and B, and so on. Assembler language, which may be specific to the type of computer, is the means by which systems programmers, helped by a systems programming language such as C, can build up a program as a sequence of basic instructions to carry out a specific task. More complex instructions are written by applications programmers in a **high-level language**, such as Fortran or Basic. This is translated by a compiler or interpreter into the more basic instruction code with which the machine can operate. A **compiler** produces a coded version of the entire program, which can be run efficiently whenever required. An **interpreter** translates and runs the program line by line as it is entered, thus allowing greater flexibility for the programmer at the expense of more work for the computer. An application program is based on an **algorithm** - the set of rules to be followed to solve the problem.

Data, the records of observations and measurements, contain many individual values. Several values referring to different properties of one thing of interest are known as a **record.** Many records from many related items could be regarded as a **database**. At one time, a database was seen as an all-inclusive set of connected records for an organization. Inevitably, however, many distinct collections of data are put together for different purposes, and so we have a collection of databases known as a **repository**. There may be weak connections between the databases that were not realized or not taken into account when the data were collected. The repository might then be termed a **data warehouse** and special "data mining" programs devised to decipher the links between the various datasets. Geoscience data are often recorded as a table, sometimes known as a **flat file**, in which each vertical column refers to a particular property or variable, and each horizontal row contains the values for a specific item or **instance** (see part C, Fig. 1). A set of such tables, in which items are cross-referenced through key fields, and which are structured according to rules which reduce needless repetition of data, form a **relational database**, widely used in geoscience.

Developments such as **hypertext**, in which cross-references are embedded in a document, enable the reader to call up a related reference in the same or in another document by clicking on a highlighted word. **Hypermedia** extends this concept to include references to images (which may have clickable highlighted areas), video, audio, discussion groups and computer processes. Tabular, quantitative datasets and the associated relational database management systems no longer dominate computer information. This greater flexibility is supported by the **object-oriented** approach (H 5, J 2.4). A thing of interest is referred to as an object, which can be a data table, document, image, or any combination of hypermedia. The **object** is a self-contained entity and may include within it the processes or references to processes that are appropriate for manipulating the data it contains. Objects are placed within **classes**, which are structured as a hierarchy, and **inherit** attributes, and relationships to other objects, from classes at a higher level.

The **interfaces**, where parts of the system join, are often of interest. Of particular interest is the user interface, through which the user communicates with the computer and vice versa. For many tasks it is convenient for the user to type in sequences of instructions. Much communication, however, is now through a **graphical user interface** or **GUI**. This uses windows, icons, menus and pointers (**WIMP**). The **windows** are rectangular areas on the screen with a separate process (program) running in each. By **pointing and clicking** with the mouse the window can be moved, resized, hidden behind other windows or made visible by placing it on top, reduced to a small icon, enlarged back to full size, or closed to remove it completely from the screen. The actions of the computer can be initiated by typing instructions in the window, clicking on items in a **menu** (list of options) or on **icons** (small symbols that indicate pictorially what actions will result).

The interface between the repositories, where the objects are stored, and the user environment, in which they are assembled and processed, also deserves some attention. The users' application programs may be linked to the data through an **applications program interface (API)** which is compatible with both. If appropriate standards are followed, finding the required objects can be delegated to an **object request broker (ORB)**, a program, which is part of the **middleware** (L 2) between client and server (E 4) and runs partly on each.

The purpose of the complexity is to enable operating systems to cope with the number and diversity of available sources, while providing the user with the ability to integrate at the desktop the numerous objects of interest from a multitude of sources (**distributed objects**) while retaining ease of use. Underlying the access to distributed objects is the ability of computers to communicate.

