10-25-1989

EASIPLOT A Graphics Tool for Ordinary Differential Equations

Stephen J. Sugden
Bond University, ssugden@bond.edu.au

Bernard Duszczyk

EASIPLOT — A Graphics Tool for Ordinary Differential Equations

Steve Sugden and Bernard Duszczyk

School of Information & Computing Sciences
BOND UNIVERSITY

Abstract

Advanced computer hardware available at Bond University made it possible to aim fairly high when planning our graphics CAL program for support of introductory calculus and differential equations subjects. Having laboratories equipped with IBM PS/2–60 machines having VGA graphics (16 colours at 640x480 resolution), some exciting possibilities emerged. The fast pace at which events happen at Bond University has had a great influence on planning, and while many candidates for plotting on MS-DOS machines were considered, none was considered entirely suitable.

EASIPLOT features cartesian, polar, parametric plotting in the (x,y)–plane with the facility to plot more than one function simultaneously. Families of curves can be defined and plotted and the integrals of first-order ordinary differential equations can be graphed.

Keywords

EASIPLOT, VGA, function plotting, graphics, differential equations.

Background

Being involved in the setup of a new university is a rather rare experience—certainly in the case of a *private* university in Australia. For academic staff there is of course the opportunity to contribute to decision-making and planning in the areas of course definition and development and these can be expected to have some very far-reaching consequences. Early and ongoing plans include proposals for computer enhanced learning and given the stated nature and goals of Bond University, such plans are especially relevant.

From the very beginning, we wished to have good-quality computer graphics to support the teaching of elementary coordinate geometry and calculus, and spent some considerable time examining various offerings from textbook publishers and many other sources. Recognizing that the effort required to build such software *ex vacuo* was very substantial indeed, the authors felt that the time spent in assessment of some readily-available products was worthwhile. Perhaps we were too fussy—no package was considered entirely suitable. Our reasons are set out in the remainder of the paper.

Existing software considered

In contrast to using pre-packaged products, in-house development of software has the advantages of being tailored to exact requirements, amenable to extension in almost any direction desired, and no problems with licensing arrangements. The major disadvantage of course is development time and cost. There is always the challenge and excitement of doing something original with the possibility of extending it at some later time, but this enthusiasm must be tempered by the cold realisation that software development is known to be a highly costly, labour-intensive process at the best of times, and also that the marketplace (especially the IBM and compatible world) is already flooded with graphics software of very good quality.

Therefore, to put our project in perspective, some discussion follows on possible choices from existing products for the task of general graph plotting (including parametric forms) and graphical solution of ordinary differential equations.

Perhaps the most exciting new piece of software to emerge in late 1988 was Wolfram Research's *Mathematica*. By early 1989 it had become available for essentially all major hardware from superminis and workstations almost down to PCs—including Macintosh and workstations such as Sun, Apollo, Pyramid, Vax. Significantly however it was not yet available for IBM PC or PS/2 and the price for multiple copies was prohibitive—apparently no site licence was available. At Bond University we have a single copy of this software running on a Macintosh IIX in the School of Information & Computing Sciences. Some very valuable use of it has been made as a research tool by the present authors and others, however its use as a teaching tool is very limited at present.

Mathematica contains intrinsics for conventional procedural programming, functional, and rule-based programming as well as a very complete library of special functions—giving a similar level of coverage as a standard mathematical reference such as Abramowitz and Stegun (1972). It has support for linear algebra, numerical and symbolic differentiation and integration. In fact, *Mathematica* is a goldmine of mathematical resources that the limited space available here could not possibly justify. A number of third-party packages are now available for *Mathematica,* and these of course only enhance the power and scope of system as a whole. For a good reference and introduction, the interested reader is referred to the reference book by the principal author of *Mathematica* (Wolfram, 1988).

