NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A SYSTEM DYNAMICS BASED STUDY OF SOFTWARE REUSE ECONOMICS

by

Pamela Jean Gallup

June, 1994

Thesis Advisor: Tarek Abdel-Hamid

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
June 1994

3. REPORT TYPE AND DATES COVERED
Master's Thesis, Final

4. TITLE AND SUBTITLE
A System Dynamics Based Study of Software Reuse Economics

5. FUNDING NUMBERS

6. AUTHOR(S) Pamela J. Gallup

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT *(maximum 200 words)*

Software productivity is a critical issue for Government agencies and the Department of Defense. Satisfying the enormous demand for new software and reducing cost creates pressure to develop new software production techniques. Of these, one of the most promising, strongly supported by Government and DoD agencies, is software reuse. This thesis studies the economics of software reuse using a system dynamics computer model, the Dynamica Reuse Model, which simulates the activities of a software development organization engaged in organization-wide, systematic software reuse. Long-term relationships between reuse rate, productivity, and unit cost are studied by varying consumption cost, production cost, employee turnover rate, and reusable component retirement age. Results suggest long-term steady state relationships may be different from short-term dynamic state relationships. After validation and customization the Dynamica Reuse Model can be used to support an organization's cost and schedule software tools. Increasing understanding of the software development process in order to make knowledgeable rather than intuitive predictions about organizational variables related to reuse such as reuse rate, productivity, and unit cost, enable the model to serve as a management support tool for the complex and costly process of software development.

14. SUBJECT TERMS
Software Reuse Economics, Software Development, Dynamica Reuse Model

15. NUMBER OF PAGES
213

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

Approved for public release; distribution is unlimited.

A System Dynamics Based Study of Software Reuse Economics

by

Pamela Jean Gallup
Lieutenant Commander, United States Navy
B.S., Old Dominion University, 1978
M.S., Old Dominion University, 1981

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
June 1994

Author: Pamela J Gallup
Pamela Jean Gallup

Approved by: T Hamid
Tarek Abdel-Hamid, Thesis Advisor

James C Emery
James C Emery, Associate Thesis Advisor

David R. Whipple, Chairman
Department of Systems Management

ABSTRACT

Software productivity is a critical issue for Government agencies and the Department of Defense. Satisfying the enormous demand for new software and reducing cost creates pressure to develop new software production techniques. Of these, one of the most promising, strongly suppor[illegible]ernment and DoD agencies, is software reuse. This thesis studies the economics of s[illegible]ware reuse using a system dynamics computer model, the Dynamica Reuse Model, which simulates the activities of a software development organization engaged in organization-wide, systematic software reuse. Long-term relationships between reuse rate, productivity, and unit cost are studied by varying consumption cost, production cost, employee turnover rate, and reusable component retirement age. Results suggest long-term steady state relationships may be different from short-term dynamic state relationships. After validation and customization the Dynamica Reuse Model can be used to support an organization's cost and schedule software tools. Increasing understanding of the software development process in order to make knowledgeable rather than intuitive predictions about organizational variables related to reuse such as reuse rate, productivity, and unit cost, enable the model to serve as a management support tool for the complex and costly process of software development.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENT

The author thanks the U.S. Navy for giving her the opportunity to study in the Information Technology Management curriculum at the Naval Postgraduate School.

The author expresses deep gratitude and respect for her thesis advisor, Professor Tarek Abdel-Hamid, for challenging her intellectually and with humor, for being so organized and providing guidance on a regular schedule, for providing encouragement and admonishment at the correct times; in short, for providing her with such a rewarding and educational thesis experience.

The author is deeply grateful for the exceptional support, understanding, and encouragement provided by her husband, Lieutenant Commander Shelley P. Gallup, who cheerfully shared the work and the responsibilities of our family and home during these two years of academia.

I. INTRODUCTION

A. WHAT IS SOFTWARE REUSE?

Software reuse is an approach to software development that reuses software components instead of building software components for one-time or unique use. "Use of existing software components to construct new systems is the definition of software reuse" (Prieto-Diaz, 1993). Reusable components span the gamut of the entire software development process, and include requirements specifications, system specifications, architectures, detailed design structures, source code fragments, documentation, test plans, test data, tools, and environments (Bollinger and Pfleeger, 1990; Prieto-Diaz, 1993).

The key activities of any reuse process are: "identification of candidate reuse objects for a given reuse request, evaluation of their potential and then selecting the best-suited one, modification of the selected object, and its integration into the ongoing development." (Rombach, 1990)

Software reuse can be approached as an ad-hoc process or as a very deliberate, structured process, worthy of significant investment. Ad-hoc reuse is akin to "scavenging," and can include reuse both vertically in the same domain and horizontally across domains, reuse by informal reverse engineering, and reuse by composing new systems from existing components that are either black-box (not modified) or white-box (modified) (Schimsky, 1992; Prieto-Diaz, 1993). Ad-hoc reuse at its best can include generation of reusable components for a project, but is not an organization-wide approach (Schimsky, 1992).

By, comparison, in systematic reuse, an organized effort is made "to analyze the need, potential application areas, and payoff of a software reuse library, and then build and maintain that library" (Schimsky, 1992). Systematic reuse "emphasizes the reuse of predeveloped solutions across the entire life" (Hooper and Chester, 1990), and occurs when components, i.e., solutions, are specifically designed to be reusable, are stored to be easily accessed and reused, and adapted in order to fit into a particular software project. This is a radical departure from traditional single-use software development, where software components are developed for a project and subsequently changed only for maintenance.

At the Software Development '93 Conference, Microsoft Chairman Bill Gates summarized a new way of thinking about software development: "Reuse before you buy, buy before you build," implying all-new component development should be used only as a last resort. He proposes a three-tiered model for software developers. In the first tier, sophisticated programmers create generic components for "resale" or reuse. In the second tier, these reusable components are categorized and stored in repositories or libraries for later retrieval and use in specific applications. In the third tier, the system creators retrieve generic components from the repository and compose into specific applications (O'Brien, 1993).

B. POTENTIAL BENEFITS OF SOFTWARE REUSE

Software development is faster and cheaper if systems are built from preexisting reusable modules rather than from unique-use components. In addition, benefits accrue throughout the lifecycle, since maintenance costs are reduced through using carefully tested modules (Rymer, 1993). Software reuse is an important tool in developing systems that can grow with technology and not be made obsolete by it (Schwartz, 1992). "Software reuse is regarded as a key to improving software development productivity and quality" (Kim and Stohr, 1992).

For some people, software development is an art form, or at the very least, a h[illegible]individualized activity. This attitude results in software developers with a "cowboy" mentality, i.e., "real men write their own software" (Yourdon, 1994). The big problem with this viewpoint is that resultant software productivity levels are agonizingly insufficient to satisfy the huge demands placed upon the software development industry. While hardware productivity has increased tremendously and fueled the current high availability of personal computing, software productivity has not kept pace. Clearly, "it will be necessary to institute fundamental changes in the way software is developed" (Williams, 1991).

Reusing software is widely acknowledged as part of the answer to the problem of low software productivity. There are many reasons to embrace software reuse. Because good software engineering practices are followed from the beginning of development of reusable component and errors are identified early due to extra reviews and additional testing, software reliability is increased (Matsumura et al., 1990; Margono and Rhoads, 1992). The software development lifecycle is shortened and software price is lowered (Schimsky, 1992). Reuse improves productivity and quality (Incorvaia and Davis, 1990; Matsumura et al., 1990).

C. DOD'S SOFTWARE REUSE POLICY

"Computers form an integral part of virtually every modern weapon system, and every computer requires software. The cost of this software is a multi-billion dollar line item in the annual DoD budget" (Schimsky, 1992). "DoD's demand for new software is equal to the entire amount it currently has in use" (Kitfield, 1989). The old ways of software development, one line at a time, are simply not able to supply the demand. "Virtually every time-critical, embedded program used in military weapon systems is constructed manually, one statement at a time" (Schimsky, 1992).

Government-mandated policies for an enormous beauracracy such as the Department of Defense tend to lag behind the faster-moving, more autonomous commercial and private sector information technologies. However, as the "world's largest consumer of information system resources," the Department of Defense's (DoD) software development policies have an enormous impact upon the process of software development (Emery and Zweig, 1994).

1. Software Reuse Initiative (1991)

DoD expects to reap the greatest cost savings in software development through software reuse. Two organizations responsible for DoD's software established a Memorandum of Agreement in November 1991 to form a cooperative partnership (GAO, 1993):

- Embedded systems and information technology research is handled by the Director for Defense Research and Engineering (DDR&E);
- Information systems and command and control systems are handled by the Assistant Secretary of Defense for Command, Control, Communications and Intelligence (ASDC3I).

On the basis of this agreement, the Director for Defense Information software reuse initiative proposed an alliance of DoD reuse activities and the three major software reuse programs. The three major software reuse programs are: CARDS (or Central Archive for Reusable Defense Software from the Air Force); STARS (or Software Technology for Adaptable Reliable Systems from DARPA or the Defense Advanced Research Projects Agency); and the Software Reuse Program (from DISA or the Defense Information Systems Agency). The executive steering committee of this alliance has representatives from the ASDC3I, DDR&E, Joint Staff, Army, Navy, Air Force, DISA, Defense Logistics Agency, Defense Intelligence Agency, and the National Security Agency. (GAO, 1993)

Defense's software reuse initiative has a strategy detailed in ten elements. The basic thrust of these elements is to identify reuse opportunities and establish processes to capitalize on those opportunities (GAO, 1993). Denis Brown, director of DISA's Center for Information Management, says DoD's reuse efforts will focus on software architectures and "systematic, not opportunistic, reuse." Ada will not be the only language used in DoD's software reuse repository, even though it is the mandated programming for all DoD software development. Classifying the code is considered crucial to maximizing the effectiveness of the repository, and a committee composed of many different DoD organizations is studying this problem. (Ferris, 1993)

2. DoD's Software Reuse Initiative Reviewed (1993)

In February 1992, the Chairman of the Subcommittee on Defense, House Appropriations Committee, Representative John Murtha (D-Pa), requested the General

Accounting Office (GAO) to conduct a sweeping review of the Pentagon's software reuse initiative. The GAO's Information Management and Technology Division sent its report to The Honorable John P. Murtha in January 1993 (GAO, 1993). The GAO report points out the disparity between what DoD's software reuse initiative states it is based on, and what methods are available to make the initiative work. For example, there are no standard methods for processing and representing information about domains, yet domain analysis is central to the initiative. Also, there are no standard methods for classifying software for repositories, yet software repositories are an important component of the software reuse project. The GAO report discusses barriers to software reuse, such as the higher initial cost to develop reusable software and possible legal battles concerning intellectual property rights of code reuse among software suppliers, repositories and users (Endoso, March 1993).

