

(Courtesy L. Chen)

Jacques-Louis Lions (1928 – ), taught at Nancy and Paris; currently professor at the *College de France* and the *Ecole Polytechnique*, director of *LABORIA (Laboratoire de Recherche en Informatique)*, secretary-general of IMU (International Mathematical Union); contributed to numerical analysis and its applications to control theory and mathematical physics.

## HISTORY OF NUMERICAL ANALYSIS

J.L. Lions

I will start with three or four preliminary remarks; then I will give a plan of my lecture. If some time is left, I will give my lecture. The first remark is that in the history of humanity we are told that there are two periods: BC and AC, and by this we mean "before computers" and "after computers".

Now the second preliminary remark. When I was shown this title, I did not know that Professor Dieudonné was going to be in the audience; otherwise, I would have chosen another title. As he has shown you in the first lecture, he knows the history of mathematics, which is not so in my case. So this was a bad choice. What I really intend to speak about is really the chronology of the main ideas of numerical analysis, say from 1910 until 1985. So this is a special way of doing history.

Now the last remark before I give the plan of the lecture is the fact that numerical analysis in itself is not a really interesting subject. But I think it is fascinating as it is connected with humanity. Firstly, it is connected with technology. In order to do numerical analysis, you essentially need a machine. This was true even long before the computers. And secondly, it is connected with a large number of chapters of mathematics; in particular, to ordinary differential equations, partial differential equations, functional analysis and a little number theory. Probably other things in algebra and combinatorics also. This is why numerical analysis did not appear as a title in the classification of Professor Dieudonné. It is connected with things which appear in Professor Dieudonné's picture.

The plan of the lecture will be like this:

1. 1910 — 1940.
2. Analog computers.
3. World War II.
4. 1950 — 75. (I) Connections with physics.
5. 1950 — 75. (II) Connections with management.
6. Where do we stand in 1975?
7. Parallel computation.
8. Software.
9. 1980 — 85.

In (1), there are two ideas, and in (6), there is something of an "energy crisis" which I will try to explain. And (7) is the result of this energy crisis. I plug in (8) to make some French propaganda and there I will take out my French flag.

We start with 1910 — 1940, and there are two interesting things which happened during that time and which are connected with numerical analysis. I assume you know what numerical analysis is about. One has to compute things and to compute things connected with a model — a model of the real world, physical, chemical, etc. So in 1910, there appeared, I think, the first paper in modern numerical analysis. And it was the paper of a man named Richardson, an Englishman working in meteorology. In 1910, he published a paper which is wrong, and so Richardson is

famous for his completely wrong paper which is, I think, an interesting counter-example. It is known as the Richardson counter-example.

Consider the heat equation for an infinite bar:

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = 0,$$

where  $u = u(x, t)$  denotes the temperature of the bar at point  $x$  and time  $t$ . In numerical analysis, we want to get approximations of the differences of  $u$ . Essentially, we discretise everything: we go back to the definition of the derivative and replace the derivative by the quotient difference. So we introduce a network, say  $h, 2h, 3h$ , etc. on the  $x$ -axis, and  $\Delta t, 2\Delta t$ , etc. on the  $t$ -axis (Fig. 1).

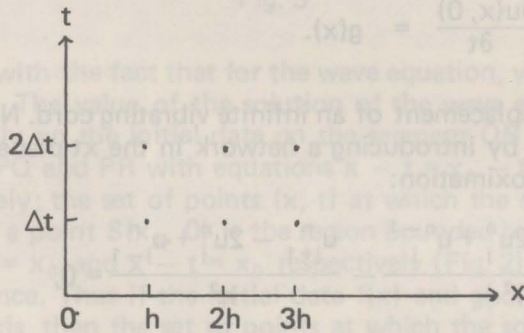


Fig. 1

Now we replace the partial derivatives by differences. Let us denote  $u(jh, n\Delta t)$  by  $u_j^n$ , where the lower index refers to space and the other index to time. Replace the second order partial derivative by  $(u_{j+1}^n - 2u_j^n + u_{j-1}^n)/h^2$ . In replacing the derivative with respect to time, we have a choice. It is the beginning of a story. Now Richardson had a very clever idea. He chose to make this symmetric by taking the approximation  $(u_j^{n+1} - u_j^{n-1})/2\Delta t$ , and then he was in trouble because now we know these approximations do not converge. This can be proved in a very simple way. Starting with a given temperature at time 0, you can compute the temperature at time 1. But if you are given the temperature at time 1, you cannot find out what happens before. It is irreversible. The backward problem is unstable. But the difference equation

$$\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} - \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{h^2} = 0$$

is completely symmetric, and if the approximation was convergent, it would give you something reversible, which is impossible. Of course, a more precise proof can be given.

