DYNASIM: EXPLORATORY RESEARCH IN BIFURCATIONS USING INTERACTIVE COMPUTER GRAPHICS

Ralph H. Abraham

Department of Mathematics
University of California
Santa Cruz, California 95064

Historical Background

The crucial early experiments regarding bifurcation theory and applications as the experimental branch of differentiable dynamics may be described in three overlapping periods. The *period of direct observation* may be much older than we think, but let us say it begins with the musician, Chladni, contemporary of Beethoven, who observed bifurcations of thin plate vibrations. Much can still be learned from his work, painstakingly reproduced by Waller.[1] Analogous phenomena discovered in fluids by Faraday are still actively studied.[2-4] These experiments, so valuable because the medium is real, suffer from inflexibility—especially in choosing initial conditions.

The next wave of bifurcation experiments, which I shall call the *analog period,* begins with the triode oscillator. The pioneering works of van der Pol, with improvements by Hayashi, produced a flexible analog computer, and institutionalized the subharmonic bifurcations. These devices offer exceptional speed of convergence, but even with the recent development of modular electronics, only a limited class of dynamical systems are tractable.

The development of the early computing machines ushered in the *digital period.* Well-known numerical methods were implemented from the start, and graphical (CRT) output began to appear in the literature by 1962. The pioneer papers of Lorenz,[5] and Stein and Ulam,[6] are still studied. By 1967, the Association for Computing Machinery recognized this new field with a symposium entitled "Interactive Systems for Experimental Applied Mathematics."[7] Special systems for experimental math that have evolved since the Culler-Fried device of 1961 are fully described by Smith.[8] The current state of the art is now readily available in the form of a very large general purpose computer with BASIC or APL language and a color video graphics terminal. The currently available terminals of this type are listed in Table 1.

An equivalent, less expensive system would replace the large computer with a minicomputer, and a fast array processor. Such systems now exist at several institutions.

These devices are extremely flexible, accommodating a very wide class of dynamical systems, but suffer from the cost/resolution quandary: high resolution implies either a vast machine (high capital costs) or long run times (high operating costs.)

Our experiences over the past three years with forced oscillation machines in all three categories (direct observation of fluids, analog systems, and digital com-

0077-8923/78/0316-0676 $1.75/1

puter graphics) has resulted in the design of a fast, economical, special purpose digital computer graphic device, the *dynasim device,* which is described here.

Conventional Computer Graphic Techniques

The dynasim device is designed around a new algorithm for dynamical systems, the *push-pull algorithm*, which is described in the next section. Here, for background, we describe the techniques now in use with general purpose computer graphic hardware to draw the phase portrait of a dynamical system.

To fix for once and for all the goal of these techniques in bifurcation research, we now specify the context. Let C and P be manifolds; $\mathfrak{X}(P)$, the space of smooth

Table I

Color Video Graphic Display Systems

Manufacturer	Model	Resolution	Max. Depth
Aydin Controls	5212/5214A	240 × 256	12
Fort Washington, Pa.		512 × 640	12
DeAnza Systems	IEC 2212	256 × 256	12
Santa Clara, Cal.			
Genisco Computers	GCT 3000	256 × 256	16
Irvine, Cal.		512 × 512	16
		1024 × 1024	16
Grinnell Systems	GMR-26	256 × 256	32
Santa Clara, Cal.		512 × 512	8
Interpretation Systems	VDI	256 × 256	32
Overland Park, Kansas		512 × 512	8
Lexidata	200-D	256 × 256	3
Burlington, Mass.			
Ramtek	6000	256 × 512	3
Sunnyvale, Cal.	9000	256 × 256	24
		512 × 512	6

vector fields on P; and $\mu\colon C \to \mathfrak{X}(P)$ a family of vector fields, with smooth graph map

