An Overview

D G Kanhere,
University of Pune
Pune, 411007, India
dgkanhere@gmail.com

1. Historical view in the Indian context: 40 years of development

I belong to that generation of physicists who began their carreers in early 1970's where there were hardly any major computer centers. I recall IIT Kanpur with IBM 7044 and TIFR with CDC 3600 as the only well equipped computing centers in the country.. That was the era of punched cards and punching machines. Job submission meant carrying boxes of punched cards early in the morning to the computer center. It will take almost the entire day to get the results of the submitted job. The rest of the day one was then free to indulge in non-computational issues, if you wish to. Interestingly, in those days purists use to look down with disdain upon researchers using computers for their calculations - sometimes overtly but most of the times in more subtle ways. Certainly it was not prestigious to do computation and hope to advance the fronyiers of research!

What a change in 40 years!

The change has been revolutionary: In every aspect of scientific endeavor, Computers play a key roll. Computing is now an integral part of the skill sets of every physicist. Almost every research student knows some programming language or other; has rather easy access to moderate computing power, uses internet to remotely access supercomputing if needed and uses rather routinely a variety of software for graphics, visualization, statistical analysis and data mining. A few hundred megaflops of power is routine and a few tera flop is no big deal. Prestigious universities and research institutes look for young researchers with expertise in numerical simulations of this or that kind, want to have groups in high performance computing applications and proudly announce post graduate degree courses or research programmes in computational science. The number of conferences devoted to computational physics / chemistry, numerical simulations, data mining etc are growing and are now indispensable part of academic world. More importantly the boundaries between different disciplines have melted. Computational techniques have cut across the disciplines and have yielded very rich dividends.

2. What has driven this change?

There are three main developments which are responsible for this change.

- **Revolutionary change in computing environment**.

In late seventies, computing environment always meant a central computing center with huge machine requiring a good deal of power and human resource. The available languages and tools for user interface were very limited. There was no versatile operating system such as linux. We did not talk of megaflops! CPU clock speeds were in Kilo Hertz. To give an example, in late seventies diagonalizing 100*100 matrix was considered as a decent work! It all changed with Intel coming up with 8086- 8087chips and quickly within a span of two decades we saw evolution to X86, Pentium ,Xeons, FPUs, attached vector processors (no more popular), RISC architecture and then parallel machines. FORTRAN evolved, standards were set, more versatile and powerful languages were designed, and excellent quality numerical analysis libraries were made available. Very versatile Unix operating systems became standard and user friendly interfaces made its appearance. Now a few hundred megaflops are our desk, a few Giga flops affordable and a few tens of teraflops accessible remotely.

- **Super Computing – High Performance Computing.**

Even with ever increasing computing power available in even desktop PCs, the need for orders of magnitude more power was always felt for a class of problems. This is the era of Grand Challenge Problems giving rise to computing facilities with multiple processors, parallel machines etc. A few tens of thousands of cores with each node having a 10-16 cores is now not uncommon. Available memory (RAM) also grew accordingly and a 128GB Ram per node is now common. Ever decreasing price per Gigaflop was always fuelling the proliferation of computing power everywhere.

- **Towards understanding real life complex world.**

Perhaps the most significant reason for gaining respect for computing as a technique and its acceptance was compulsion for attacking real life problems with accuracies going much beyond toy models. In a number of areas problems demanded higher accuracies of results than what simplified models could provide. Let us consider age old dictum: Physics lies in making meaningful approximations. Almost all real life problems (in physics or elsewhere) are very complex. We built models as realistic as possible. However approximations are dictated by our ability to solve the resulting mathematical formulation. There are only a few models that can be analytically solved. The demands from engineering disciplines, climate modeling, weather forecasting, real time control systems, nuclear and space science left with no choice but to use numerical methods and computer models which are much closer to reality than the highly simplified, exactly analytically solvable models. During the last few decades' material science exploded. Almost continuously, newer materials are discovered, designed with specific targeted properties. These incredibly complex materials ranging from new superconductors to nano materials need detailed microscopic and quantitative predictive tools which invariably mean realistic models and numerical tools. Thus, high performance computing, large data base mining became standard paradigms for physics and indeed for other branches of science.

Today computational science is a matured interdisciplinary area. Almost all leading universities and institutes run post graduate courses in Computational physics, chemistry, bioinformatics and modeling and simulations. It is very common to hear about workshops and conferences in areas such as computational finance, social computing, bio computing applied and Industrial mathematics, computing in statistics and data mining in sciences.

3. What is Computational Science and Computational Physics?

Since it is easy to say what is NOT we begin with that!

It is not computer science; it does not deal with study of compilers and operating systems, languages, hardware etc.. It is not about gates such as AND / OR /NOR / NAND etc.. We do not worry about design and architectural issues in computing and certainly do not address formal questions like what is computing!