So in 1910, what does it show? There are difficulties: you cannot just replace the derivatives by something which you think is reasonable. You have to be careful, you cannot make an approximation in a completely random way.

Then the next very important thing happened in 1928. This is not because that was the year in which I was born, but because there appeared a paper by Courant, Friedrichs and Lewy. They discovered something absolutely fundamental and simple. They considered the wave equation in two variables with given initial data:

$$\frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2} = 0,$$

$$u(x, 0) = f(x),$$

$$\frac{\partial u(x, 0)}{\partial t} = g(x).$$

This describes the displacement of an infinite vibrating cord. Now again we play the same game as before by introducing a network in the  $x$ - $t$ -plane and using the following most natural approximation:

$$\frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{(\Delta t)^2} - \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{h^2} = 0,$$

subject to  $u_j^0 = f(jh)$ . If we let  $h$  and  $\Delta t$  tend to zero, the approximation should give the solution. Although it was trivially simple, they were the first to observe that if you do not impose a condition on  $h$  and  $\Delta t$ , this is not going to work, and this is extremely elementary.

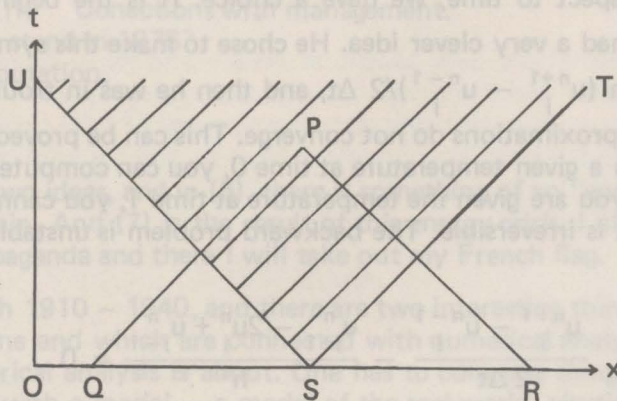


Fig. 2

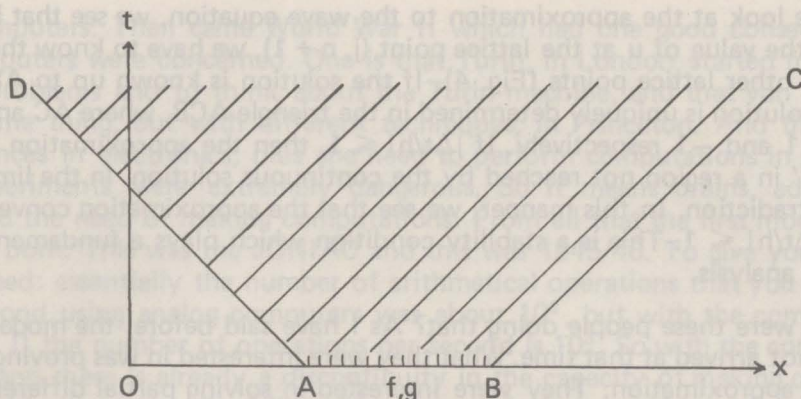


Fig. 3

It is connected with the fact that for the wave equation, you have the so-called domain of influence. The value of the solution of the wave equation at the point  $P(x_0, t_0)$  depends only on the initial data on the segment QR (Fig. 2) of the x-axis cut out by the lines PQ and PR with equations  $x - t = x_0 - t_0$  and  $x + t = x_0 + t_0$  respectively. Conversely; the set of points  $(x, t)$  at which the solution is influenced by the initial data at a point  $S(x_0, 0)$  is the region bounded by the lines SU and ST with equations  $x + t = x_0$  and  $x - t = x_0$  respectively (Fig. 2). This region is called the domain of influence. Thus if the initial data  $f(x)$  and  $g(x)$  are given on the segment AB of the x-axis, then the set of points at which the solution is affected by these given values of  $f$  and  $g$  is the region bounded by the lines AB, AD and BC where AD and BC have gradients  $-1$  and  $1$  respectively (Fig. 3). In a sense,  $f$  and  $g$  propagate a wave.