3. Other DoD Reuse Policies

The Air Force's reuse plan hedges its bets toward immediate component reuse by including guidelines for designing components for future reuse, since "if extensive reuse of existing software is not possible, then reuse for future applications should be the objective." The Air Force's Software Reuse Incentive Policy "requires bidder's proposals facilitate reuse and rewards vendors that reduce project schedules and costs through reuse." (Endoso, August 1993)

The Naval Information Systems Management Center, commanded by Rear Admiral Robert M. Moore, has plans to create a "software executive officers council at flag and

Senior Executive Service level to work on reuse policy and issues." Another major part of the software implementation plan will be to name "domain managers" for software domains such as MIS and command and control, where the demand for r usable software is greatest. (Menke, 1993)

4. DoD Reuse Activities

DoD's reuse centers promote the sharing of domain-specific as well as cross-domain Ada software (Menke, 1993). There are several DoD software reuse repositories and centers (Schwartz, May 1993). DoD's infrastructure to support software reuse is steadily growing stronger. By January 1994, other DoD resources to be added to its linked software reuse libraries include the Defense Software Repository System (DSRS), the Air Force's Reusable Ada Avionics Software Packages, NASA's AdaNet, and the Computer Software Management and Information Center run by the University of Georgia for NASA (Schwartz, April 12 1993).

The ARC (Army Reuse Center) library in Falls Church, Virginia, contains an impressive one million lines of Ada code comprising 1,400 software components accumulated in a brief time span (Green, 1992). According to Marre Riggs, director of the ARC, by 1995 ARC should be able to match up donors with clients. ARC's goal is to be a bona fide reuse library, and has projected large life cycle cost savings through domain analysis, "the process of identifying commonalities among systems conducive to code reuse." Already 26 domains in the Standard Army Management Information Systems have been analyzed, and other reuse opportunities will come in the Reserve Component

Automation System and the Army WWMCCS Information System. ARC has identified about $400 million saved in the $2 billion Army Tactical Command and Control System, the battlefield systems integration project. (Green, 1993; Menke, 1993)

One key factor in the Army's reuse strategy is the depth of the Army's commitment to reuse. Program managers realize there are no short-term savings to be gained using Ada; however, they do see the long-term potential implicit in Ada, and look for ways to turn the Ada mandate to their own advantage (Green, 1993).

In April 1993, DoD established the first electronic link between software reuse laboratories, initiating the capability for government users and contractors to access all the software reuse libraries from desktop PC's. Using a command center approach, these software components will be accessible to software engineers at the desktop level, and are expected to significantly reduce programming costs. One end of the link is to the Air Force's Central Archive for Reusable Defense software (CARDS), where data is contained on Sun Sparcstations running the SunOS operating system. The other end of the link is a branch of the STARS program of DARPA, called the Asset Source for Software Engineering Technology (ASSET), where data is maintained on an IBM RS/6000 AIX workstation running on the Oracle relational database system. (Schwartz, 12 April 1993)

DoD's policies toward software reuse are in flux and will continue to change. DoD policies are strongly influenced by the House defense appropriations processes. The Defense appropriation bill for FY 1993 included $52 million earmarked for the Shared Resources Center, an initiative proposed by the Pentagon's Continuous Acquisition and

Life Cycle Support office (Endoso, October 1993). By March 1994 the Defense Department must submit a more formal software reuse program with centralized management to the House's Defense appropriation committee (Endoso, October 1993).

DoD's software reuse policy may not be capable of changing as fast as perhaps it ought to, but inexorably it is moving in the right direction. The DoD's Software Development and Documentation standard (Mil-Std-SDD to be renamed to Mil-Std-498) will incorporate software reuse principles by requiring evaluation of reusable software for engineering and provide evaluation criteria (Schwartz, April 26 1993).

D. THESIS OBJECTIVE

The objective of this thesis is to focus on the economics of software reuse using a system dynamic simulator called the Dynamica Reuse Model, a software program simulating a software producing organization engaged in software reuse. In order to better understand the complexity and dynamics of this model, the next chapter will focus on distinguishing characteristics of software producing organizations dedicated to producing and consuming reusable software components, i.e., organization-wide reuse. A clear understanding of these distinguishing factors will facilitate understanding the importance of the economics of software reuse.

II. ORGANIZATIONAL ISSUES IN SOFTWARE REUSE

A. SUCCESSFUL SOFTWARE REUSE REQUIREMENTS

Ironically, one of the industries closest to the source of many technological changes, software development, has been slowest to apply reengineering to the process of software development. "That reuse of software increases productivity has been known for some time. Nevertheless, it seems to be difficult to reach a high level of reuse" (Wolff, 1992).

This is beginning to change. Successful economic reuse practices demand a certain viewpoint: "Reuse needs to be viewed in the context of a total systems approach. Envision a software system or reuse support system that helps document and elucidate existing application systems" (Kim and Stohr, 1992).

Organizations committed to the idea of software reuse must not only change their way of conducting business, but must change the very way they are structured. To increase reuse levels, organizations must change their strategy by planning and instituting specific policies (Isoda, 1991). Reuse must be emphasized as an "integral part of an effective software engineering development and maintenance process" (Hooper and Chester, 1990). Issues in achieving high levels of reuse can be divided into two broad categories: technical and managerial.

B. TECHNICAL ISSUES

Detailed discussions of reuse engineering techniques (Bailey and Basili, 1990; Durin et al., 1990; Gall and Klosch, 1992), reuse metrics (Reifer, 1990; Tirso, 1991) and surveys of various methodologies and tools for reuse technologies (Kang and Levy, 1989; Mii and Takeshita, 1993) have been published. The following discussion will be more general in nature and covers reusable component production and consumption.

1. Reusable Component Production

Organizations serious about reuse concentrate on producing reusable components through an engineered process, i.e., using techniques such as domain analysis and principles such as open architecture, instead of through an opportunistic process, i.e., using techniques such as ad-hoc reuse or scavenging. This engineered process has three basic parts: domain analysis, reusable component creation, and reusable component classification and storage. (Palmer and Cohen, 1990)

The production cost of creating reusable components is higher than that for creating single-use, custom components. Included in reusable component production cost is the cost for domain analysis and the cost for providing information to users of reusable components, i.e., the cost of the repository, catalog, or library of reusable components. Higher production costs are justified because these production costs are amortized through reuse in multiple applications. (Palmer and Cohen, 1990)

a. Domain Analysis

Domain analysis is associated with *vertically reusable* components, as compared to *horizontally reusable* components. Horizontally reusable components are those used across a wide spectrum of application areas, and include components such as "data structures, sorting algorithms, user interface mechanisms" (Hooper and Chester, 1990). Vertically reusable components are those used within the same problem domain.

and this is where the highest leverage is available. The more overlap there is, the higher the reuse levels are. "Domain analysis assists in identifying areas of commonality from which components can be built" (Williams, 1991).

The easiest domain analysis occurs with applications within a narrow and well-defined domain (Kang and Levy, 1989). In contrast, it is more demanding to analyze domains for applications that are part of embedded systems with "hard real-time constraints, limited computer memory and data storage available, extremely high-reliability requirements, and require extensive, customer-mandated documentation" (Palmer and Cohen, 1990). One example is the U.S. Air Force's Common Ada Missile Packages (CAMP) project, which produced reusable software components for missile operational software (Palmer and Cohen, 1990; Drake and Ett, 1990).

Other DoD systems that are candidates for demanding domain analysis include tactical and strategic command and control problem domains, DoD manufacturing problem domains, and finally, aircraft, shipboard and land vehicle problem domains that have "guidance and control systems, navigation systems, offensive and defensive support systems, and weapons systems" (Drake and Ett, 1990).

b. *Creating Reusable Components*

The earlier a component is reused in the lifecycle, the greater the savings. This is where domain analysis really pays off, with the generation of domain-specific requirements, designs, algorithms, and test results. "Specification reuse appears to assist in structuring the problem space and defining the problem scope" (Sutcliffe and Maiden, 1990).

Probably the highest level of "reuse" is in the area of personnel. Someone who is intimately familiar with a domain may be the best choice for analysis of similar domains (Hooper and Chester, 1990; Drake and Ett, 1990). The organization can justify the higher cost of the expert because of the higher productivity of the expert and because potential problems are minimized by not using inexperienced developers (Kang and Levy, 1989). One approach is to establish a separate team whose sole function is to build and produce parts for the organization's library (Schimsky, 1992).

Good programming practices are especially important in developing reusable code, since the components will be used in subsequent projects with far-reaching results. Reusable code must demonstrate "understandability, reliability and maintainability" (Hooper and Chester, 1990). Since reusable components will be used in many projects, it is critical that the components be correct. Component validation and verification must be stringent for reusable components, and should stress portability and adaptability (Hooper and Chester, 1990).