In the 1987 CALITE proceedings, (Clayton *et al,* 1987) we find a paper describing the *CAPGRAPH* program developed at Capricornia IAE. This program does a very creditable job of cartesian function plotting and is certainly cheap enough for a site licence. It was considered to be a good product but did not allow parametric plots or differential equations, and had no mouse support. In their paper an extensive list of planned enhancements was given, so perhaps also some of the features that we found lacking are now included.

We heard many good reports about the *Graphic Calculus* collection by David Tall but at the time it was not available for IBM or Macintosh but only for bizarre machines such as the BBC microcomputer.

Many disks of software supplied with standard first-year calculus textbooks were examined, and the overall quality was rather disappointing. Most seemed to be written in a comparatively old dialect of BASIC, which in some cases was interpreted and other cases compiled. While this fact alone does not imply poor quality, it does give some indication of the level of commitment of many authors to an ongoing development of their software in a modern, structured and maintainable language. Items of software considered were those supplied with the books (Finney and Thomas 1988; Hunt 1988; Anton, 1988). Being aimed at a wide market of tertiary students with PCs or compatible, it is understandable that the programs must support the lowest common denominator of MS-DOS graphics—the awful CGA. That they were not able to autodetect and support better installed graphics hardware unfortunately ruled them out as it would have been ludicrous to be running chunky old CGA on VGA colour monitors. The abysmal quality of CGA graphics is quite simply unacceptable in 1989.

An interesting collection of plotting programs was a set purchased from Bridge Software directly from their author Mark Bridger in the USA. Bridger teaches mathematics at Northeastern University, Boston, Massachusetts and has written reviews of at least one version of Turbo Pascal (Bridger, 1986). Bridger's programs were also written in an early version of Turbo Pascal, and were reasonable for plots of $y = f(x)$ and $z = f(x, y)$, and solution of $dy/dx = f(x, y)$. Once again, these were useful, but quite a long way from the features we planned for software to be used by our calculus and quantitative methods students.

Borland's equation solving package *Eureka* was also considered, but its graphing ability, although impressive, was very limited. Site licence was not available.

No MS-DOS software that we considered had mouse support, and this fact alone dates it as a class. We planned from the beginning to provide such support. One of the authors (SS) finds it rather irritating to use some of the Macintosh software where one is forced to use a mouse for almost everything—the keyboard is essentially redundant. Examples include *ANUGRAPH* in which each character/token must be laboriously click-selected and then click-deposited into the function definition string being built. Otherwise, we found *ANUGRAPH* to be an excellent package. Other mindless features which tend to be found on Macintosh plotting programs are scroll-bars for selecting one value from a continuous set of values. For most purposes, we argue that these and related functions can be more easily executed by keystrokes. There is little doubt however when using the computer screen as a window onto the (x,y)–plane for plotting, some kind of pointing device is a great benefit. For example, an idea which we were keen to implement was to allow students to see graphical solutions to ODE's by simply rolling the mouse cursor around on the screen and then clicking on the desired point. Our program would then compute the integral curve though that point and plot the curve. It is argued that such a facility is of great benefit to students who are trying to visualize the nature of families of solutions to a particular differential equation, or the dependence of a single member of the family on a parameter in the differential equation (Sorli, 1988).

Some packages for the Macintosh, e.g. those from the Kinko catalogue (Kinko, 1988) allow plotting of solutions of ordinary differential equations and these were available, but once again, they were stand alone packages which we could not

extend or tailor to our requirements. Further, our aim was to develop something for the IBM PS/2 machines in the student computer laboratories, since these were all VGA colour (16 colour with 640x480 pixels).