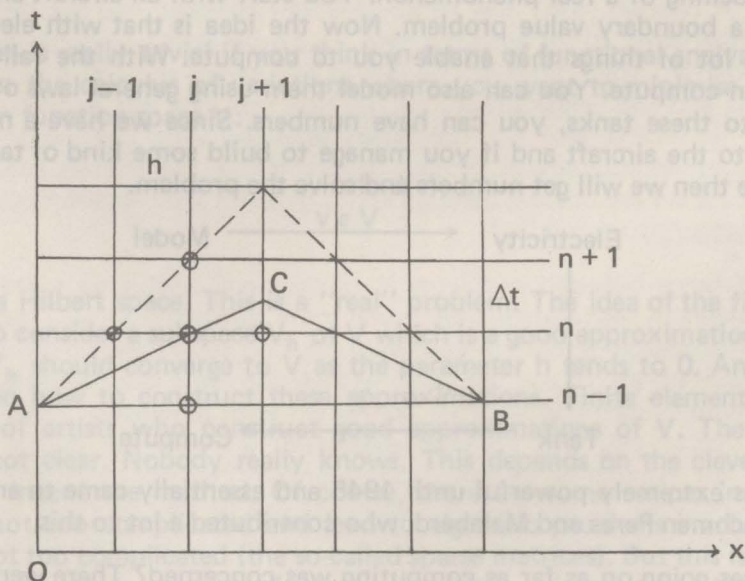


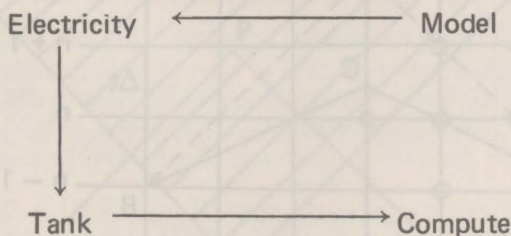
Fig. 4

If we look at the approximation to the wave equation, we see that in order to compute the value of  $u$  at the lattice point  $(j, n + 1)$ , we have to know the values of  $u$  at four other lattice points (Fig. 4). If the solution is known up to AB (Fig. 4), then the solution is uniquely determined in the triangle ACB, where AC and BC have gradients 1 and  $-1$  respectively. If  $|\Delta t/h| \leq 1$ , then the approximation provides a "solution" in a region not reached by the continuous solution. In the limit, we will get a contradiction. In this manner, we see that the approximation converges if and only if  $|\Delta t/h| \leq 1$ . This is a stability condition which plays a fundamental role in numerical analysis.

Why were these people doing that? As I have said before, the modern computers had not arrived at that time. What they were interested in was proving theorems using the approximation. They were interested in solving partial differential equations by approximation. After all if you do not know how to solve them, the natural idea is to replace these complicated looking equations by something that looks more elementary. So the idea was: let us consider these approximations, solve these and pass to the limit, and in this manner we will obtain a solution of the equation that we want to solve.

That was before 1940. So except for one particular case in meteorology, mathematicians were not interested in getting numbers. But there were the engineers and the engineers needed the numbers. For instance, the first computer work was done in the aircraft industry. It was done with analog computers. The details have nothing to do whatsoever with numerical analysis or mathematics, and I must confess that I do not know the details.

The general idea of analog computers is quite interesting. It uses double modelling. What you do is this. You have a problem in the aircraft industry. Suppose you end up with solving some kind of boundary value problem, say Laplace's equation. This is a modelling of a real phenomenon. You start with an aircraft and you have, as a model, a boundary value problem. Now the idea is that with electricity you can realise a lot of things that enable you to compute. With the railway electric tank, you can compute. You can also model them using general laws of electricity, and thanks to these tanks, you can have numbers. Since we have a model which corresponds to the aircraft and if you manage to build some kind of tank with this sort of device then we will get numbers and solve the problem.



This idea was extremely powerful until 1945 and essentially came to an end. There are two Frenchmen Peres and Malabard, who contributed a lot to this.