$$\Gamma_\mu\colon C \times P \to TP\colon (c,p) \mapsto \mu(c)(p).$$

We want to know for each control value, $c \in C$, the complete phase portrait of $X_c = \mu(c)$, a dynamical system (vector field) on the phase space, P. But machine computation can never reveal the asymptotic motions (ω-limit sets) of probability zero (separatrices, basic sets of hyperbolic type) so we ask only for the attractors, their basins, and (optionally) the phase foliation of each basin into isochrons (stable manifolds) of points of the attractor. As the control parameter, $c \in C$, can only be sampled at a finite set of points, our problem is just to *draw the ABP portrait* (attractors, basins, and phase foliations) of a fixed vector field. To be

reasonable, let $P \subset \boldsymbol{R}^n$ be an open box,

$$P = I_1 \times \cdots \times I_n, \quad I_\alpha = (a_\alpha, b_\alpha) \subset \boldsymbol{R}.$$

We now describe the conventional algorithm, using color video graphic output from a general purpose computer, for the ABP portrait:

(1) Write a vector field, X, into the program.

(2) Define a finite set of initial points in P, such as a uniform grid, and a time duration, $t_0 > 0$.

(3) Draw the integral curve forward from each until $t = t_0$, using the Runge-Kutta or Adams-Moulton formula.

(4) Continue until $t = kt_0$ if necessary, until all ω-limit sets are identified in white (PUSH).

(5) By manual (interactive) graphic input, define a new grid of initial points near the attractors, variously colored.

(6) Integrate retrograde $(-X)$ until $t = kt_0$ (PULL).

(7) Observe the ABP portrait, with basins and isochrons (if any) in different colors.

(8) Repeat in a smaller domain if needed, for higher resolution.

Our experience with this algorithm, even for a single basin system such as the van der Pol equation and with a very convenient interactive program (ORBIT, written by R. Palais) indicates that most of a day is required for a single portrait. A dedicated computer, and fast array processor, would reduce this interval to perhaps an hour. Important discoveries revealed by such experiments are understandably rare. Subharmonic resonance, for example, might still be unobserved if analog devices were not available. Note that if $C \subset \boldsymbol{R}^2$ (say force and amplitude) and $P \subset \boldsymbol{R}^2 (X_c$ = forced Duffing or van der Pol equation) the flow

$$F: C \times P \times \boldsymbol{R} \to P$$

was a domain of five-dimensions, so with a resolution of $w = 2^b$ bits for real numbers, $w^5 = 2^{5b}$ calculations are necessary. To shorten the time requires parallel processing (more calculations per unit time) or a faster processor (there is a limit) and increased device cost, either way. Doubling the resolution increases the cost by a factor of $2^5 = 32$, or lengthens the time of computation by the same factor.

This cost/resolution or speed/resolution problem is the main obstacle to effective experimental work in bifurcations. Yet many conjectures of bifurcation theory, and of differentiable dynamics, could be furthered by such experiments. See, for example, the problem set of Palis and Pugh.[9]

The Push-Pull Algorithm

This technique, inspired by our studies of the video feedback phenomenon,[10] is based on the induced action of a map upon subsets or functions.

If $\varphi: P \to P$ and $f: P \to \boldsymbol{R}$, let $\varphi^*(f) = f \cdot \varphi$ denote the *pull-back* of f by φ. Then $\varphi^*: \boldsymbol{R}^P \to \boldsymbol{R}^P$ is the induced action on real-valued functions on P, and like-

wise, $\varphi^*: 2^P \to 2^P: A \mapsto \varphi^{-1}[A]$ pulls back subsets. Considering only functions with discrete values, say z possible values, we have $\varphi^*: z^P \to z^P$, which pulls back partitions of P into z disjoint subsets. If $\varphi: P \to P$ is bijective, then we also have a push-forward action on functions, $\varphi_* = (\varphi^{-1})^*$. For example, $\varphi_*: 2^P \to 2^P: A \mapsto \varphi[A]$ takes disjoint sets into disjoint sets.