4. Communication

Scientists working on the same project have generally tended to be in close proximity, often in the same building. This facilitated discussion and sharing of information. Over these short distances, it is economically feasible to connect computers with high bandwidth coaxial cables or fiber-optic cables, thus giving rapid data transfer. The **local-area network** (**LAN**) built up in this way can be supported by powerful software. A wide range of computers and their operating systems are designed to be compatible with such software, which can support a large network of many hundred devices. A small office with only a handful of users can be networked with simpler systems at lower cost. As the network grows, the task of designing and maintaining systems becomes more complex, and an expert may be required to ensure that it is robust and works consistently.

Local area networks can be linked together through the worldwide network of networks - the **Internet** (D 2). Its **protocols** (the rules, definitions and conventions that govern a cooperating activity) can also be used on a local network, thus ensuring that the in-house network or **intranet** has the same characteristics, and can use the same software, as the Internet. For example, Web browsers designed for global communication can also be used locally. The cost of providing high capacity links over long distances is obviously much greater than that for links within a building. The Internet has been in existence for many years since it began as a research project of the US government. But it is only in the last few years that faster modems, better compression techniques and better software made it practicable to connect home or office computers through telephone lines, fiber-optic cables, microwave and satellite transmission. Telecommunications companies generally provide the physical links. **Internet service providers** (**ISP**) may contract to use some of this transmission capacity, and resell smaller amounts, together with appropriate software and services, to local businesses and individuals.

The emergence of third-generation mobile phone technology is freeing communication from physical connections (International Telecommunications Union, 1999). Broadband **wireless** links are made practicable by **cellular radio**. The area to be covered is divided into smaller patches called cells, each served by a low-power transmitter. The same bands of the radio spectrum can be used in different cells. A computer tracks all subscribers, handing them over from transmitter to transmitter as they cross each cell boundary. The wireless industry is likely to agree global standards in the early years of this century, and the overlap with the computer industry must increase. Geoscientists in the field, and remotely-controlled devices, will be fully linked to the information system.

Where one computer is supplied with information by another, the two computers are known as **client** and **server**. The server may be configured for this specific purpose and may supply several client computers with data and programs on request, possibly over a local area network. The server can be managed by specialist staff within the organization to ensure that secure, up-to-date information is available. A client computer, such as the one on your desk, may also access remote servers across a **wide area network**, to obtain information that is not available locally. The GUI (E 3) can develop into a network user interface. This also has a simple point-and-click procedure to select actions, but the actions are not confined to the local computer and windows can be connected to a remote server. This is achieved by means of a **Uniform Resource Locator** (**URL**), which is a form of address standardized within the Internet. It identifies the servers, of which a central list is maintained, and the file names, which are assigned locally. The URL also has a prefix indicating the protocol in which the contents will be transmitted, and a suffix indicating their format, as described later in this section.

Standards are essential to ensure that the communicated information is meaningful to the recipient. The Internet works because standard protocols (**TCP/IP**) are used throughout. The Internet Protocol (IP) defines the routing between computers. The Transmission Control Protocol (TCP) defines how data are wrapped in packets for IP to transmit. Other protocols, such as **NFS** (Network File System) and **HTTP** (Hypertext Transport Protocol) are compatible parts of the TCP/IP suite. Most modern operating systems provide links to these protocols. A computer can be linked to the Internet, or to an Internet Service Provider, through a **modem**, a device that, by modulating and demodulating the signal, allows computers to communicate over telephone lines. Where available, an **ISDN** link (Integrated Systems Digital Network) may offer higher speed at greater cost.

A local network can be linked to the Internet through a **router**, a computer dedicated to controlling the traffic between the network and the outside world. Security is always a problem with networked equipment, where interlopers prowl in search of passwords, credit card numbers and the like, in the hope of being able to obtain and possibly interfere with information to which they are not entitled. It may therefore be necessary to have password protection on all shared resources on the local network, as well as ensuring that password protection is adequately enforced on all machines connected to the Internet. The router may be connected to a separate computer, which has the task of maintaining security, providing a **firewall** between the local network and the outside world. Each device that can be accessed on the Internet has its own unique identification number (IP address) provided through the ISP or by the Internet Information Center. For most geoscientists, arrangements for networking are handled by the local computer communications manager, who is responsible for organizing and maintaining the local network.