The Carnegie-Mellon authoring language cT evolved from the CMU Tutor system (Isaacs, 1987; Sherwood and Sherwood, 1988) and this was used to develop a very early prototype of the differential equation component of EASIPLOT on a Macintosh. cT was rather impressive and at first it was thought that the function plotting program could be developed on a PC using this tool. The PC version of cT—like everything else it seems—was 'almost ready', but this was simply not good enough. We were just not prepared to develop on a Macintosh in the hope that the PC version would soon appear and be totally compatible with the Macintosh version, with possible attendant problems of conversion. It became clear that we had to develop something ourselves directly on a PC or PS/2 machine. In fact the PC version of cT arrived well into the second half of 1989, after being promised for January or February 1989, although it does appear to be highly compatible with the Macintosh version—our simple ODE prototype ran without modification after porting the source code from the Macintosh to a PS/2-70 and recompiling.

In summary, then, there seemed to be sufficient justification to begin the project, given the general unsuitability—for one reason or another—of existing commercial offerings. In addition, some code had already been written for an earlier project, (Sugden and Simmond 1987) and while admittedly in a very crude form, was nevertheless a useful start for the EASIPLOT project.

Requirements for the software

Initially, we thought our requirements for the plotting software to be fairly modest, with the main features being

- ability to plot $y = f(x)$ in the cartesian plane
- ability to plot polar relationships $r = f(\theta)$
- ability to plot parametric relationships represented by $x = f(t), y = g(t)$
- capability of plotting more than one function simultaneously. This included being able to plot a function and its derivative together.
- ability to plot inverse functions (the parametric capability allowed this desideratum to be conveniently catered for)
- ability to plot solutions of first-order differential equations of the form $dy/dx = f(x, y)$

As the project progressed, the inevitable temptation to add extra features became far too strong to resist, so the following were also included

- graphing of straight lines including automatic tabulation of points of intersection and shading of intersection of half-planes specified by the user (in fact equivalent to a feasible set for a linear programming problem)
- extended mouse support for creation of drag-box and zoom after initial plotting
- autodetection of function zeros, stationary points, global extrema, and vertical asymptotes (singularities) during scan of the plot interval
- availability of a scrollable table of function and derivative values to the user—one use of this feature is to search for a change of sign of ordinate and

thus pinpoint a zero of a function
- extension of function definition retrieve / save facility to allow mouse-scrollable selection with essentially full details of the saved case rather than just MS-DOS filename to be read and selected by the user

Technical and implementation issues

For the EASIPLOT project, the development tool used was Turbo Pascal 5.0 and later version 5.5, both of which contain very comprehensive collections of graphics routines. The routines, while not entirely device-independent, approach this ideal reasonably closely. Version 5.5 includes quite a good implementation of *objects*, and this facility may be to advantage later if time permits further development of EASIPLOT. Version 6 of Turbo Pascal promises to have mouse support and hardcopy support for the graphics library, and this product is eagerly awaited by those who have invested much time and effort in graphics applications in Borland's flavour of Pascal.

Turbo Pascal 5.5 still has the limitation of a maximum of 64k bytes of statically allocated data, and while this limit was never reached during the EASIPLOT project, it was clear that planning of the implementation had to proceed with this constraint in mind. For example, the desirability of allowing multiple functions to be plotted simultaneously, including families of curves indexed by a parameter, meant that some form of dynamic memory allocation was required—either by use of heap space / pointers or by writing intermediate results out to disk for later retrieval.

An unexpectedly difficult problem was the achievement of a reasonable axis annotation once given the endpoints for x-axis plot interval by the user. The first step is to determine the length and then the logarithmic range of the interval—these are clearly easy. However, to decide on the number of subdivisions for the interval partition so as to obtain 'round figures' for the axis annotation was not so easy. In the end, a simplified algorithm was chosen which gives reasonable gridpoints most of the time. Time was being consumed which could be better spent on other, more fruitful problems related to EASIPLOT.