One systematic approach to producing reusable components is referred to as "standardized components and their composition approach," using components that are black box, object-type, and domain-oriented. Three assumptions are required for this approach: (1) fixing the product architecture allows component development in advance, (2) standardizing the development environment through design methodology allows for efficient product development by teams, (3) developing the reusable components independently allows for product development by a different group. (Matsumura et al., 1990)

Another systematic approach further divides the software reuse lifecycle into two lifecycles. The generally accepted software reuse lifecycle starts with reusable component production, classification, and deposition, then proceeds into identification, modification, and composition of reusable components into application products. In contrast, the two lifecycle paradigm for software reuse makes the distinction between developing generic reusable products and developing application-specific products: the Generic Product Development Life Cycle produces generic domain assets for reuse, and the Application-Specific Product Development Life Cycle produces specific application systems. The generic product development process results not only in domain models, but also common-problem specifications, generic architectures, process models, domain-specific asset libraries, and test capabilities. "The motivation behind the two life cycles paradigm is to lessen the dependence on the individual engineer by focusing attention on the capturing of domain-specific knowledge." (Drake and Ett, 1990)

c. Classifying and Storing Components

Components should be stored and classified in a library or repository based on domain analysis or other system, such as the "faceted" classification method based on library science (Prieto-Diaz and Freeman, 1987). Good documentation, including relationship information between components, should be thoroughly adhered to (Hooper and Chester, 1990). The database should be set up so "that retrieval of any of the parts of a linked set will automatically inform the retriever of the existence and location of the other linked parts" (Schimsky, 1992).

Beyond the requirements of selecting a system for classifying and storing components, all the services normally associated with a library must be provided. "The Army's RAPID program, for instance, includes a full library staff, including administrative assistant, librarian, software engineers, system analyst designers, other technical consultants, and persons to train potential RAPID users" (Schimsky, 1992).

Some researchers propose that reusable component repositories "should present components to potential reusers as if the components were being marketed commercially." In this proposal, commercial off-the-shelf software products have many desirable characteristics that reusable components would do well to emulate:

·Visible components are well known to users.
·Supported components have someone who aids users, i.e., provides training, demonstrations, installation routines, and telephone support.
·Broad domain components are useful in a broad range of situations.
·Trustworthy components are tested and reliable.
·Accepted components conform to standards.
·Complete components include documentation. (Allen et al., 1990)

2. Reusable Component Consumption

a. Searching for and Retrieving Components

User-friendly interfaces between the user and the archive or library should be well planned from the very beginning. The processes for searching and retrieving components are heavily dependent on the system used for classifying and storing components, and should at least support query and browsing modes (Hooper and Chester, 1990). However, it is not at all clear that this should be computer based. In one case, users "preferred printed catalogues to computer tools because printed catalogues enabled faster search" (Isoda, 1991).

Sophisticated software metrics programs such as the Partial Metrics or PM System extract planning knowledge from a software system to automatically build a knowledge base. This knowledge base is used at each of the four phases of the process of software reuse, i.e., finding, understanding, modifying and composing reusable components. The knowledge base enables the PM System to "learn" criteria from the users to generate reuse decisions based upon examples of acceptable and unacceptable decisions at each phase. (Reynolds et al., 1992)

b. Understanding and Assessing Components

This is where good software engineering practices reap desired results, because if "good practices are followed in developing, classifying and storing reusable components, good understanding should be a natural by-product" (Hooper and Chester, 1990). With good understanding comes good reuse levels. Other information valuable in assessing components is the operational history of components, i.e., "number of uses, degree of satisfaction, and errors" (Hooper and Chester, 1990). How to get "good understanding" of reusable components varies according to what the component is. For example, "specification reuse in a CASE environment is unlikely to succeed without tutorial support" (Sutcliffe and Maiden, 1990).

c. *Adapting and Composing Components*

Usually some adaptation is necessary. This is where retaining design and specifications in the library pays off. Some code only requires input of parameters (Hooper and Chester, 1990). One enormous difference in *adapting* components for reuse and *designing* components from scratch is that the processes are fundamentally the opposite of each other. In adapting components, one *composes* existing components into a new system to solve the problem. In designing components from scratch, one *decomposes* a concept into subparts to solve the problem. (Williams, 1991)

Another important distinction between producing and consuming reusable components is that the software developer *producing* reusable components must be able to conceptualize the domain in *generic* terms. On the other hand, the software developer *consuming* reusable components must be able to conceptualize the domain in *specific* terms related to the application. (Drake and Ett, 1990)

C. MANAGERIAL ISSUES

Certainly from the viewpoint of a Government software project manager, saving money is one of the prime considerations in software reuse. Any economic evaluation of software reuse must balance the cost to obtain or produce reusable components (the production cost), and the cost to use or adapt reusable components (the consumption cost), against the cost to use single-use or unique components. A reusable component costs more to develop than a single-use component, "due to extra effort required to generalize the components, conduct extra testing, provide adequate documentation, and to classify and store them for reuse" (Hooper and Chester, 1990). This means the accounting side of the organization must have a long-term view, i.e., reusable component production cost must be amortized over the production cycles of all the applications that use a particular component.

1. Management and Organization

To get the highest productivity and the greatest cost savings by making reuse a standard organizational process, the organization must establish a support structure. One way to do this is to create a central support staff organization. At IBM's different software production sites, the support staff has a site coordinator or "reuse champion" who assists the project reuse leaders of each project. The site coordinator helps create and manage the site-wide reuse library holding certified reusable components from across the spectrum of problem domains for that site. The site coordinator represents the site at IBM's Corporate Reuse Council, which "sponsors work that benefits more than one site, such as the creation of standards, parts and tools." The project reuse leaders help develop the reuse strategy for each project, and with the site coordinator, specify reuse goals. These project leaders ensure educational needs are met, for example, "general reuse education, object-based design or object-oriented programming techniques." (Tirso, 1991)

2. Organizational Behavior

Organizational behavior is strongly dependent on corporate culture. If upper management supports reuse in concrete ways, then personnel will be more inclined to discard the "not invented here" syndrome. Indeed, in organizations where reuse is high, "individuals believed in and wished to promote reuse". . . especially "where software reuse seems to be part of the corporate culture" (Incorvaia and Davis, 1990).

One way to increase reuse is to have large inventories of components in the libraries, since the more reusable components are available to the programmer, the more likely the programmer is to look in the library first. To have large component inventories, contributors must write components that meet the library's criteria. One incentive system gives financial awards quarterly based on points earned for contributed items and for reuse of items (Tirso, 1991).

There are some indications that cultural differences between countries play a role. For example, the levels of reuse in Japan are from 60 to 70 percent. In the US, the levels are from 20 to 30 percent. In Japan, an up-front assessment of the level of reuse in particular projects results in schedule and budget adjustments. This implies a strong incentive to meet the level of reuse initially projected. Even the language used to refer to software errors is indicative of cultural attitudes. In the US, software errors are referred to as "bugs," implying they have a life of their own and are not due to developer error. In Japan, software errors are referred to as "spoilage," implying totally different things than the term "bug." (Yourdan, 1994)

In certain software producing organizations in Guadalajara and Manila, 25 percent of the staff is dedicated to reuse. Software developers are not allowed to write their own components unless they can prove they can't use a reusable component. "If you can't reuse a component, you must be doing something wrong." In Australia, funding is based on not only how much you can take out of a repository, but also on how much you can put in. (Yourdan, 1994)

3. Contractual and Legal Considerations

The traditional contracting arrangements of Cost-Plus and Firm Fixed Price don't encourage reuse at all. In fact, they encourage exactly the opposite behavior. Because so much money and effort must be invested up-front with the pay-off coming so much later, contracting arrangements must be radically different. It would be to the distinct advantage of both the purchaser and developer of software to develop a long-term, cooperative arrangement, thus capitalizing on the initial investments in domain analysis, training of domain experts, setting up a repository, and producing and modifying reusable components (Hooper and Chester, 1990). Government software managers must have incentives included in the contracting process "to allow creation of a reasonable base of software reuse libraries. After establishment of such a base, normal competitive pressures should assure continued software reuse by Government contractors." (Schimsky, 1992)

Legal issues revolve around accountability for a product's development, and are an increasing concern in Government contracting. Increasingly, Government project managers are expected to deliver clear requirements, and in turn, software contractors are to deliver products meeting these requirements. Contracts including software reuse will have to resolve issues of "what the Government gets and owns as its delivery from the prime (contractor) in the first place." (Schimsky, 1992) Thorny questions arising from software reuse policies in Government software contracts include:

> Should the Government mandate the use of specific, available software, even if it was not developed by the prime contractor? If that reused software fails, who is liable for the time and money needed for its repair, or for any damage its failure may have incurred? If total accountability for the prime contractor is the goal, must the Government select only those primes that have developed their own software reuse libraries and have proposed using them on Government contracts? . . . After using a company-owned reuse library to develop software for the Government, will that developer be contracted to maintain/upgrade that software for the life of the software, often exceeding 15 years? If not, will the new contractor selected for maintenance be held accountable for the software obtained from the former prime's reuse library? Will the former prime contractor give the new contractor (or the Government) the source code? (Schimsky, 1992)

Case studies of organizations reusing software emphasize the same requirement: for an organization to attain high levels of reuse, the entire organization must be oriented to the goal of reuse (Incorvaia and Davis, 1990; Hooper and Chester, 1990; Banker et al., 1993). Economic models of software reuse help in making managerial decisions. The next chapter presents economic modeling of software reuse.

III. ECONOMIC MODELING OF SOFTWARE REUSE

Software reuse is a complex process with many technical and managerial issues. Deciding "where and how resources should be invested" is an important part of this process (Bollinger and Pfleeger, 1990). Economic models are important tools for a software producing organization or a Government software manager for making decisions about software reuse. Models are used to better understand processes and provide a way to experiment with relationships among variables. This thesis is based upon the Dynamica Reuse Model, a computer program that simulates the process of software reuse in a large software development organization (Abdel-Hamid, 1993). First, however, two popular models by Gaffney and his colleagues are presented for comparison.