TCP/IP is an example of an **open standard**, agreed by national and international standards organizations such as ANSI and ISO, and available for all manufacturers and suppliers to follow. There are also many ad hoc and **proprietary standards** that have been defined within a company, such as the Windows standards defined by Microsoft. The specifications of some proprietary systems, such as IBM's PC-DOS, have been put in the public domain. A consequence is the availability of compatible personal computers and software from many suppliers.

Personal computers can be self-contained, and if users are concerned only with their own computing, communication may be unnecessary. Even at this level, however, it may be advantageous to download data and programs from a central server rather than storing all that may be required on the local machine. Maintaining an adequate range of material in up-to-date versions can then be the responsibility of the systems manager. Workgroup computing requires a degree of interaction between the participants that demands good communication. Geoscience is a worldwide activity, however, and to take full advantage of the potential benefits, global communication is called for.

Fortunately, the means of communication are available. They take a number of forms (D 2). The most widely used means of communication, accessed by many tens of millions of users, is electronic mail (**e-mail**). The message is generally in straightforward text. The e-mail address of the intended recipient may be hard to find, as there is not always a reliable equivalent of the telephone directory. Large files or those with a more complex format, such as computer graphics or documents with a complicated layout, may be better sent by file transfer protocol (**ftp**). This involves establishing a two-way link before transmission, and the complexities are normally concealed from the user behind a simple **drag-and-drop** operation (using the mouse to move an icon from one point on the screen to another). Shared documents that are being worked on by a collaborating group might use a format suited to workgroup activities, such as MS-Notes. Discussion groups can follow **Usenet** protocols that can be found through Web search engines. Documents for the world at large can be prepared in hypertext markup language (**HTML**) (E 6) and made available through the World Wide Web.

The **World Wide Web** (WWW) consists of many millions of pages stored in standard formats on numerous servers throughout the world. It can be accessed through a Web **browser** - software that runs on desktop client computers, and allows users to make general searches, follow links, and display documents held on the Web. The Web pages are distributed across a wide range of servers and are connected through links that are embedded in the pages. The link appears to the user as a highlighted phrase in a text document or area on an image. Normally concealed from the reader, but embedded in the text at that point, is the address of a point in the Web pages in the form of a URL (Uniform Resource Locator). It looks something like this: <A HREF:="http://www.bgs.ac.uk/bgs/w3/free/reports.html"> *text here* </A>

The first item (**tag**) enclosed within angle brackets indicates the start of an **anchor**, which is the link to another document or to a point in a document. The second set of angle brackets </A> indicates the end of the anchor. On the reader's screen, the text within the anchor is highlighted (usually by printing in a different color) and underlined to indicate that it is a "hot-spot". Placing the cursor within the anchor changes the icon, typically to a pointing finger, and clicking activates the anchor. The HREF attribute contains a parameter within quote marks indicating the transfer protocol (here, http means hypertext transfer protocol), the name of the server (www.bgs.ac.uk), the path and name of the document (/bgs/w3/. . . indicates the directory and the file name) and the format (html means hypertext markup language). Optionally, it can move to a location marked by a flag in the original document. Clicking on the hot-spot causes the specified Web page to be retrieved from the server computer, and displayed on the screen at the flagged point.

The server name indicates the country name (USA if none is specified), preceded by the type of organization, such as com or org for a commercial organization, edu or ac for academic community, gov for government organization, and so on. This is preceded by an abbreviation for the name of the organization (bgs for British Geological Survey) and the name of the computer (here, www is the web server). This "domain name" identifies the specific server and is registered with the **domain name server** (DNS) which links the domain name to its unique IP address.