Features

Major features of the EASIPLOT package are

- a moving-bar windowing menu system with options accessible by cursor keys, initial character
- list of available functions with clearly-defined domains
- context-sensitive help screens
- extensive mouse support
- parametric families of curves, and parametric relations
- solution of first-order differential equations
- mouse-driven recursive zoom feature
- optional derivative plot and optional coordinate grid
- optional auto / manual scale—manual scaling is necessary to give some kind of reasonable-looking plot when large excursions of the function must be accommodated
- save and retrieval of function and parameter and interval definitions to / from disk file

- allow user to specify number of points at which function values are sampled—complicated functions may take a long time to plot, but can be done much more quickly if say 50 or 100 points are used instead of 1000. If the curve is interesting or more detail required, then the number of points can be increased and the function replotted
- intelligent non-rescan detection, i.e. function values are not recalculated unless a parameter on which they depend is changed, e.g. density of points for a plot
- scrollable tabulation of function values
- autodetection of function zeros, stationary points, global extrema, asymptotes—oblique and vertical (singularities)—within interval of plot
- on-screen help is available for the opening main menu and major submenus, as well as context-sensitive pop-up help for the data entry (function definition) operations. Help is invoked by clicking this option on a menu or by pressing the ubiquitous F1 help key.

Planned enhancements to EASIPLOT

- special treatment of polynomials
- calculation and illustration of limits
- higher order ordinary differential equations
- special treatment of rational functions
- precise definition of function domains to include singularities (this is perceived to be a hard problem since there may be infinitely-many such points not in arithmetic progression or other easily-characterizable sequence—e.g. $f(x) = 1/J_0(x)$, where $J_0(x)$ is the Bessel function of order 0)
- extension to allow multiple, independent functions to be plotted, and also to permit functions to be defined in terms of other, previously-defined functions; this requires dynamic allocation for function value arrays (1000 points x 8-byte reals = 8000 bytes per function)
- improvement of mouse support
- 3-D surfaces in perspective with ability to rotate—this seems ambitious, not so much from the programming point of view, but from the sheer computing power needed to support fine meshes/grids in something approaching real-time; the power of a graphics workstation is really required for this job, or at least a graphics coprocessor
- isometric coordinate transformations to study e.g. conic sections

Experience

The third trimester at Bond University begins in September each year and the EASIPLOT package was considered to be in a sufficiently stable form to allow a preliminary trial use with some student groups after two or three weeks into that semester in 1989. The hope was to gain some feedback on its general usefulness from the students and tutors so that some early findings could be reported at the present ASCILITE conference. At the time of writing, the system has only been is use for some two weeks, however it is hoped to have some more information at the time of the conference, or perhaps for ASCILITE ’90. Feedback on the whole has been encouraging, however some further experience with other student groups would be desirable before any meaningful assessment of the program’s true worth could be made.

Some sample screens

Some sample problems and screens from EASIPLOT are given below.