If $X \in \mathfrak{X}(P)$ is a vector field on P, choose a common (for all initial points) step-size, h, for numerical integration, fix $t = nh$ for some (large) integer n, and compute the flow diffeomorphism $\varphi_t \cong (\varphi_h)^n$ by the chosen formula, say Runge-Kutta. Rather than integrating separate initial points ad infinitum, our idea is to calculate the graph of φ_t at a fixed grid (as fine as memory resources allow) and store the graph as a look up table. Then this map, $\varphi = \varphi_t$, will be iterated (with its approximate, calculated inverse, $\psi \cong \varphi_{-t}$) as a cascade, without further numerical integrations, to draw the phase portrait, coarsely, of X. Dispensing with initial points, we choose a partition $f \in z^P$ of P into z disjoint sets. Typically, $z = 8$ or 16, in a practical trial. In fact, f is visualized (with a color video graphic terminal) as a colored map (partition) of P. Now push forward with φ, look at the image map $\varphi_* f$, and iterate. This is the *push cascade.* A few trials, with different partitions f, reveals the most probable attractors after n iterates, say. Then a new partition g is chosen by inspection, covering each attractor with a different color. Iteration of the *pull cascade*, $\psi_* g \cong \varphi^* g$, for n iterates (or more) colors the basins in different colors. This is the partition desired, as the goal of the ABP portrait. Finally, one attractor may be covered by small boxes of the various colors, in a new partition h. Iteration of the pull-back cascade, $\Psi_* h$, reveals (roughly) the phase foliation in the basin of the chosen attractor. This can be accomplished for all the basins, in a single step, yielding the ABP portrait.

Comparison of this algorithm with the conventional (orbit-by-orbit) technique described above shows that they are identical, except for the order of steps, and the more important distinction: repeated numerical integration (serial calculations) is replaced by storage of the graph (serial access to mass memory).

The Dynasim Device

The push-pull algorithm sacrifices resolution in favor of speed. Implementation of the algorithm in conventional general purpose computers requires an enormous core, or cache memory, rarely available to mathematicians. Special purpose, fast, random-access memory (RAM) is available as a peripheral device for general purpose computers, but is very expensive at present. An ideal example is the Array Processor (of Floating Point Processors, Inc., Portland, Oregon.)

The idea of the dynasim device is to read the graph of the transformation, φ, always in the same serial order. Thus, serial access memory (SAM) can be used in place of RAM for the graph memory. Current SAM units (CCD) are much less expensive than RAM, and this inequality is expected to increase in the coming years. Even with the more expensive RAM storage, the dynasim device is substantially less expensive than general purpose hardware.

The dynasim device, illustrated in FIGURE 1, is a peripheral device for a general purpose computer, to which it may be serially interfaced, and made of existing