A. ANALYTICAL MODELS OF REUSE ECONOMICS

Gaffney's 1989 model uses three variables in a simple linear relationship. *Productivity, P,* is defined to be the inverse of the cost of software development relative to all-new code, or

- $P = \frac{1}{C} = \frac{1}{R(b-1)+1}$
- C =cost of software development relative to all new code ($C \leq 1$),
- R =proportion of reused code ($R \leq 1$),
- b =the cost relative to that for all-new components of incorporating the reused component into the new product ($b \leq 1$) (Gaffney and Durek, 1989).

In 1991, the model was expanded to include the prorated cost of domain engineering (($(C_{DE}/N)R$), the cost to develop new code ((C_{VN} (1 - R)), and the cost of reusing code (C_{VR}/R) in calculating the unit cost (C_{US}) of the application system, where

- $C_{US} = (C_{DE}/N)R + C_{VN}(1 - R) + C_{VR}R$ = (LH/SLOC or labor/hours per source line of code), and,
- C_{US} = unit cost of the application system,
- C_{DE} = unit cost of domain engineering,
- C_{VN} = unit cost of new code developed for this application system,
- C_{VR} = unit cost of reusing code from the reuse library in this application system (with no modification of code during incorporation),
- N = number of application systems,
- R = proportion of code that is reused code (Cruickshank and Gaffney, 1991; Gaffney and Cruickshank, 1992).

B. THE DYNAMICA REUSE MODEL

The model used in this thesis, the Dynamica Reuse Model, proposes a completely different approach to analyzing the economics of software reuse. The Dynamica Reuse Model is a computer program that simulates a software development organization that practices organization-wide software reuse. (Abdel-Hamid, 1993)

The model has three important characteristics that differentiate it from models discussed thus far. First, the model <u>integrates</u> the complex functions required for organization-wide software reuse discussed in Chapter II. It integrates both the technical issues such as reusable component production, classification, storage, identification, and consumption, and the managerial issues such as setting reuse production and consumption policy and goals. Second, the model uses the <u>feedback</u> principles of system dynamics to better understand the complex system of organizational software reuse. Feedback occurs when "an action taken by a person or thing will eventually affect that person or thing. Circular feedback processes are universal in social systems, the software engineering domain being no exception." Third, the model uses computer <u>simulation</u> to handle over 200 difference equations integrating hundreds of variables relating to technical and managerial issues in organization-wide software reuse. Computer simulation enables controlled experimentation. Unlike the simple linear equations discussed previously, that provide only static values, the Dynamica Reuse Model can simulate dynamic behavior over time. (Abdel-Hamid, 1993)

1. Overview of Model Structure and Behavior

The model is composed of five major sectors. The technical issues discussed in Chapter II are found in Sectors 2 and 3. The managerial issues discussed in Chapter II are found in Sectors 1, 4, and 5. What is important to remember is that each one of these sectors affects and is affected by each one of the other sectors.

a. Software Development and Maintenance

The function of this sector is to provide broad policy on software production for the entire organization, such as the project portfolio size, the average project size, software type, maintenance backlog, etc. This section defines the overall organizational setting. (Abdel-Hamid, 1993)

b. Reusable Component Production

This sector models software reuse activitities associated with production of reusable components, such as domain analysis and the "degree of functional overlap between applications in the domain." Activities associated with the reusable component repository include reusable component creation, classification, and storage. (Abdel-Hamid, 1993)

)

c. Reusable Component Consumption

The activities modeled in this sector include searching for and retrieving components, understanding and assessing components, and adapting and composing components. Some of the factors affecting the activities in this sector include repository size, perceived benefits of reuse, reuse support, overlap between applications, and schedule pressure. (Abdel-Hamid, 1993)

d. Human Resource

The size and characteristics of the organization's personnel are affected by such activities as the "hiring and firing of staff, staff resource allocation, training and turnover." All these activities are captured in the Human Resources sector. (Abdel-Hamid, 1993)

e. Management Policy

Management policy interventions include setting reuse goals, allocating resources, organization, etc. These interventions are management's leverage points to affect the software reuse process in the organization. (Abdel-Hamid, 1993)

C. LONG-TERM VERSUS SHORT-TERM MODELING

A unique function of the Dynamica Reuse Model is that it provides the means for long-term modeling of the software reuse process in a large software producing organization. Unlike the static values obtained from economic models described earlier in this chapter, which only provide a snapshot in time, the model presented here simulates the dynamic behavior of an organization over many years.

Why is this so important in studying and understanding the software reuse process? Due to the large initial investment in domain analysis, reusable component production, and repository deposition and maintenance, the organization must be able to recoup this investment through subsequent reuse of components from the repository. This cost amortization takes place over a long time span of many projects. Thus the software reuse process is best modeled and analyzed over a long-term versus a short-term basis. The significance of this point is important for both managers of software producing organizations and for government contractors who contract for government software projects with these organizations.

D. MACRO VERSUS MICRO MODELING

A second unique function of the Dynamica Reuse Model is that it provides for comprehensive modeling of a complex social process. Large software producing organizations can employ over a thousand people and have projects with over a million lines of code. Coordinating the reuse process in such a large organization over a long period of time is complex and difficult for even the most skilled and experienced managers. Studies have shown that managers have difficulty in predicting the consequences of actions, "especially when cause and effect are distant in time and space." The software reuse process is a complex socio-technical system, as discussed in Chapter II. Unlike the simple linear equations in the economic models discussed in Chapter III, a model such as the Dynamica Reuse Model provides a method for managers and researchers to "reliably and efficiently trace through time the implications of a complex web of system interactions." (Abdel-Hamid, 1993)

The mechanism by which the Dynamica Reuse Model achieves these two unique functions is its "complex network of interconnected feedback loops." By using both positive and negative feedback loops, this model avoids the fallacy of linear-type interactions, in which rates such as reuse production and consumption constantly increase or decrease. Because of the balance achieved between the positive and negative feedback loops, both rapidly changing dynamic states and stable steady states can be achieved just as occur in real life socio-technical systems. (Abdel-Hamid, 1993)

E. LIMITATIONS

It should be noted here that while the Dynamica Reuse Model was thoroughly tested (e.g., dimensional consistency, extreme conditions, reference mode replication), its accuracy in replicating reuse patterns in real organizations has not been evaluated yet.

IV. DYNAMICA REUSE MODEL SIMULATION RESULTS

A. INTRODUCTION

All computer simulations of a software development organization were run using a program called Dynamica Reuse Model. This software program, developed by Dr. Tarek Abdel-Hamid, was installed and run on a 486-33DX computer. Most of the simulations used the Reuse5 Model of Dynamica; however, some used the later Reuse6 Model.

Appendix A presents a complete list of variables studied in the simulations. For the beginning of each simulation, all variables remained at the default values except the particular variable being studied and another variable called *Application Overlap* (NOMOVL), which was increased from 60% to 80% overlap. This variable defines a nominal degree of overlap between the simulated organization's software systems. For example, applications developed from similar domains will have a higher degree of overlap than applications developed from diverse domains.

Simulations typically were run for either ten years or twenty years. To examine relationships between variables, plots were made using data extracted from simulations and tabulated by the Dynamica Reuse Model program. Two methods were used to extract data, depending on whether a dynamic state or a steady state was being simulated.

1. Dynamic State Simulation

Ten-year simulations represent a system in a <u>dynamic state</u>. For this paper, the term "dynamic state" simply refers to systems not at equilibrium. A nominal ten-year