In addition to retrieving hypertext documents, as has just been described, anchors can point to other places in the same document, or can access images. These are held in other formats such as .gif or .jpeg, rather than .html. This information is included in the anchor and is used by the browser to display the image correctly. Audio (.au) and video clips (.mpeg) can also be accessed from an anchor. The flexibility of this hypermedia system can be increased further by using the anchor to link to a computer program. This can then request information from the user through a simple form, and can perform operations such as searching a database and listing retrieved items on the screen.

Like many facilities accessed from the desktop, the Web contains its own documentation. The facilities it offers are rapidly expanding. Rather than attempting a description here, it is better to explore the documentation of your own installation. An up-to-date account of the range of facilities is available. There are also guides to authors, which describe the many types of tag that appear in angle brackets. They are normally hidden from the viewer but control the appearance and structure of the page. Information can be obtained by following links from the ISP, the Web search engines or Web developers, such as the W3 consortium. Geoscientists can readily find their way to lists of relevant sites on the Web by using a search engine to find entries dealing with their own specialist subject. Alternatively, they can look at the Web pages of organizations such as university departments or geological surveys which provide links to related sources (Ingram, 1997, Butler, 1996). If you are new to the task, a demonstration from a local expert familiar with the system can be very helpful, but in the longer run there is no substitute for experience.

5. Generic software systems

Information comes in various easily-recognized types: text (the ordinary language used in most documents); spatial or graphical information (such as that found in maps and diagrams); structured data (like the tables of data in a database); and information like video or audio records that are less frequently found in this context.

Conventionally, information products have one predominant information type, as in the case of books and serials, maps, data files, video tapes. Major systems of computer software, mentioned in this section, also tend to focus on specific information types. These generic systems are designed to perform operations analogous to familiar actions with conventional products, such as: go to page 52, center the map on this latitude and longitude, select data where a specified variable lies within a given range. The metaphors make the integrated systems easier to use, and they now provide most of the general computing tools for geoscience.

The close links between information types and software systems suggest that they might give a good basis for organizing a course (or a book) on geoscience computing. This structure has not been followed here, partly because of the belief, expanded in J 1.8, that we should break away from these traditional divisions and explore ways to integrate all the information types that have a bearing on an investigation. Links among generic systems are being built into many of the more recent products, easing the task of integration.

Text documents are now generally prepared on a word processor. If they are subsequently published, they will be indexed in numerous computerized library catalogs, but only a few geoscience documents are at present archived as full digital records. For those that are, a markup language or a standard format (E 6, L 3) can ensure that their content can be organized and printed appropriately by computer. **Document management software** is available to manage and retrieve software from a repository of such documents.

Spatial information, which would normally be recorded on maps and cross-sections, can be managed and manipulated on the computer by a geographic information system (**GIS**). The GIS makes it possible to establish, manage, analyze and display a database of cartographic information. Contouring programs can interpolate three-dimensional data and display them as contour maps and cross-sections. Image-editing software can manipulate and adjust other images, such as photographs and satellite imagery. Computer aided design (**CAD**) and scanning software help to capture data and draw maps and diagrams. Visualization programs present datasets graphically, to make it easier to see the relationships between variables.

Structured data benefited from computer methods at an early stage in the development of IT, as they could be handled relatively easily and cost-effectively with long-established programming languages, such as Fortran. The tabular layout is appropriate for much geoscience data as it enables like to be compared with like, and is well-suited to computer analysis. Relational databases fitted this layout well, extending it to keep track of complicated relationships. Relational database management systems (**RDBMS**) provided the means to separate **data management** (input, editing, deleting, updating, selecting, sorting and retrieving) from subsequent analysis and presentation. Statistical analysis and spreadsheet software make it possible to explore the properties and relationships of the data, and other quantitative models throw light on the underlying physical relationships.

Processes or computer programs are generally seen as distinct from the data, so that they can be reused with many datasets, while one dataset can be analyzed by many processes. This separation is not always appropriate, as some data are dependent on a particular process for their interpretation. For example, data points chosen to be representative of surfaces or lines on a map may recreate the original only if a specific process is applied to them. In an object-oriented system (H 5), objects are seen as linked data and processes, both of which, however, should remain reusable in other contexts.