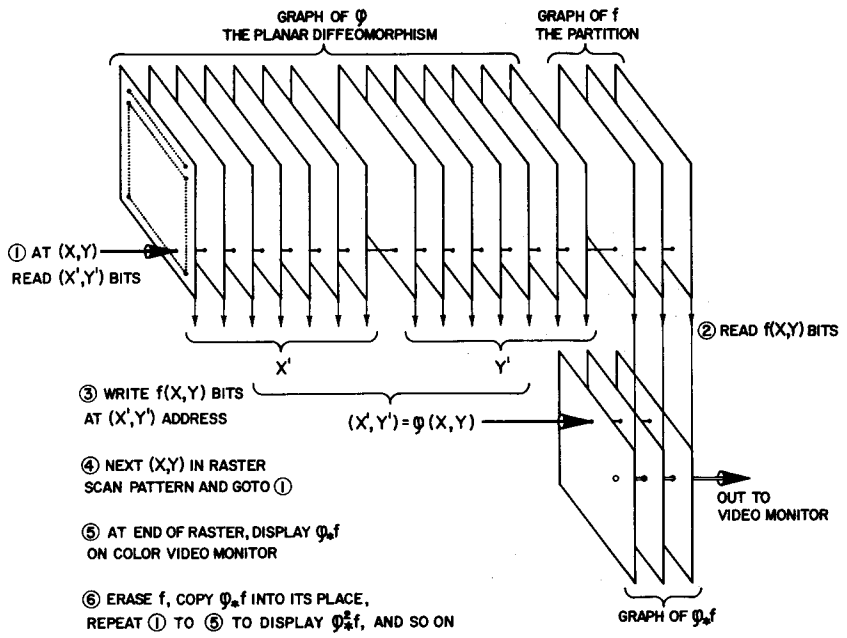

FIGURE 1. Schematic of the dynasim device, 128 × 128 pixels by eight colors.

hardware modules of computer graphics technology. It contains a number of parallel planes of memory, each a bit matrix corresponding to the domain P. For simplicity, let P be a two-dimensional unit square, discretized into a $w \times w$ array. Then each memory plane in the dynasim device is a $w \times w$ array. Suppose $w = 2^x$. Then the graph of φ is stored in $2x$ planes. If $\varphi(i,j) = (i',j')$, $i,j,i',j' = 0,\ldots,2^x - 1$, then x planes store the i' in binary, and the remaining x planes store j' in binary. An additional three planes store the partition (color) $f(i,j)$ if $z = 2^3$. Finally, three more planes record the push-forward, $\varphi_* f$, as it is computed, point-by-point. These last are read by a conventional color video graphic imaging device, creating a visible copy of $\varphi_* f$. In fact, the entire device can be built within a video graphic image device, if it has a large enough card cage to contain the memory planes, $2x + 6$ planes of $2^x \times 2^x$ bits. Commercially available devices are listed in TABLE 1, and the memory size and costs of the dynasim device, for various x values, are tabulated in TABLE 2.

To iterate φ_*, the f planes are erased, f and $\varphi_* f$ memories swapped (electronically), and another routine video raster scan proceeds. Each video frame (1/30th second) executes an iterate of the push (or pull) cascade.

In practice, the dynasim device would implement the push-pull algorithm as follows:

(1) The vector field is written into the numerical integration program in the core of the computer (CPU).

(2) The domain box is written in, and the dynasim resolution, $w \times w$. The program calculates the array of initial points (x_i, y_j).

(3) The CPU program calculates $\varphi_t(i,j) = (i',j')$ and $\varphi_{-t}(i,j) = (i'',j'')$, stores these graphs in arrays in its mass storage, and copies the graph of $\varphi = \varphi_t$ into the dynasim device.

(4) The operator indicates a partition f with a data tablet, light pen, or cursor. The CPU loads the partition into the dynasim device. At "PUSH", the device iterates the push cascade until "STOP." Repeat with another trial partition if necessary.

(5) The operator describes a new partition g (or h) to the CPU with interactive graphics input device.

(6) The CPU downloads g into the dynasim device, and the inverse map φ_{-t}. At "PULL," the device iterates the pull cascade until "STOP."

(7) Observe the AB (or ABP) portrait, with basins and isochrons (if any) in different colors.

(8) Repeat in smaller domain if needed, for higher resolution.

As in the conventional algorithm described previously, the output portrait can be stored in CPU mass storage, local floppy disc, videotape with verbal annotations, or color photograph. The time required for a successful portrait of a single vector field in two-dimensions can be estimated. Steps (3) and (6) require transfer of a large array, of $2^x \times 2^x \times 2x$ bits. If the CPU/Dynasim interface is serial, at 9600 BAUD, and $x = 8$, then about two minutes are required for downloading, in each step. The other steps take a few seconds, only.

And not only is this many times faster than the conventional algorithm for a single vector field, but if a bifurcation is being explored, the steps can be overlayed. Thus, the CPU integrates X_2, while the X_1 flow is being downloaded, and so on.