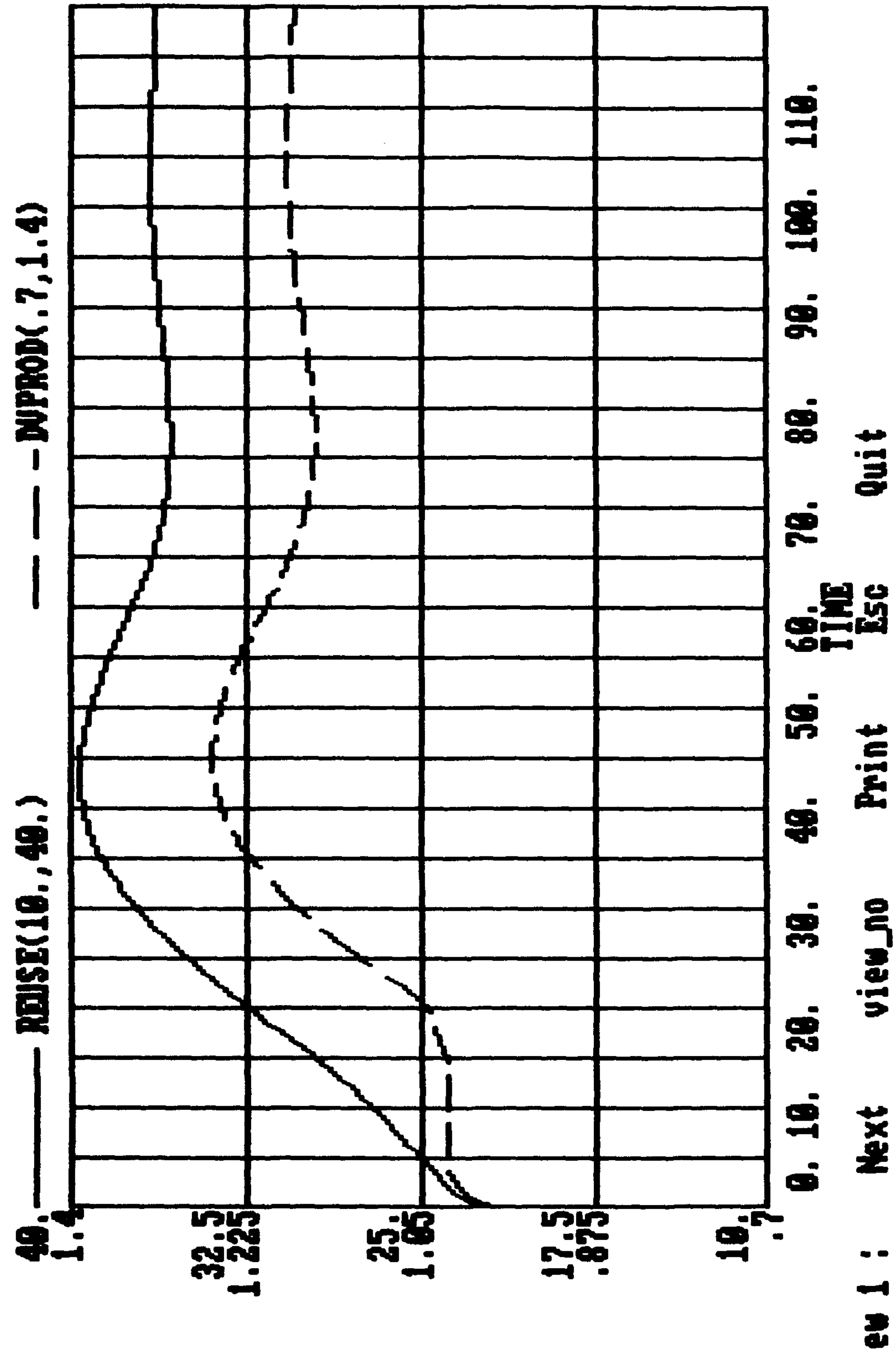


FIGURE 1: Nominal Dynamic State Simulation

simulation is presented in Figure 1. The left vertical axis serves as the axis for two variables. It is scaled from 10% to 40% for the variable *Reuse Rate* (REUSE) and from 0.7 to 1.4 person-months per component for the variable *Productivity* (DVPROD). The horizontal axis measures time in months. For simulating a software producing organization in a dynamic state, ten-year simulations were run. For further analysis, data was tabulated at one-year intervals only. Each ten-year tabulation produced 11 data points (including the value at year zero).

2. Steady State Simulation

Twenty-year simulations represent a system attaining equilibrium or steady state. The term "steady state" refers to a system at equilibrium. A nominal 20 year simulation is presented in Figure 2. For simulating a software producing organization in a steady state, 20 year simulations were run, with data taken only once at the end of the simulation, when the system was at steady state. Each 20 year simulation produced only one data point.

Appendices B through F present the Dynamica Reuse Model simulation graphs and tabular data from which all Dynamica Reuse Model plots were produced. Sorted tabular data used to make graphs comparing variable relationships are included in Tables 1 through 11. To more clearly compare the Dynamica Reuse Model results with literature models, Figures 4 and 7 are copies of graphs from published articles.

The remainder of this chapter is devoted to presentations of simulations and discussions investigating the impact of the following variables on software reuse economics: *Productivity, Reuse Rate, Unit Cost, Consumption Cost, Production Cost, Average*

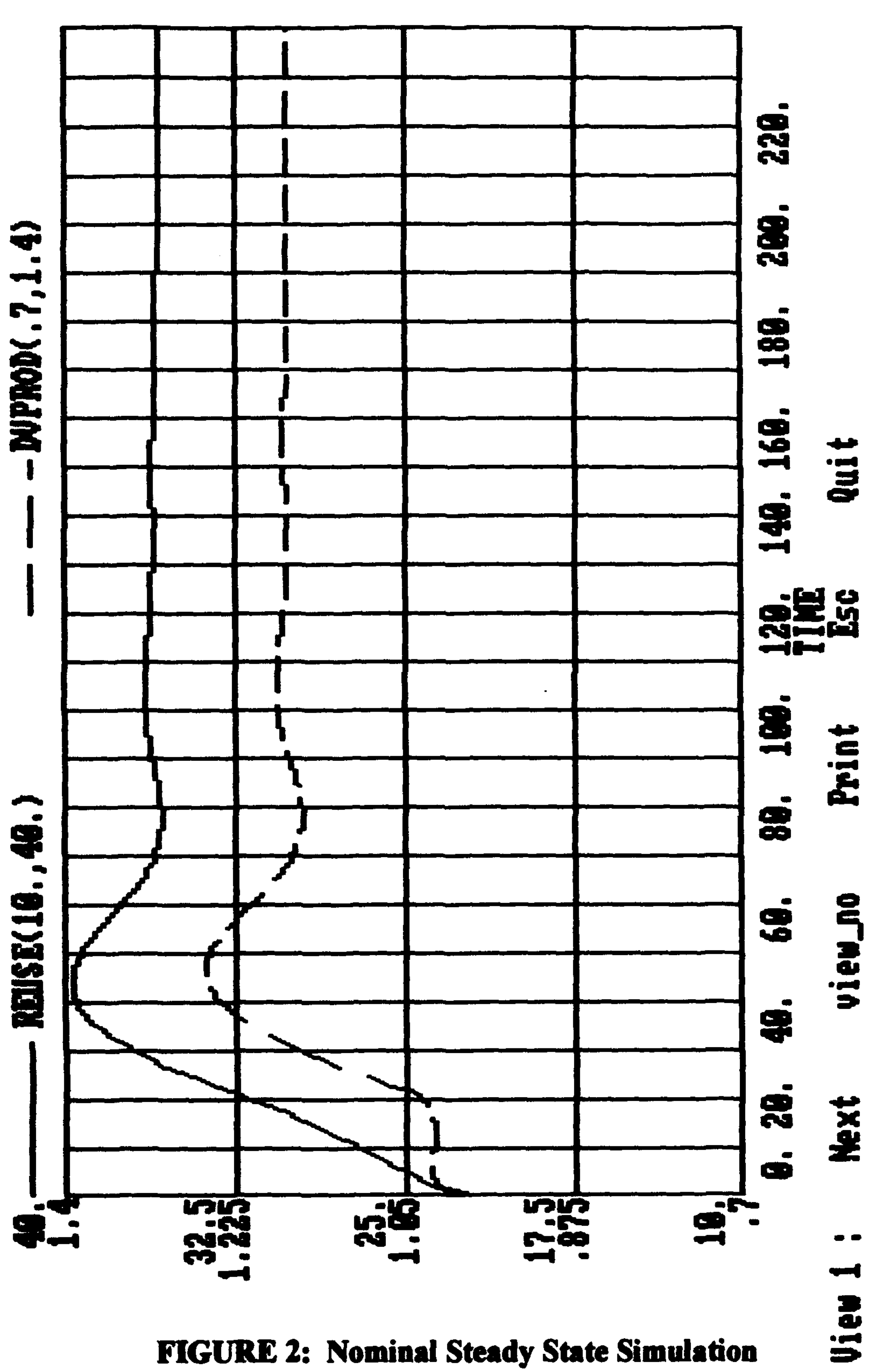


FIGURE 2: Nominal Steady State Simulation

Employment and *Reusable Component Retirement Age*. The first three variables studied, *Productivity*, *Reuse Rate*, and *Unit Cost*, may be considered "dependent" variables only in the sense that they were never varied in order to study their effects on the system. The last four variables studied, *Consumption Cost*, *Production Cost*, *Average Employment*, and *Reusable Component Retirement Age*, may be considered "independent" variables only in the sense that they were varied in order to study their effects on the system.

In reality, since this is a dynamically interacting system, all of the variables are inter-related, just as in real-life socio-technical systems. During preliminary studies for this thesis, certain variables were found to be sensitive to *Reuse Rate* and *Productivity*. In order to study the economics of reuse in such a system, this thesis concentrates on those particular variables. Other variables such as *Reuse Goals*, *Experienced Employees Quit Rate*, etc., are outside the scope of this thesis and are the focus of another thesis.

B. UNIT COST

1. Variables Studied: *Unit Cost*, *Productivity* and *Reuse Rate*

Unit Cost (DVPMPC) and *Productivity* (DVPROD) are two very closely related variables. Each is the inverse of each other. *Unit Cost* is the average cost to develop a software component, and is expressed as person-months per component. As discussed in Chapters I and II, applications produced using reusable components typically cost less than those produced from all-new components.

Productivity is a measure of how productive the software development process is, and is expressed as components per person-month. As previously discussed, production of applications using reusable components should improve productivity rates.

Reuse Rate (REUSE) is the rate at which components are reused, and is expressed as the number of reusable components used divided by the total number of components, i.e., both reusable and new components.

2. Dynamic State Simulation Results

The Dynamica Reuse5 Model program was run for a ten-year simulation period. All variables were initially set at default values. Values of three variables, *Unit Cost* (DVPMPC), *Productivity* (DVPROD), and *Reuse Rate* (REUSE), were taken at one-year data intervals and are presented in Table 1 and plotted in Figure 3. Each data point represents the value of a one-year interval of the ten-year simulation, i.e., a single ten-year simulation run provides eleven data points (including time zero). In Appendix B, the Dynamica Reuse 5 Model simulation graph is presented as Figure B; the tabulated results are presented in Table B.