Video and audio records have not been widely used for storing geoscience information. Now that they can be readily linked to hypermedia, however, there is considerable scope for their use in demonstrating, say, the appearance of a rock slice when rotated under crossed nicols, or a picture of a soil profile at the time of excavation. Specialist software is available for compressing these files to reduce their large size for storage or communication.

6. Programming languages

The importance of programming languages to the average user is diminishing. In most applications, user costs greatly outweigh machine costs. Building on the existing software repertoire is preferable to writing new programs from scratch. For most users, the well-established and commercially available generic systems, together with specific application programs, are sufficiently flexible to meet their needs, and it is more economical to buy than to build. Effort in selecting and understanding existing systems may be more rewarding than gaining skills in a programming language. In these circumstances, it is questionable whether it makes sense for a geoscientist to become a proficient programmer. The learning overhead is considerable, and practice is needed to remain fluent.

Most commercial systems deliberately hide the programming code from the user, and the task is to learn the idiosyncrasies of the system and the means of achieving the desired results. Until recently, software systems tended to be compartmentalized, often in a deliberate attempt to prevent the user's escape to rival systems through importing or exporting data. Programming skills made it easier to cross the interface. This is now less of a requirement as it is easier to find an exchange format supported by both systems.

However, good reasons remain for learning a computer language. For the applications programmer, a geoscience training supplemented by programming skills is a powerful combination. In areas like the development of quantitative models, the needs of the individual or the organization may be so specific that only home-made code will do, detailing the programmer's instructions step by step. In other cases, standard software may handle many of the tasks, but programming may be needed for specific additions. There is a large amount of existing code written within organizations or available from colleagues or the literature, for example, Press et al. (1992) and Universal Library (1999). You need programming skills to modify it for the task in hand, or to keep it up to date. Extensive libraries of high-quality subroutines are available for mathematical and statistical analyses, notably in Fortran. They can be included in your own programs. It can also be argued that programming skills provide a deeper understanding of how the computer works and thus of how methods can best be developed in future. A look through journals such as *Computers and Geosciences* (1997) suggests that extending the range of applications calls on an ability to program.

For most users, it is worth knowing something about computer languages in general, as they have much in common. A short course in one language could also give useful background. Languages you are likely to encounter include Fortran, Pascal, Basic, C, C++ and Java. This section offers a very general introduction for the non-programmer.

The languages just mentioned are **procedural**, setting out line by line the sequence of procedures which the computer is instructed to follow, as opposed to stating the objectives and leaving the computer to select the method, as in SQL (mentioned later in this section). They deal with variables and resemble familiar algebraic formulas, such as $x=1/2(y+z)^2$. In **Fortran** one might write X=0.5*(Y+Z)**2. This, however, is not stating an equality. Rather it is indicating that the right-hand side should be calculated, and stored in a variable called X. Perhaps = should be read as "becomes" rather than "equals". The meaning of the Fortran statement could be interpreted as follows: the names X, Y and Z refer to storage locations; if the names have already been used in the program, look up their locations, otherwise assign new locations for them; take the contents of Y and Z, apply the arithmetic operations indicated and store the result in X. The / denotes division, * multiplication, and ** raising to a power. Variables are usually given names which the programmer can remember more easily than X, Y and Z, thus: Distance = Time * Velocity. Data in sequence, as in time series or tables, are conveniently denoted by suffixes in algebra: y_{10} is the tenth measurement of y, y_i is the ith. Similarly, in Fortran, Time(5), Time(I), or Height(I,J) would represent the fifth and Ith measurements in a series called Time, and the entry in the Ith row and Jth column of a table of measurements (an **array**) called Height.