While speed has been obtained at the cost of resolution, the accumulation of round off errors will not indicate false attractors, but only miss some, and

TABLE 2
DYNASIM MEMORY REQUIREMENTS
EIGHT COLOR VERSION

2-Dimensional Resolution (pixels)	3-Dimensional Resolution (pixels)	Memory Size Per Plane (Cube) (kilobytes)*	Total Memory Cost	
			(RAM/RAM)†	(RAM/SAM)‡
128 × 128	25 × 25 × 25	2	$ 2,000	$ 1,020
180 × 180	32 × 32 × 32	4	4,200	2,100
256 × 256	40 × 40 × 40	8	8,800	4,320
360 × 360	50 × 50 × 50	16	18,400	8,880
512 × 512	64 × 64 × 64	32	30,400	18,240
720 × 720	80 × 80 × 80	64	80,000	37,440
1024 × 1024	100 × 100 × 100	128	166,400	76,800

*Kilobyte = 2^{13} = 8192 bits.
†At $50/kilobyte.
‡At $15/kilobyte.

misrepresent the shape of basins through the well studied phenomenon of *aliasing*. We now turn to this problem, common to all simulation schemes.

Decidable Stability

Suppose that a dynasim device is available—or a general purpose computer fast enough to execute the push-pull algorithm—with an interactive color video graphic terminal for graphic input and output. If a vector field is entered, an ABP portrait is returned, showing calculated attractors, basins, and phase foliations. No matter how brightly colored, the picture is but a pale ghost of the time phase portrait. We have formula error and round-off error for the diffeomorphism φ. As the round-off error dominates in the context of the practical design characteristics described above, we may forget formula error altogether. The round-off error for φ is reducible by zooming in: a small piece of the domain is restudied, using all available memory. But this will still miss long thin basins of attraction. We will expect in practice:

(1) The discrete φ is many-to-one, not onto.
(2) The $\varphi * f$ partition gets errors on the boundaries of the colored regions.
(3) The image $(\varphi *^n f)$ shrinks with increasing iteration.
(4) The screen eventually goes black, except for twinkling attractors.

As not all the qualitative features of a dynamical system are decidable (by machine) we seek now *decidable versions of stability and bifurcation.*

Firstly, we take account of the fact that only relatively probable attractors are machine discoverable.

Let M be a manifold; $X \in \boldsymbol{X}(M)$, a vector field on M with complete flow; and $\{A_i \mid i \in I\}$, the set of all topological attractors of X. Let B_i denote the maximal basin of A_i, $B = \bigcup\{B_i \mid i \in I\}$, and $S = M \setminus B$, the *separatrix* of X.

Now suppose M has a uniform topology, let S^ϵ be the ϵ neighborhood of S for $\epsilon > 0$, and $B^\epsilon = M \setminus S^\epsilon$. Then $B_i^\epsilon = B_i \cap B^\epsilon$ is the *ϵ-reduced basin* of A_i.

Given two vector fields, $X, Y \in \boldsymbol{X}(M)$, and $\epsilon > 0$, we say X and Y are *ϵ-basin equivalent* if $X \mid B_X^\epsilon$ and $Y \mid B_Y^\epsilon$ are topologically equivalent.

Now this looks more machine approachable, as the fine structure at the separatrix (homoclinic cycles, etc.) has been pruned away. Further, many small basins of attraction (infinitely many, no doubt) have been thrown out with the ϵ-neighborhood of the separatrices. Yet many small basins remain (perhaps too small to be machine discoverable) that are nowhere near the separatrix. So now, we will throw these out as well.

Let M now have a probability measure μ, and for simplicity suppose that μ and the uniformity of M are both derived from a common metric (distance) function. Without assuming compactness, we may suppose that the volume of an ϵ-disk, $\mu[N_\epsilon(m)]$ for any $m \in M$, is bounded below, by $k\epsilon^d$, where $k > 0$ is a constant, and $d = \dim(M)$.