TABLE 1: DYNAMIC STATE *UNIT COST* AND *PRODUCTIVITY* VERSUS *REUS RATE*

Reuse Rate (REUSE)	*Unit Cost* (DVPMPC)	*Productivity* (DVPROD)
22.60	1.02	0.98
28.08	0.98	1.02
34.62	0.91	1.10
35.96	0.86	1.16
36.01	0.86	1.16
36.45	0.85	1.18
36.55	0.85	1.18
36.65	0.84	1.19
37.46	0.83	1.21
38.98	0.81	1.23
39.46	0.80	1.26

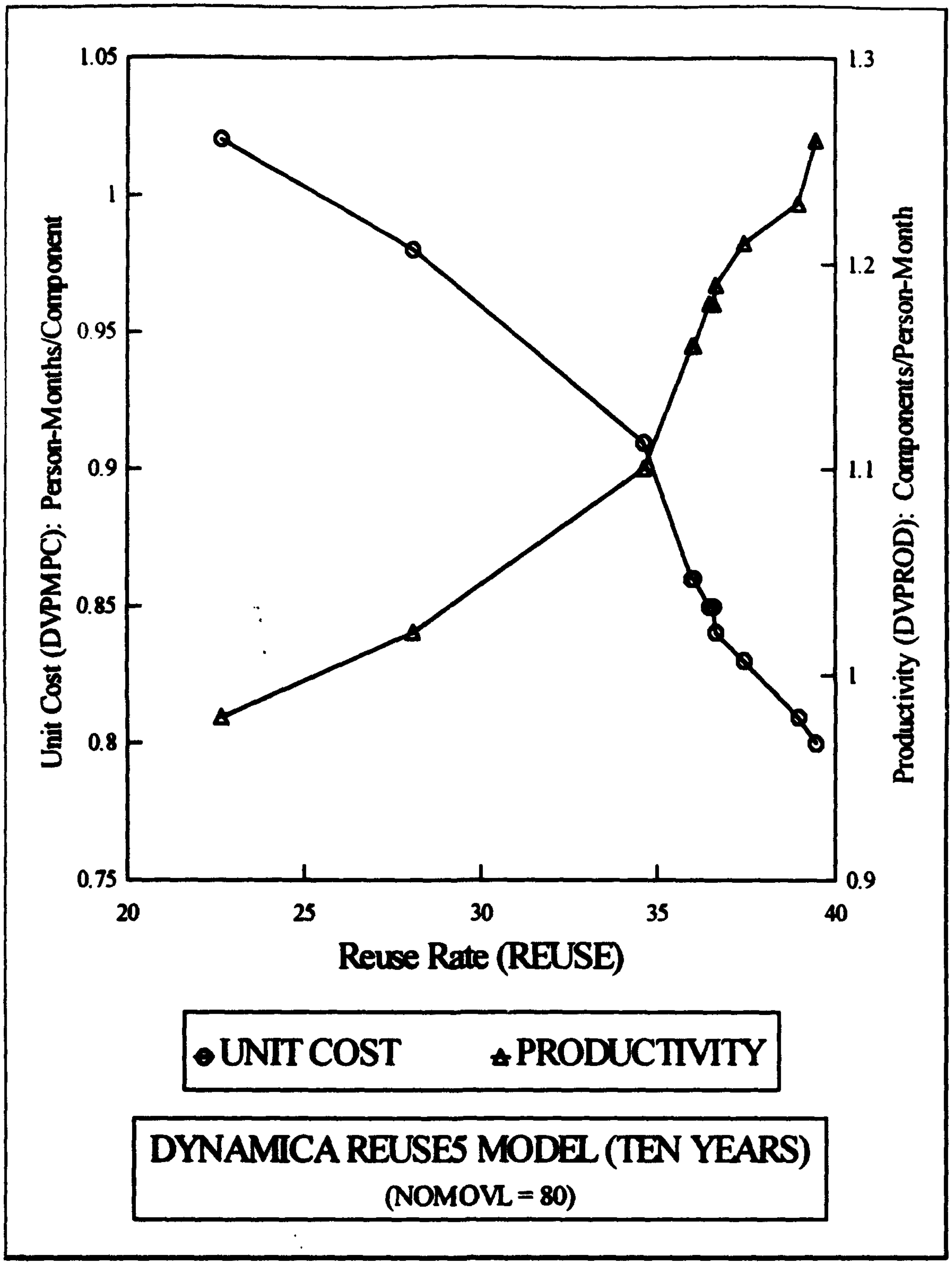


FIGURE 3: Dynamic State *Unit Cost* and *Productivity* versus *Reuse Rate*

The inverse relationship of *Unit Cost* and *Productivity* is illustrated by the mirror imaging of the two lines of Figure 3. Both variables improve with increasing rate of reuse, that is, *Unit Cost* decreases and *Productivity* increases as the *Reuse Rate* increases. Over the ten-year simulation, *Reuse Rate* never gets higher than 40%.

3. Comparison with Literature Results

Cruickshank and Gaffney (1991) present a reuse economics model (discussed in Chapter III and presented in Figure 4) in which total unit cost is calculated as labor-hours per source-line-of-code (LH/SLOC) for an application system composed of new and re-used code. Each data point represents the percentage of code reuse for one of eight technical software applications from the aerospace industry. When total unit cost is plotted against percentage of code reuse (Figure 4), the plot reveals the same basic relationship between unit cost and reuse rate as was demonstrated in Figure 3. In other words, as the percentage of code reuse increases, the unit cost decreases.

Although Cruickshank's and Gaffney's model (Figure 4), shows the same relationship between unit cost and reuse rate as the Dynamica model (Figure 3), there are differences between the two models. In Cruickshank's model, each point represents a value from an individual software application from the aerospace industry. In the Dynamica model (Figure 3), each point represents an average organization-wide value at one-year intervals. In Cruickshank's model, it is unclear if the applications are from different organizations or only one organization, or if the applications are developed in the same language.

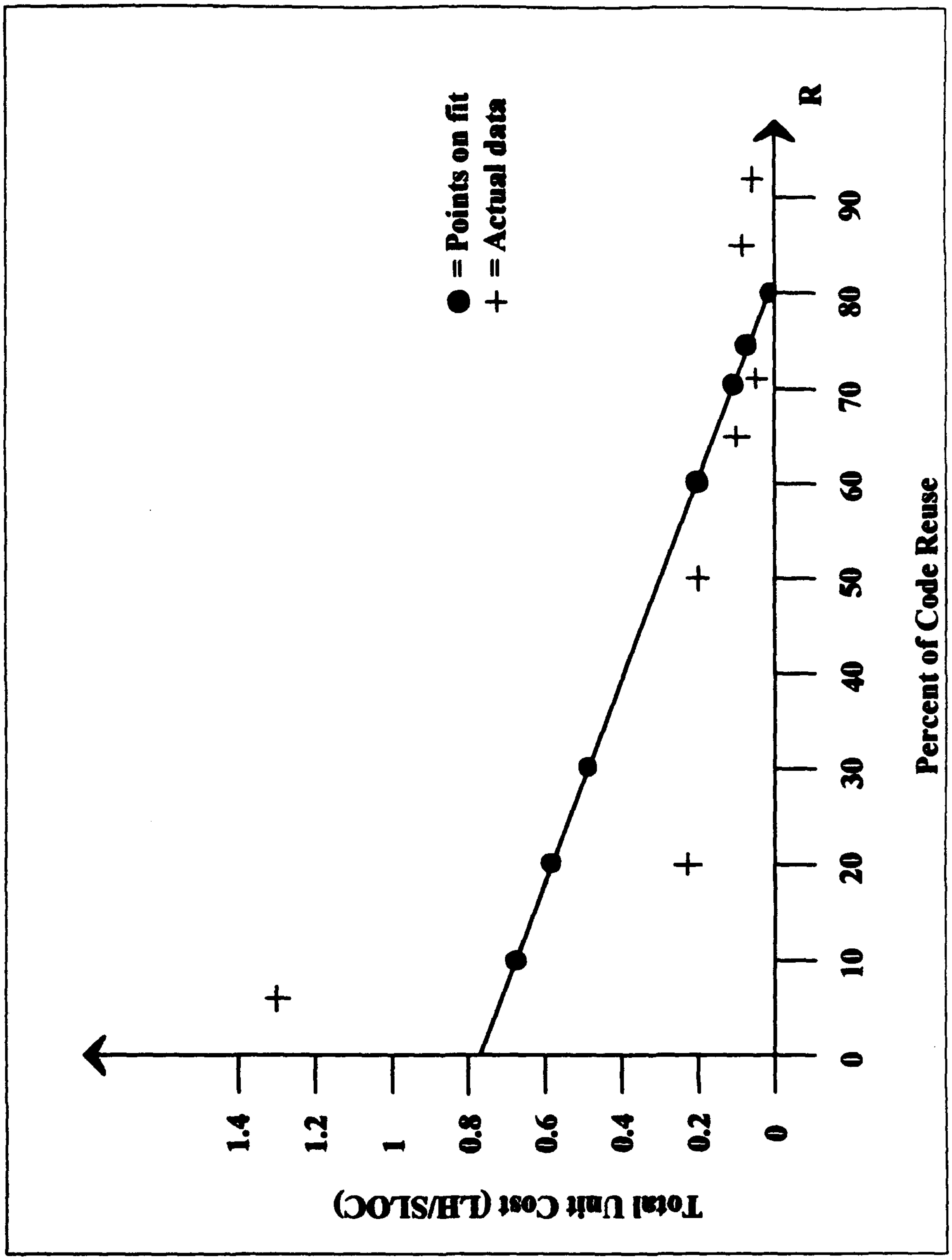


FIGURE 4: Cruickshank's and Gaffney's (1991) Model

In the Dynamica model, each point represents the average unit cost per component for a software producing organization over many applications.

In Cruickshank's model, the unit is an application system; in the Dynamica model, the unit is a component. Cruickshank and Gaffney discuss components, which they call reusable software objects, or RSO; however, none of their data is measured by RSO. Instead, all data is expressed in LOC, or lines-of-code. LOC as a measurement of productivity and unit cost has notable pitfalls. One pitfall is its sensitivity to line counting variations and type of language. As pointed out, all that is known about their applications is that they are technical software applications from the aerospace industry.

In Cruickshank's model, each data point represents the total unit cost for a finished application. In the Dynamica model, each data point represents the average unit cost of many components of many applications in a single organization, obtained from any one of many points in time over a long interval of ten years. Yet even though the time span is much greater, the same inverse relationship between *Unit Cost* and *Reuse Rate* is demonstrated in the Dynamica systems model as is demonstrated in the simpler Cruickshank model.