What was going on as far as computing was concerned? There were essentially two plans: one in trying to solve equations using numerical analysis and the other in

analog computers. Then came World War II which had one good consequence as far as computers were concerned. One is that Turin, in London, started to work on practical problems and to think about the Turin machine, and that von Neumann did the same thing, but with different techniques, in Princeton. And then, there were advances in electronics, plus the need to perform computations in situations where experiments were extremely dangerous. So it means brains, advances in physics and the need of making computations. From all that the first modern computer was born. This was the JONIAc and this was 1945/46. To give you a rough idea of speed: essentially the number of arithmetical operations that you can make in one second using analog computers was about  $10^2$ , but with the computers of World War II the number of operations per second is  $10^3$ . So with the computer of von Neumann there is already a discontinuity in the capacity of making operations and also with respect to flexibility. You can apply it to any kind of model. With analog computers, you have a model and then you have to build an appropriate tank; it is not "multi-purpose". So there are really two discontinuities.

Then excitement started in 1950. Now I am going to explain what is in (4) and (5). Two topics, two ideas, and two motivations. Because of the motivation coming from mathematics (as I have mentioned before, the work of Richardson, Courant, Friedrichs and Lewy) there is available a method which could be applied immediately to the computers. This led to the application of computers to the numerical approximation of partial differential equations, and started a fantastic industry which is far from being complete. The main idea here comes from essentially two or three sources. There is a limited number of ideas introduced but an infinite number of applications of these ideas. The main idea is the so-called "finite element method". Essentially, it was invented by an engineer in structural mechanics but then it was realised by a mathematician that it was universal, that it has nothing to do with structural mechanics and that this idea can be applied to all fields.

The idea is really trivial if you think in terms of functional analysis. You have a problem in the calculus of variations where you want to minimise a functional  $J$  over a given function space  $V$ :

$$\inf_{v \in V} J(v),$$

where  $V$  is a Hilbert space. This is a "real" problem. The idea of the finite element method is to consider a subspace  $V_h$  of  $V$  which is a good approximation of  $V$  in the sense that  $V_h$  should converge to  $V$  as the parameter  $h$  tends to 0. And then there are tricks on how to construct these approximations. Finite element experts are some kind of artists who construct good approximations of  $V$ . The meaning of "good" is not clear. Nobody really knows. This depends on the cleverness of the people who invent the method. Of course, there are some recipes in which these spaces are not too complicated and lead to algebraic problems involving matrices which are not too complicated (the so-called sparse matrices). But this is not enough, and that is why there are tons of papers, not without interest, published on these topics, and this is still going on.

Other ideas were introduced by von Neumann himself and the most famous is the method of artificial viscosity which is absolutely fundamental and is used in many situations. Von Neumann was interested in solving problems in which shocks occur with discontinuities. And these create a terrible problem for the computer. People did not know exactly what to do. Von Neumann realised that the best way to treat these shocks was to suppress the shocks by essentially regularising arguments. The artificial viscosity method consists in introducing a regularising factor which suppresses all these discontinuities. But nevertheless it gives just enough of the discontinuity: you replace something like

by something like

There was a fantastic excitement generated by computers among people in physics, astrophysics and engineering sciences. So people in management also want to do something about this. And it is at this time that the idea of linear programming by Dantzig appeared in 1948, which is, by the way, a completely trivial idea. I think he just introduced the right idea at the right time. Others have the same idea; for instance, Aronszajn. Very few people realise Aronszajn had the same idea, but he did not know what to do with the idea. This generated a fantastic excitement in operations research. Then, later on, people in economics showed that they can also invent models and use linear programming. But this was not enough because the models were non-linear and the objective functions were not quadratic. So this led to non-linear programming, integer programming. We started from numerical analysis and here we are moving through combinatorics and certainly number theory (because of integer programming).

The difference between problems in physics and problems in management is that the problems in management are much more difficult and, as a result, they are not solved. The reason is that if you want to find something optimal in management, then you end up, if you use, for instance, dynamic programming introduced by Bellman, with fantastic difficulties connected with dimension. When you model the physical world you end up with essentially a non-linear partial differential equation and the number of variable is three or four, and not more. In management problems the number of independent variables can be 100 or 1000. So we have a fantastic problem connected with management; maybe it has no solution.