Now suppose, for the sake of discussion, that all basins are measurable. The probability of attractor A_i is then $p_i = \mu(B_i)$, and $p_i^\epsilon = \mu(B_i^\epsilon)$ is substantially smaller, perhaps even zero. Our discrete algorithm can discover an attractor

only if its ϵ-reduced basin is sufficiently large. Thus, let

$$I^\epsilon = \{i \in I \mid p_i^\epsilon > k\epsilon^d\}$$

and $D^\epsilon \subset B^\epsilon$ be the union of the discoverable basins,

$$D^\epsilon = \cup \{B_i \mid i \in I^\epsilon\}.$$

At last, we say $X, Y \in \boldsymbol{X}(M)$ are *ϵ-decidably equivalent* if $X \mid D_X^\epsilon$ and $Y \mid D_Y^\epsilon$ are topologically equivalent.

Likewise, $X \in \boldsymbol{X}(M)$ is *ϵ-decidably-stable* if every Y sufficiently close to X is ϵ-decidably-equivalent, and a family of vector fields $\{X_\mu \in \boldsymbol{X}(m) \mid \mu \in \boldsymbol{R}^c\}$ has an *ϵ-decidable bifurcation* if not all its members are ϵ-decidably-equivalent.

Here, ϵ is not to be chosen arbitrarily small, but is a fixed fraction of the width of the domain, a characteristic of the simulation machine at hand.

Conclusion

We have described here a decidable version of the stability and bifurcation concepts of differentiable dynamics, an algorithm for exploring the decidable bifurcations in low dimensions using interactive computer graphics, and an inexpensive special purpose computer for implementation of the algorithm. The design characteristics of this system may now be correlated as follows: The cost of the device, C, is proportional to w^d, where w^d is the number of bits in the width of the domain, and d is the dimension. If $\epsilon = 1/w$, attractors of reduced probability ϵ^d can be resolved. Thus, the resolution is inversely proportional to the cost. The smaller ϵ, the more basins discovered, and the more bifurcations observed, especially, in the neighborhood of the separatrix (where the principal action is.)

We propose that dynasim devices are useful not only in bifurcations research, to develop theory, but also in applications, as fast, graphic presentation of the most probable equilibrium states is one of the goals of the qualitative theory.

References

1. Waller, M. D. 1961. Chladni Figures: A Study in Symmetry. G. Bell. London.
2. Brook Benjamin, T. & F. Ursell. 1954. Proc. R. Soc. London A **225:** 505–517.
3. Abraham, R. 1976. The macroscopy of resonance. *In* Structural Stability, The Theory of Catastrophes, and Applications in the Sciences. P. Hilton, Ed.: 1–9. Springer-Verlag. New York, N.Y.
4. Marsden, J. E. & M. McCracken. 1976. The Hopf Bifurcation and its Applications. Springer-Verlag. New York, N.Y.
5. Lorenz, E. N. 1962. The statistical predictions of solutions of dynamic equations. Proc. Internat. Symp. Numerical Weather Prediction, Tokyo: 629–635.
6. Stein, P. R. & S. Ulam. 1964. Nonlinear transformation studies on electronic computers. Rozprawy Mat. **39:**66.
7. Klerer, M. & J. Reinfelds, Eds. 1968. Interactive Systems for Experimental Applied Mathematics. Academic Press. New York, N.Y.

8. SMITH, L. B. 1970. A survey of interactive graphical systems for mathematics. Computing Surveys **2**: 261–301.
9. PALIS, J. & C. PUGH. 1975. Fifty problems in dynamical systems. *In* Dynamical Systems—Warwick 1974. A. Manning, Ed.: 345–353. Springer-Verlag. New York, N.Y.
10. ABRAHAM, R. 1976. Simulation of cascades by video feedback. *In* Structural Stability, the Theory of Catastrophes, and Applications in the Sciences. P. Hilton, Ed.: 10–14. Springer-Verlag. New York, N.Y.