C. CONSUMPTION COST

1. Variables Studied: *Consumption Cost, Productivity, Reuse Rate, Cumulative Productivity* and *Repository Size*

Consumption Cost (NMFRRU) is the relative cost of reusing a component. It is the nominal fraction of development effort to reuse a component, and is a unitless fraction (development person-months per component to reuse divided by development person-months per component to develop an all-new component). A *Consumption Cost* value of one means the development cost to reuse a component equals the development cost to develop an all-new component. Typically, *Consumption Cost* values are less than one.

Cumulative Productivity (CDVPRD) is the cumulative development productivity expressed in components per person-month. Whereas *Productivity* (DVPROD) is the instantaneous average development productivity at a single point in time, *Cumulative Productivity* is the accumulated development productivity up to the time it is measured. It is the average productivity at time *t*. *Repository Size* (RPSTRY) represents the number of reusable components in the organization's repository. *Productivity* and *Reuse Rate* variables were discussed in Section IV.B.

2. Dynamic State Simulation Results

The Dynamica Reuse5 Model program was run for a series of four ten-year simulations. In each run the value of the variable *Consumption Cost* (NMFRRU) was set to one of the following values: 0.25 0.50, 0.75, and 1.0. Values of three variables, *Consumption Cost* (NMFRRU), *Productivity* (DVPROD), and *Reuse Rate* (REUSE), were taken at one-year data intervals and are presented in two different plots (Figures 5 and 6) in order to emphasize relationships differently. In Appendix C, the Dynamica Reuse5 Model simulation graphs are presented as Figures C1 through C4; the tabulated results are presented as Tables C1 through C4.

TABLE 2: DYNAMIC STATE *PRODUCTIVITY* VERSUS *REUSE RATE*: VARYING *CONSUMPTION COST*				
Reuse Rate (REUSE)	*Productivity* for *Consumption Cost* = 0.25	*Productivity* for *Consumption Cost* = 0.50	*Productivity* for *Consumption Cost* = 0.75	*Productivity* for *Consumption Cost* = 1
22.60	0.98			
28.08	1.02			
34.62	1.10			
35.96	1.16			
36.01	1.16			
36.45	1.18			
36.55	1.18			
36.65	1.19			
37.46	1.21			
38.98	1.23			
39.46	1.26			
22.09		0.91		
27.33		0.92		
34.01		0.97		
36.62		1.04		
36.99		1.05		
36.96		1.05		
36.71		1.05		
37.10		1.06		
37.95		1.08		
30.63		1.08		
39.34		1.10		
17.94			0.85	
19.17			0.83	
25.35			0.86	
31.79			0.93	
34.59			0.94	
34.59			0.94	
34.84			0.94	
34.98			0.94	
35.08			0.95	
35.57			0.96	
35.68			0.96	
10.50				0.82
11.58				0.79
14.14				0.80
17.67				0.83
22.57				0.84
27.74				0.85
31.47				0.86
32.80				0.86
33.03				0.86
33.05				0.86
33.13				0.86

Figures 5 and 6 (and Tables 2 and 3, respectively) present different ways of examining relationships using the same data. In Figure 5, *Productivity* is plotted versus *Reuse Rate* and grouped by *Consumption Cost* (NMFRRU) relative values of 0.25, 0.50, 0.75, and 1.0. In Figure 6, *Productivity* is plotted versus *Consumption Cost* and grouped by *Reuse Rate* (REUSE) values of 10's (less than 20%), 20's (from 20% to less than 30%), and 30's (30% and over).

TABLE 3: DYNAMIC STATE *PRODUCTIVITY* VERSUS *CONSUMPTION COST*: VARYING *CONSUMPTION COST*

Consumption Cost (NMFRRU)	*Productivity* for *Reuse Rate* = 10's	*Productivity* for *Reuse Rate* = 20's	*Productivity* for *Reuse Rate* = 30's
0.25		0.98	1.10
0.25		1.02	1.23
0.25			1.26
0.25			1.21
0.25			1.16
0.25			1.16
0.25			1.18
0.25			1.19
0.25			1.18
0.50		0.91	0.97
0.50		0.92	1.08
0.50			1.10
0.50			1.08
0.50			1.05
0.50			1.04
0.50			1.05
0.50			1.06
0.50			1.05
0.75	0.85	0.86	0.93
0.75	0.83		0.96
0.75			0.96
0.75			0.95
0.75			0.94
0.75			0.94
0.75			0.94
0.75			0.94
1.00	0.82	0.84	0.86
1.00	0.79	0.85	0.86
1.00	0.80		0.86
1.00	0.83		0.86
1.00			0.86

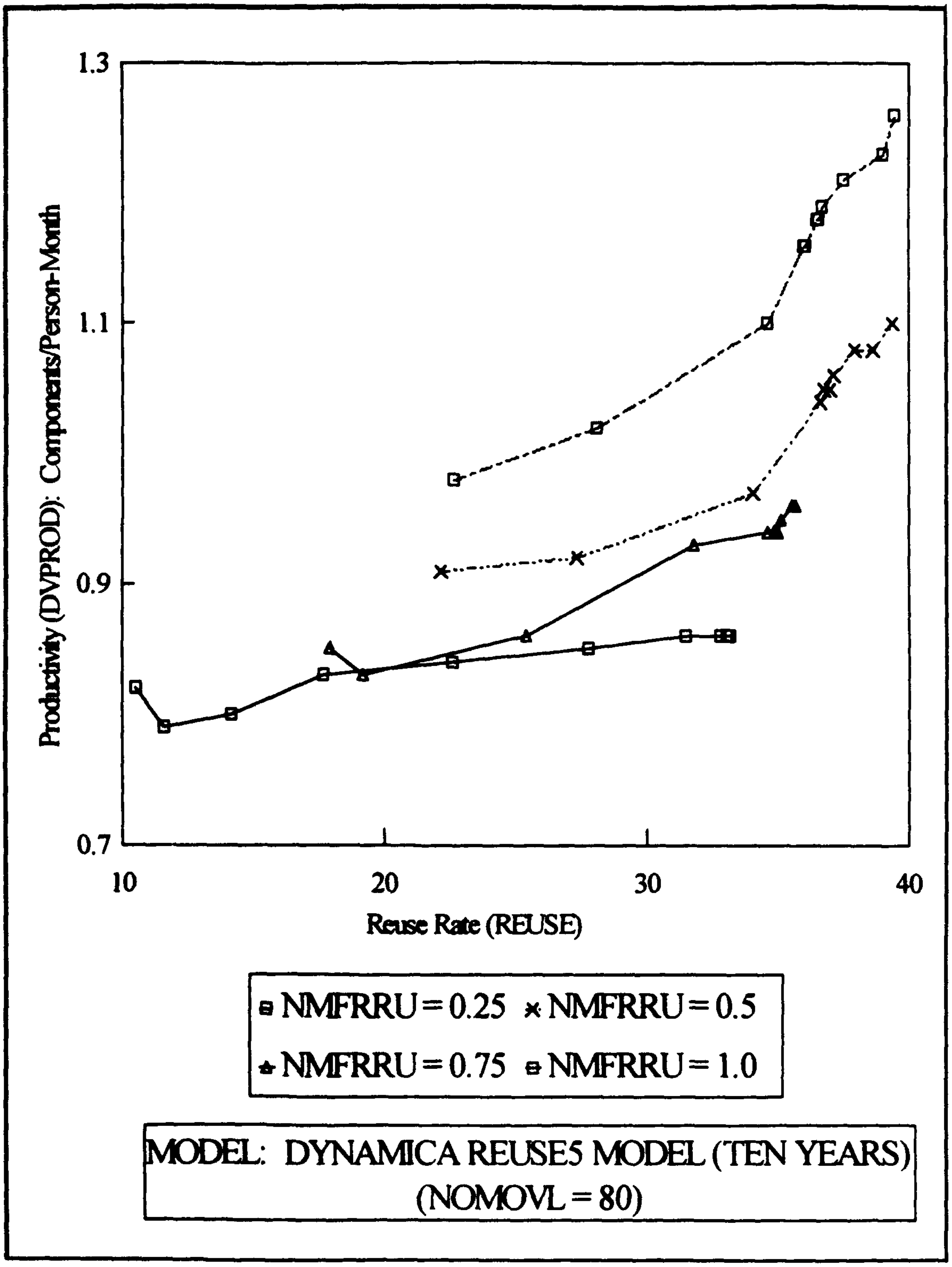


FIGURE 5: Dynamic State *Productivity* versus *Reuse Rate:* varying *Consumption Cost* (NMFRRU)

In Figure 5, each group of points connected by a line represents one entire simulation run. The simulation run with the highest *Productivity* values had the lowest *Consumption Cost* relative value, i.e., 0.25. The simulation run with the lowest *Productivity* values had the highest *Consumption Cost* relative value, i.e., 1.0. In other words, *Productivity* is inversely related to *Consumption Cost.*

All simulation runs have *Reuse Rate* values between 22 percent to about 33 percent. The runs with higher *Reuse Rate* values have lower *Consumption Cost* values; the runs with lower *Reuse Rate* values have higher *Consumption Cost* values. Thus, these results suggest *Reuse Rate* is inversely related to *Consumption Cost.*

To compare these results to those published in the literature, the next figure was plotted following Gaffney's and Durek's (1989) model. The results from each of the four simulation runs are plotted in Figure 6. For each of the four values of the variable NMFRRU (the x-axis variable), *Productivity* values are plotted and grouped in three groups depending on Reuse Rate (each with its symbol). Each data point represents the value at the end of a one-year interval of the ten-year simulation.