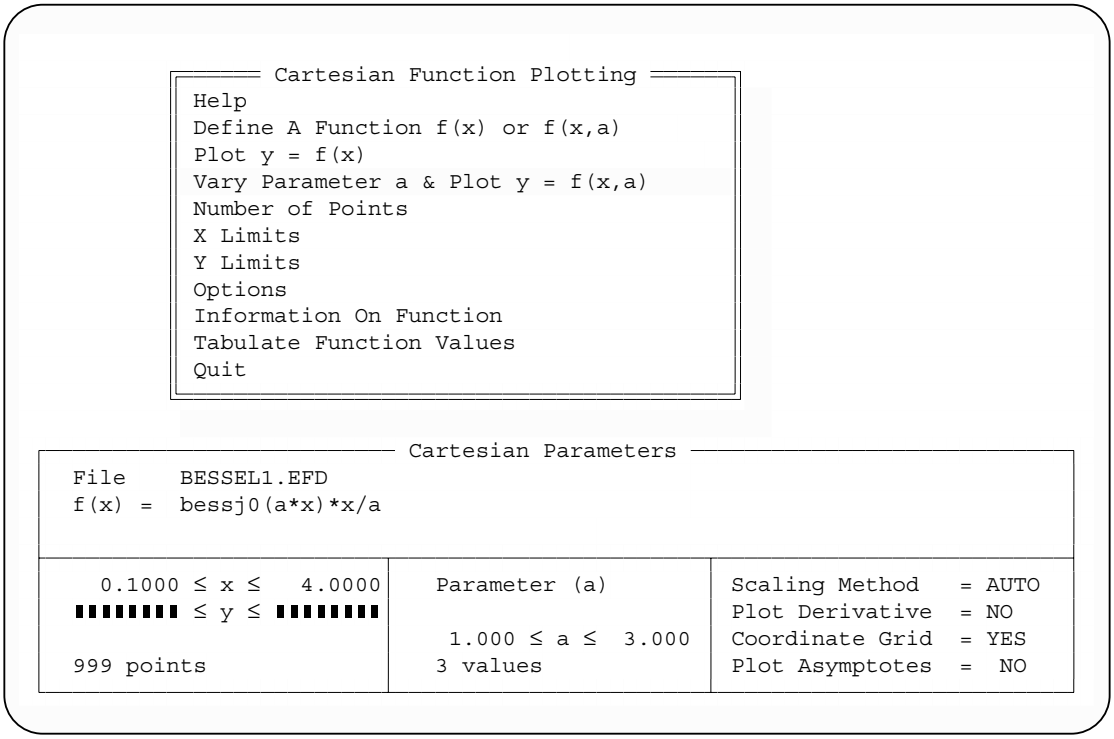

Figure 1 Cartesian plotting menu in EASIPLOT

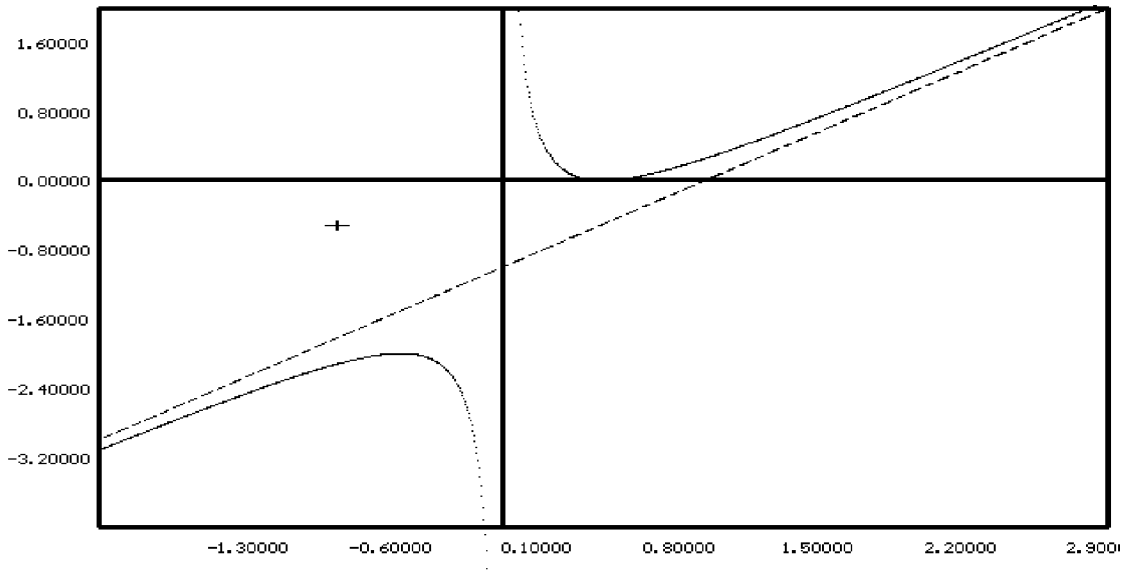

Figure 2 $y = 1/4x + x - 1$ and its oblique asymptote

Name	Description	Domain	Singularity
ABS(x)	Absolute value	all real x	
ARCTAN(x)	Inverse tangent	all real x	
BESSJ0(x)	Bessel function; 1st kind; order 0	all real x	
BESSI0(x)	Modified Bessel; 1st kind; order 0	all real x	
BESSK0(x)	Modified Bessel; 2nd kind; order 0	x > 0	
BESSI1(x)	Modified Bessel; 1st kind; order 1	all real x	
BESSK1(x)	Modified Bessel; 2nd kind; order 1	x > 0	
COS(x)	Trigonometric cosine	all real x	
COSH(x)	Hyperbolic cosine	all real x	
ERF(x)	Error function	all real x	
ERFC(x)	1 - ERF(x)	all real x	
EXP(x)	Exponential; inverse of LN	all real x	
GAMMLN(x)	Logarithmic gamma function	x > 1	
LN(x)	Natural logarithm; inverse of EXP	x > 0	
SIN(x)	Trigonometric sine	all real x	
SINH(x)	Hyberbolic sine	all real x	
SQR(x)	Squaring function; inverse of SQRT	all real x	
SQRT(x)	Square root function; inverse of SQR	x ≥ 0	
TAN(x)	Trigonometric tangent; equals SIN/COS	all real x	(2n-1)π/2
TANH(x)	Hyperbolic tangent; equals SINH/COSH	all real x	

