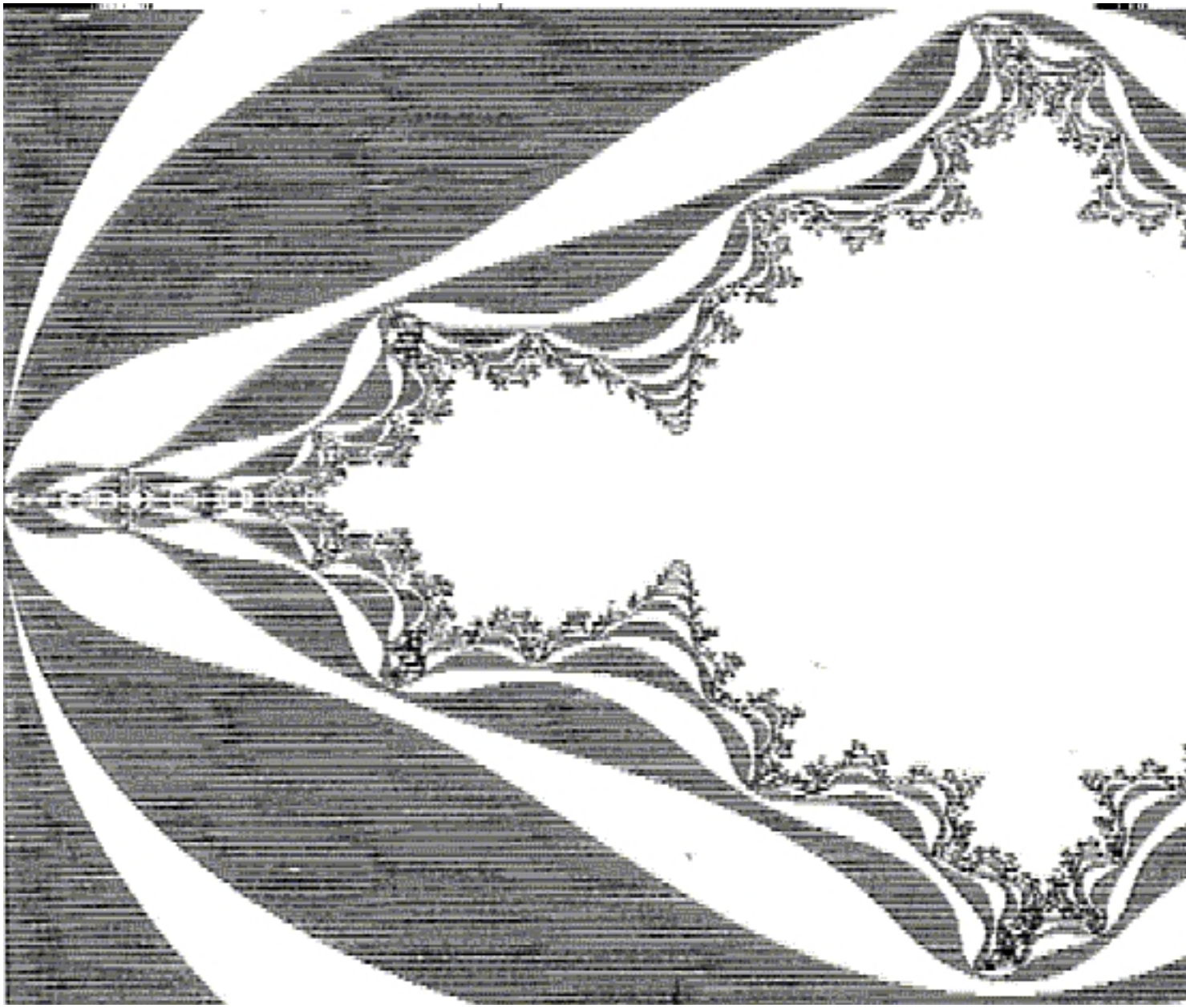


Well I was not long turned 19 years old when I arrived at Imperial College, University of London to study a physics degree. In the first year, three afternoons per week were to be spent in the laboratory on practical experiments. One of the lab subjects was computing, and by good fortune I was in the lab group which did computing first.