We are simple folks and like to use computer as a tool to solve physics problems. We do not get involved into questions like: "How human are today's machines?" "Do these machines have programmable human intelligence?" We simply want these tools to solve our complex problems faster and with greater accuracy. We do firmly believe that if the answer turns out to be wrong, we are to blame ourselves only.

Let us look at Wikipedia View: (Google it, as they say!)

Computational science

Computational science is an interdisciplinary field in which realistic mathematical models combined with scientific computing methods are used to study, usually through computer simulation and modeling, systems of real-world scientific or societal interest.

While Computing means

Computing is any goal-oriented activity requiring, benefiting from, or creating algorithmic processes - e.g. through computers. Computing includes designing, developing and building hardware and software systems; processing, structuring, and managing various kinds of information; doing scientific research on and with computers; making computer systems behave intelligently; and creating and using communications …. (Wikipedia).

Thus computing happens to be more general activity. Further, the field is different from theory and laboratory experiment which are the traditional forms of science and engineering. The scientific computing approach is to gain understanding, mainly through the analysis of mathematical models implemented on computers.

Computational science and engineering (CSE) is a relatively new discipline that deals with the development and application of computational models and simulations, often coupled with high-performance computing, to solve complex physical problems arising in engineering analysis and design (computational engineering) as well as natural phenomena (computational science). CSE has been described as **the "third mode of discovery"** (the other nodes being theory and experimentation). In many fields, computer simulation is integral and therefore essential to business and research. Computer simulation provides the capability to enter fields that are either inaccessible to traditional experimentation or where carrying out traditional empirical inquiries are prohibitively expensive. CSE, let us emphasize, should neither be confused with pure computer science nor with computer engineering.

A few comments are in order.

Evidently this is a multidisciplinary area. When the problems are complex, use of computational techniques is warranted. Using computers does not mean abandoning rigor. You may notice simulations and numerical analysis being used interchangeably. In fact both are very powerful techniques and at time the boundary is blurred. Simulation techniques allow us to carry out "experiments" when carrying these in fields is dangerous or too expensive. One of the most important aspects is its predictive power. Thus computational science has following ingredients.

1. Existence of a complex problem not having analytical solution
2. A mathematical model based on underlying theory
3. Reduction of this model to computationally tractable form. This may need some level of approximation
4. Development of algorithm(s) for obtaining numerical solution
5. Translation of above to working code using suitable language.
6. Validation and testing of the code
7. Simulating the model (in step 2) for further analysis.

What do we plan to achieve in this series?

Evidently Computational physics has evolved into a powerful discipline, inevitable part of the skill sets that must be acquired by any working physicist (indeed by any working scientist). It is multidisciplinary field. One has to master a number of facets such as Numerical analysis, algorithms, computing language (may be a few scripting ones), learn parallel computing - may be GPU programming. Some may focus on large data base handling, while others on tools like Mathematica for carrying out analytical computations. The list is endless and quite clearly a young students needs to pick up and master a few of these as a part of PhD/ Post doc training. At the same time it must be emphasized that he or she is expected to be good at, in fact master of the domain area. Given the complexity and the variety it is no wonder that many have and will turn out to be specialists in some technique of other, perhaps leaving the basic physics aspect secondary! So be it!

There is another reason for this series. With proliferation of a variety of easily accessible, high quality, and user friendly (What I call click- click codes) codes , young (and even old) researchers tend to use these as Black boxes, at times not paying careful attention to and not spending enough time on studying the underlying numerical approximations, formalism and methodology. A general consequence is likely hood of loosing desire to develop own codes / algorithms / methods. Although we do not claim that our articles will fill this gap, we hoped to introduce standard techniques and hope to present the basic algorithms.

We plan to bring out article on (But not limited to),

- **Understanding Numerics**

  A discussion of errors: Integer and floating point representation in fixed bytes, Errors due to finite precision, numerical approximations, statistical sampling etc.

- **Understanding Algorithms**

General algorithms: Simple algorithms well known as warming up exercises for analysis of algorithms, developing good habits for code writing – clean and readable codes, efficiency considerations, reliability and testing, importance of documentation etc.

- **General Techniques**

  Handling and storage of spare matrices, obtaining extreme few eigen values by Lanczoz method and variants.

  Optimization methods and applications

  Conjugate gradient methods

  Fast Fourier transforms, wavelet transforms

  The power of Visualization tools

- **Determinist methods**
- **Stochastic methods**
- **Model Hamiltonians such as Heisenberg and Hubbard models**
- **Non Linear dynamics**
- **Cellular automata**
- **Simulation and modeling of some real life systems such as transport dynamics, pray-predator models, foraging behavior etc..**
- **Bio Computing**

And hopefully many more

Quite clearly the above list is just representative as the field is vast. We cannot do justice to even these topics unless experts in the field participate actively. We invite computational scientist to contribute by sending articles. Finally we end this article by quoting well known, time tested and irrefutable quote.

Garbage in Garbage out

****