Figure 3 Functions available in EASIPLOT

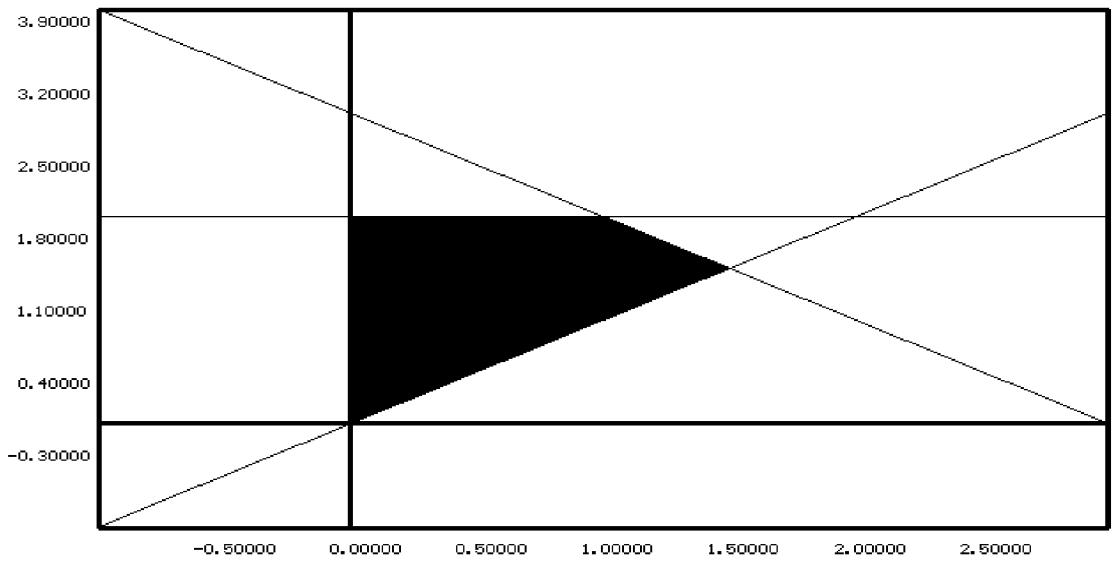

Figure 4 Intersection of half-planes

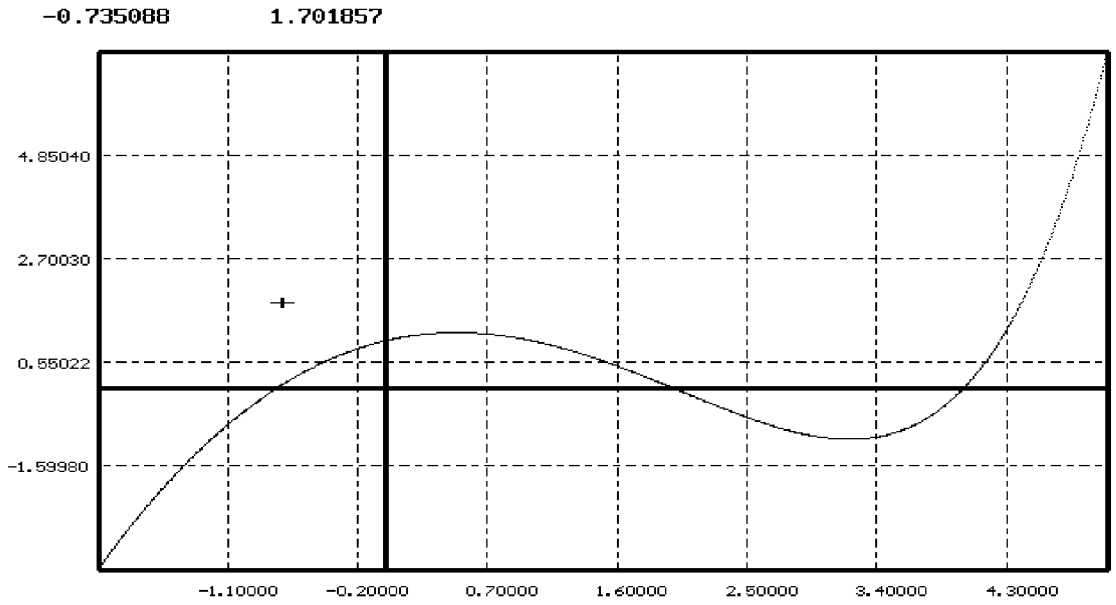

Figure 5 Graph of $y = 2^x - x^2$

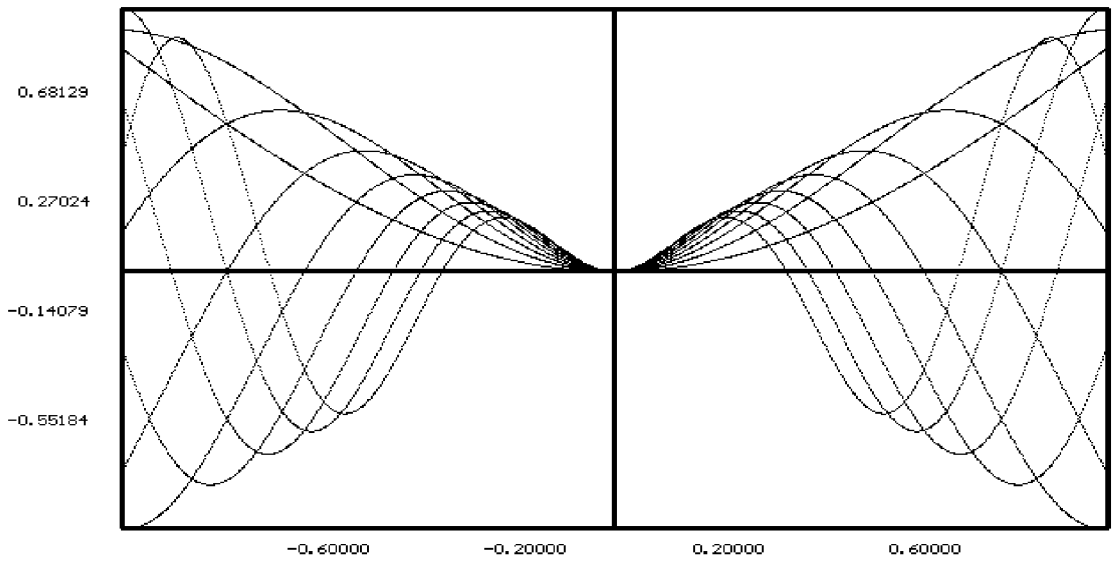

Figure 6 Family of curves $y = a\sin(ax)$ for $1 \le a \le 9$

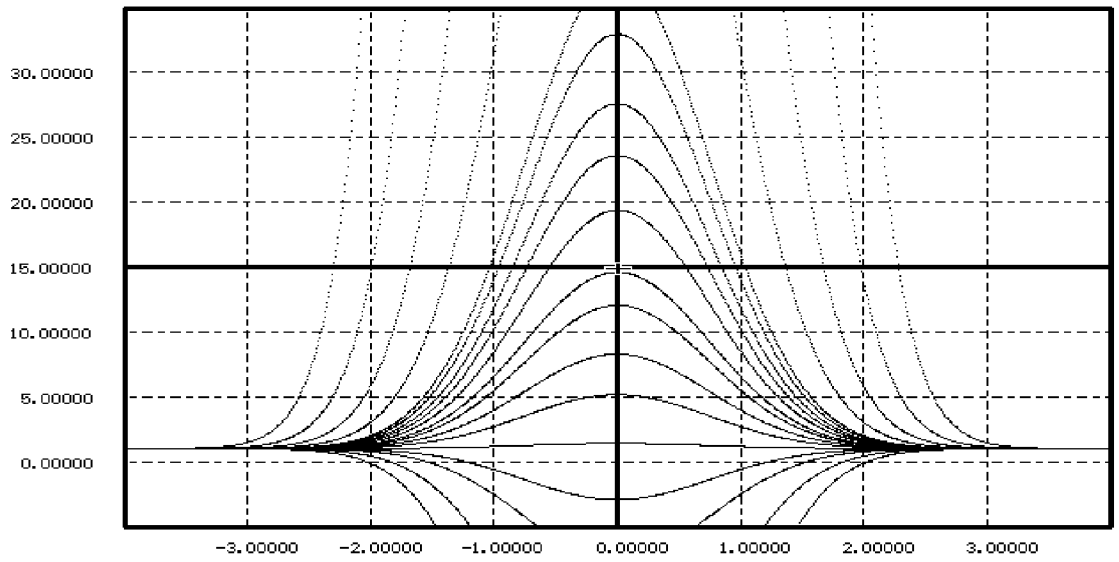

Figure 7 Family of integrals to $dy/dx = x - xy$

References

Abramowitz, M., and Stegun, I. A., (1972) *Handbook of Mathematical Functions,* Dover.