It is often necessary in a program to apply the same type of operation to each member of a series in turn. Rather than writing out each operation individually, it is written once, using **index** variables such as I and J rather than numbers. It is then placed within a **loop** which is an instruction to perform the operation, or set of operations, with stated values of indexes. In Fortran, it might look like:

```
DO  I=1,15
        sequence of statements (operations)
END DO
```

The sequence of operations is performed from the beginning to the end, in this case 15 times. The variable I, which could also be the index of variables in the statements, takes the values 1, 2, 3 . . . 15 in successive loops. To give the necessary flexibility, the programmer can cause control to jump to another point in the program under defined conditions. The command can be **conditional** on a variable having a particular value or a value within a certain range.

```
        IF (Height(I,J) < 500.0) THEN
```

would indicate that control would pass to the next statement if the value of Height(I,J) is less than 500. Otherwise, control passes to a later statement that begins with the word ELSE.

It is thus possible, even without knowing much about a programming language, to get some idea of the calculations by looking at a program. Generally, one statement goes on one line, but & indicates that it continues on the next line. The ; separates short statements on the same line. Comments are generally inserted to explain the program to anyone reading the code. They are introduced by ! and continue to the end of the line. They are ignored by the compiler.

A surprisingly complex set of calculations can be built up from these simple basic building blocks. As the same set of operations can be useful in many different applications, they can be written as a self-contained **subroutine** or **procedure**, which is given a name and a means of indicating the variables on which it is to operate. Thus, Subroutine Sum(X,N) might be written to calculate the total of the first N values of the series called X. The subroutine can be invoked by a statement in another routine, such as Call Sum(Time, Number). Calling the subroutine is equivalent to repeating all the code of the subroutine at that point.

Statements are also required to instruct the computer to acquire data from a particular source, or send it to a particular destination. It might, for example, request the user to enter information from the keyboard, or might read it from a disk, or send output to a screen or printer. The READ and WRITE or PRINT statements in Fortran indicate the variables holding the information, and where the data are to be acquired or delivered.

Fortran is a long-established programming language for scientific use, which has undergone substantial improvements over the years, and is still widely used in geoscience. The huge investment in existing programs and expertise mean that it is likely to remain in use for some time. It is a powerful language capable of representing complex tasks in numerical calculation. It is a reasonably tolerant language, allowing programmers to express the same idea in different ways, some inherited from earlier versions of the language. Programmers can consequently fall into bad habits which make their programs difficult for others (and themselves) to understand and to maintain or modify. For training purposes, therefore, a simpler language such as **Pascal** may be better because it takes a more stringent view of the way the sequences of commands (program **code**) are presented. It thus forces the user to acquire better programming habits. For less complex tasks, **Basic** in its various forms lacks the power of some other languages, but is simpler to learn and to run. Basic is interpreted, rather than compiled like Fortran (E 3), and it is therefore possible to spot mistakes as each statement is written, and the programmer can correct them before proceeding. Visual Basic is widely used to give programming flexibility in a desktop environment.

The WIMP graphical user interface (E 3) is currently the norm for the desktop computer, and is a more recent development than Fortran. The interface is handled at a deeper level in the computer software than the applications which were just mentioned, and special languages such as **Motif** have been written to help the programmer to organize the objects, such as the windows, cursors, icons, and menu-bars, which appear on the screen. More generally, languages such as **C** provide the basic facilities to access the systems functions of the computer. **C++** is its object-oriented counterpart. Programming at this level is a specialized activity. It implies a need to modify or extend the standard functions supplied by commercial systems, which may be as likely to confuse as to help the average user.

A number of other specialized languages deserve a brief mention. The success of the World Wide Web has encouraged some language developments. **Java** is designed to operate within a virtual Java environment. In effect it runs in its own operating system on the desktop client. The server supplies information to the client, including "**applets**" or small applications - processes or programs which operate on the information. The entire object, data and process, is thus supplied from the server. A simple, low-cost client can take full advantage of the server's power. Furthermore, the client can access a wide range of servers worldwide, receiving and combining applets from them all. The drawback is the heavy communications load and inefficiency in the handling of the data. **Perl** is another language which is widely used on the Web, for bringing to life information delivered by the server.