The 1'st year physics laboratory computer was a VAX. Connected to this were about 40 or 50 terminals. A handful were real WYSE terminals, the rest were ex-BBC computers (popular

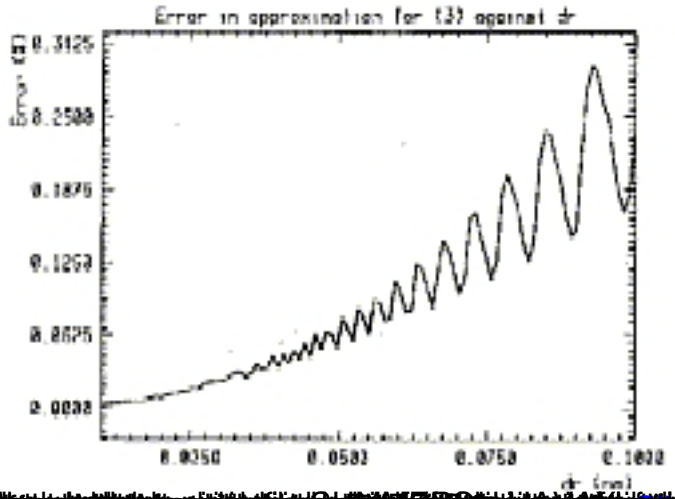
1980's UK home/educational computer); they had had their ROM replaced such that they behaved like terminals.

The computer course aimed to teach us Fortran 77 and we were advised to forget any previous bad habits, such as the BASIC we might have learnt from our 1980's home computers (ZX Spectrum, BBC, Commodore 64 etc). This was supposed to make it easier to learn a "real" programming language like Fortran. I suspected this advice would be unuseful, and was not proved wrong. Why Fortran? Apparently most scientific programming took place in Fortran because of the large scientific libraries available in Fortran. I did not particularly like Fortran though the VAX was the fastest computer I had ever used. I have in any case always felt that a good programmer could make sense of ANY programming language. In the course of my professional life I have used several, finding it easier each time to learn a new one. Some things might be easier or more difficult in different languages but ultimately any program comes down to logic, and if you are good with logic you'll be able to write good programs in almost any language once you've learnt the particular syntax and rules.

Of some concern was the disk space allocation to each student, of just 500 blocks. One VAX "block" is actually 512 bytes, so 500 blocks corresponds to about 250K of disk space. This was plenty for the simple Fortran programs but not so much space for storing the output files of Mandelbrot sets. But then again we weren't supposed to be creating Mandelbrot sets.

The VAX operating system, VMS, I found very nice. I later worked on a legacy VAX application in my first job, where I learnt a lot more VMS. It is easy to use, type "Help" and you can be sure of finding a clear explanation of anything you want to know about the VMS commands. VMS is not case sensitive like UNIX, and you can abbreviate commands as short as you like so long as there is no ambiguity. For example, for a directory listing you can type "directory", "direc", or "dir". But not "di" as other commands also start "di" and VMS obviously can't resolve the ambiguity.

By reading the Help files I learnt a lot of interesting VAX commands. Things like "broadcast", which allowed you to cause a message to appear on another terminal once you knew their port number. Or "phone" where you could chat to one or more other students using the keyboard. Not that there was much utility in messaging someone who sat only a few feet from you but it was fun. Naturally a lot of friends learnt the use of these commands from my example, curious about how the messages on their screens were caused. Messages appearing on the recipient's screen in the middle of their program editing caused some disruption since the


Error in approximation for (3) against dr
Error (%)
0.3125
0.2500
0.1875
0.1250
0.0625
0.0000
0.0250
0.0500
0.0750
0.1000
dr (rad)