Anton, H., (1988) *Calculus with Analytical Geometry, 3rd edition.*

Bridger, M., (1986) Turbo Pascal 3.0—an update on Borland's compiler, *Byte*, February, 1986.

Clayton, D., Farrands, P., and Kennedy, M., (1987) Computer Enhancing Using a Function Graph Plotter, *Proceedings of ASCILITE '87*

Finney, R.L., and Thomas, G.B., (1988) *Calculus and Analytical Geometry, 7th edition*.

Hunt, R.A., (1988) *Calculus with Analytical Geometry*

Isaacs, G., (1987) Athena, Andrew, and Stanford—a look at implementation and evaluation in three large projects, *Proceedings of ASCILITE '87*

Kinko (1988) *Kinko's Academic Courseware Exchange—Fall 1988 Catalog*

Sherwood, B.A., and Sherwood, J.N. (1988) *The cT Reference Manual* and *The cT Language,* Carnegie-Mellon University.

Simmond, J., And Sugden, S.J. (1988) *Computer Mathematics Using Pascal, 2nd Edition*, ISBN 0 9587483 0 6.

Sorli, R., The New Maths, (1988) *Proceedings of ASCILITE '88,* p233.

Wolfram, S., (1988) *Mathematica—A System for Doing Mathematics by Computer,* Addison-Wesley, 1988.