Markup languages place, within a document, symbols which can be read and operated on by appropriate systems. Thus a text report or document can be marked up to identify topics or the various sections, such as title, abstract, chapters, sections, paragraphs, references, or illustrations. The Standard General Markup Language (**SGML**) has been used in this role for some time (Seaman, 1999). The advantages of subdividing a document in this way are considered in D 6. Here, it should be mentioned that **HTML**, the hypertext markup language, is a subset of SGML which is used in many Web documents (E 4), and that **XML** (extensible markup language), has recently been developed as another simpler subset of SGML, with more powerful facilities than HTML. Markup languages can also be used to subdivide three-dimensional graphical objects using **VRML**, the virtual reality markup language.

Postscript is a page description language, describing the layout of text and images on a page, in a form that can be edited or modified. The Postscript files which it generates are widely used in medium to high-quality printing. **Acrobat** offers some of the features of HTML while preserving the page layout in a portable data format (**PDF**) (Kasdorf 1998). **LISP** (LISt Processor) is another language used with text and graphics, where the information is stored as a consecutive sequence (string or list) of characters or of points on a line. It found an important niche in work on machine intelligence, and has been used in cartographic and word processing applications. Structured Query Language (**SQL**) has been widely adopted as a standard interface for querying relational databases (H 3). The advantage of this standard interface is that information can be spread across several databases, each with their own data management systems, and can still be processed by many clients. Communication is made possible by adhering to the SQL standards.

Computer languages can thus be seen as rigorously defined interfaces between the application and the operating system (Fortran, C), the GUI and the operating system (Motif), the client and the server (HTML, Java), the document and the printer (Postscript) and the database and the application (SQL). Numerous other languages, such as APL, Cobol and Ada have played their part in geoscience applications, but introduce no new ideas at this point. Special-purpose languages are available for some software products, enabling the user to modify or customize the products, without compromising the original code.

The computer can follow with speed and accuracy a set of rules expressed as the instructions for executing an algorithm. It lacks the capacity to understand the underlying reasons or to make decisions about unexpected results, tasks at which human beings are much more adept. The systems analyst and user must therefore decide what can better be done by machine and what should remain the task of the scientist. The best features of both can be combined in an interactive system (J 1.6) where the user can keep track of progress and guide the computer in its operations.

7. References

Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., 1992. Numerical Recipes in Fortran – the Art of Scientific Computing, 2nd edn. Cambridge University Press, Cambridge, 963pp.

7.1 Internet references

Butler, J.C., 1996- . Another node on the Internet for those with interests in geosciences, mathematics and computing. http://www.uh.edu/~jbutler/anon/anon.html

Computers & Geosciences, 1997- . Computers & Geosciences Online. http://www.elsevier.nl/locate/compgeosci

Ingram, P., 1997. The Virtual Earth: a tour of the World Wide Web for earth scientists. http://atlas.es.mq.edu.au/users/pingram/v_earth.htm

International Telecommunications Union, 1999. IMT 2000: A vision of global access in the 21st century. http://www.itu.int/imt/

Kasdorf, B., 1998. SGML and PDF - why we need both. The Journal of Electronic Publishing, June 1998, vol 3 (4). http://www.press.umich.edu/jep/03-04/kasdorf.html

Seaman, D., 1999. About Standard Generalized Markup Language (SGML). http://etext.lib.virginia.edu/sgml.html

Universal Library, 1999. Numerical recipes on-line. Hosted by Carnegie Mellon University. http://www.ulib.org/webRoot/Books/Numerical_Recipes/

Disclaimer: The views expressed by the author are not necessarily those of the British Geological Survey or any other organization. I thank those providing examples, but should point out that the mention of proprietary products does not imply a recommendation or endorsement of the product.

<<<Back to Table of Contents

On to Part F: Familiarization with quantitative analysis>>>