In 1975 we have an energy crisis in computers because in 1975 computers were able to do  $10^6$  or even  $n \times 10^6$  operations per second, where  $n = 1, \dots, 5$ , and

people started to realise there is a limit set by the speed of light which allows you to make additions and multiplications. So here we have a limit and we are approaching this limit. We are approaching the limits of natural resources. As we in France think, if we do not have oil and we need it, we must have new ideas. And it is around 1975 that people introduced a completely new idea: parallel computation. We are approaching the limits of what we can do with one computer. No computer will be able to perform, say  $10^{10}$  operations per second. However, why not use  $10^3$  computers at the same time? Now conceptually this is alright but from the practical viewpoint it looks crazy for at least two reasons. Firstly, who can afford to buy  $10^3$  computers? But this is not as ridiculous as it looks because the cost of computers is going down. For \$1000, I think you can make the same amount of computation as it was possible in 1960 on the largest available computer. If you compare the price of computers with that of automobiles, the decrease is fantastic. So it is not ridiculous any more to use a large number of computers, and I will return to that because it means you can now use computers in a cheap way. The second point is what about the software? How to organise such a horrible thing? After all, to use a computer is not trivial. You have to speak to the computer. Now how do you speak to  $10^3$  computers at the same time? So modulo these two points this idea could be good.

Let me give you an example. Suppose you want to compute an approximation of the heat equation

$$\frac{\partial u}{\partial t} - \Delta u = f \text{ in } R^3$$

The idea is to introduce several operators. For example, we have the parallel algorithm

$$\frac{u^{n+\frac{1}{4}} - u^n}{3\Delta t} - \frac{\partial^2 u^{n+\frac{1}{4}}}{\partial x^2_1} = f_1^n \quad \dots (1)$$

$$\frac{u^{n+\frac{2}{4}} - u^{n+\frac{1}{4}}}{3\Delta t} - \frac{\partial^2 u^{n+\frac{2}{4}}}{\partial x^2_2} = f_2^n \quad \dots (2)$$

$$\frac{u^{n+\frac{3}{4}} - u^{n+\frac{2}{4}}}{3\Delta t} - \frac{\partial^2 u^{n+\frac{3}{4}}}{\partial x^2_3} = f_3^n \quad \dots (3)$$

$$u^{n+1} = \frac{1}{3}(u^{n+\frac{1}{4}} + u^{n+\frac{2}{4}} + u^{n+\frac{3}{4}}) \quad \dots (4)$$

I am not writing down the details of the approximations. Now one computer will compute approximation (1), a second will compute (2), a third will compute (3), all in a non-sequential manner. Then I need a fourth computer which is assigned

to do the average in (4). In this way, of course in a more sophisticated way, computers are made to perform  $10^3 \times 10^6$  operations per second. We are close to something new.

Now I open my French flag to tell you two things on where the difficulties are. The difficulties are in numerical analysis (always) and in modelling and also in the software, that is, in the programming languages which enable us to communicate with computers. Along these lines, I would like to indicate two things which are important. One is the modular programme. Not because it has a historical importance but because I think it indicates a good trend. The idea is to write programmes in a common, single and unified way so that people can exchange their programmes and not spend too much time in duplicating each other's programmes. Along this line, I want to indicate that recently in France people in private industry and also State engineers have been able to invent a language which has been accepted by the Department of Defence in the United States. That means this language is going to be an extremely important language all over the world. I mention this not because it is French but because I know the people who work there. And the people who work on these programmes and languages are people who are trained in logic and combinatorics and who have the equivalent of a Ph. D. It is an important fact that mathematics, even pure mathematics, can be connected with software and this was known at least 20 years by von Neumann. The fact that it is connected with an industrial language shows that in this business of computers, intelligence, hard work and research pay. It is better to know some mathematics to do these things.

The history from 1980 to 1985. This is not presumptuous because the analogous thing in pure mathematics would be just ridiculous. In numerical analysis it is different because it is so much connected with technology and with computers. This needs the work of large teams and you do not create such teams overnight. So one can really understand what is going to happen in the next five years. Essentially, what is going to happen is easily described. There will be supercomputers, each making  $10^3 \times 10^6$  operations per second, and for some time these computers will be in the United States only. On the other hand, industry is thinking along the lines of supercomputers and in a completely opposite way along the lines of so-called "microprocessors" which are extremely tiny computers and are very cheap, of the order of US\$100 each. The difficulty is that they come without any software and you have to use several microprocessors at a time in order to get a machine, and this is connected with parallel computations.

Before concluding my lecture, I want to mention that microprocessors are extremely important for two reasons. First, you will have cheap home-made computers with mathematicians and electrical engineers using 20 of these microprocessors. With electrical engineers and mathematical brains, you will have computers which can do a lot of things. Secondly, microprocessors should be taken into account in education. I do not yet know how. There are a lot of people thinking about this. The fact that now you can have these things in schools obviously has some importance for education.