As *Consumption Cost* decreases, *Productivity* increases. The greatest amount of variability in *Productivity* occurs when *Consumption Cost* is least, at only 25 percent of the cost to create an all-new component. *Productivity* values converge when *Consumption Cost* is the same as the cost to create an all-new component, i.e., when NMFRRU equals one. The lowest *Reuse Rates* are associated with the lower *Productivity* values and the higher *Consumption Costs*. When *Reuse Rates* are less than 25 percent, *Productivity*

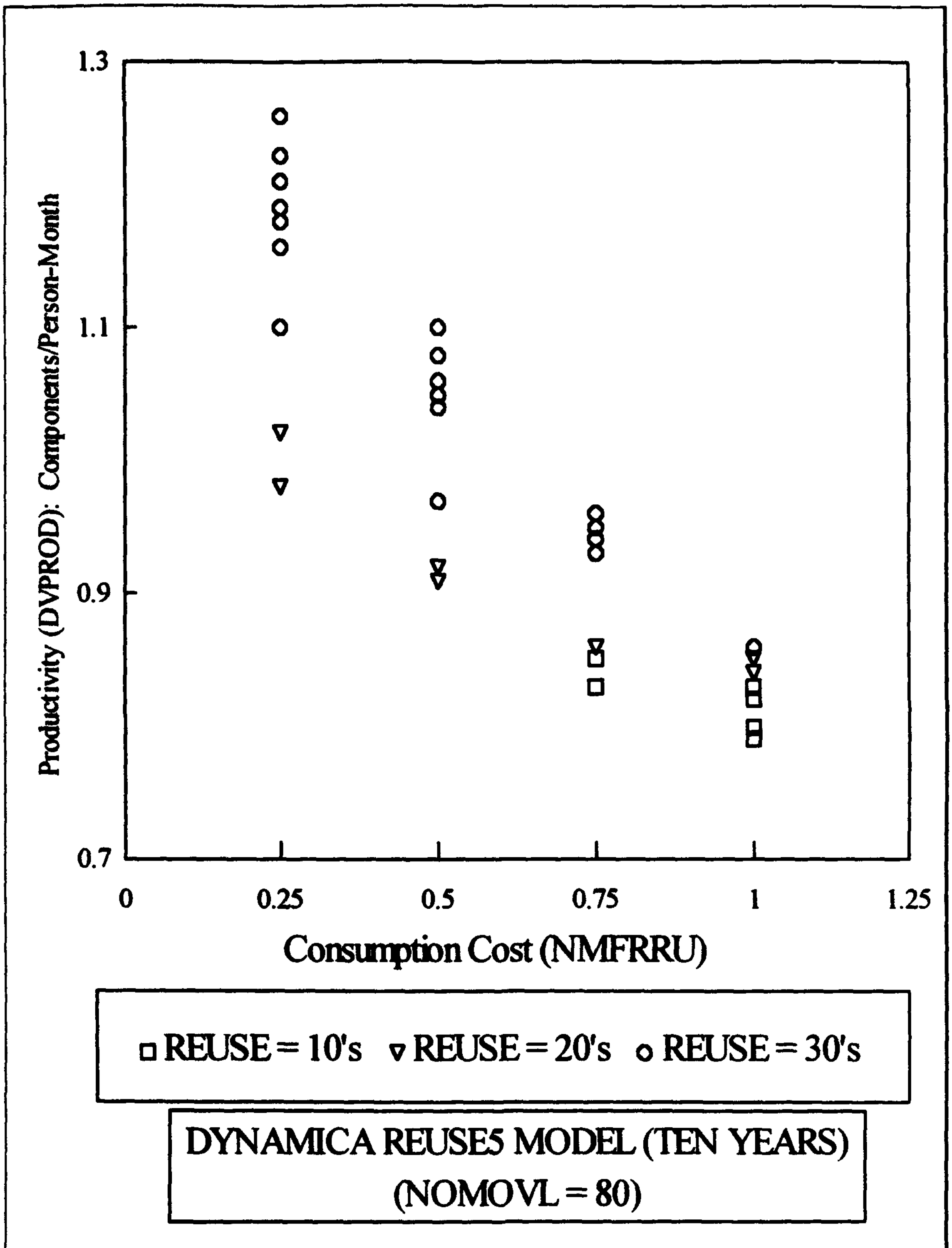


FIGURE 6: Dynamic State *Productivity* versus *Consumption Cost*: varying *Consumption Cost* (NMFRRU)

never rises above 0.85 components per person-month. When *Reuse Rates* are less than 30 percent, *Productivity* never rises above 1.02 components per person-month. *Reuse Rates* above 30 percent reach *Productivity* values of 1.26 components per person-month.

3. Comparison to Literature Results

Gaffney and Durek (1989) present a simple reuse model (discussed in Chapter III and presented in Figure 7) in which the cost of reusing software components is presented relative to the cost of all-new code. As in the Dynamica Reuse Model, the components being integrated are <u>reusable</u> components, that is, components <u>designed</u> to be reusable. Both Gaffney's model (Figure 7) and the Dynamica Reuse Model (Figure 6) define *Consumption Cost* (or Relative Cost to Reuse, in Gaffney's model) to be <u>relative</u> to the cost of developing an all-new component, and as such is less than or equal to one. *Reuse Rate* is defined in both models as the proportion of reused components to the total number of components used, and is defined to be less than or equal to one. However, Gaffney's model defines *Productivity* to be <u>relative</u> to software productivity for <u>all-new</u> code, whereas the Dynamica Reuse Model defines *Productivity* as an <u>absolute</u> value of components per person-month.

Gaffney's simple reuse model shows the same basic relationships as seen in Figure 6. However, in Gaffney's model, *Reuse Rate* values used are 0.2, 0.5, and 0.8, a range much larger than the *Reuse Rates* seen in the ten-year Dynamica Reuse Model simulations (from 10.5 to 39.34).

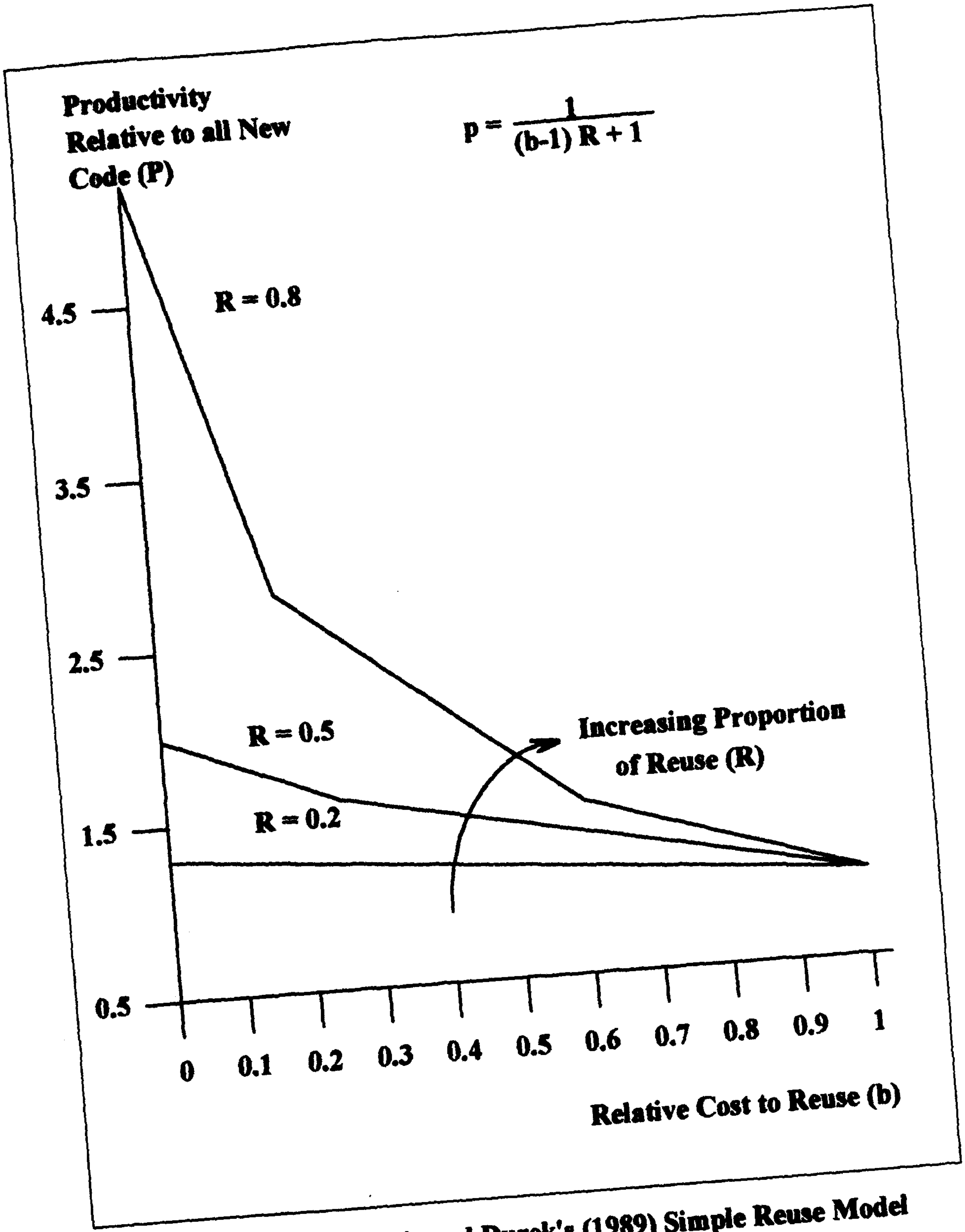


FIGURE 7: Gaffney's and Durek's (1989) Simple Reuse Model

Both models define software components as more than just code. Gaffney considers software components to be composed of three main types or levels: code (lowest and simplest level), design (intermediate in complexity), and requirements (most complex level). This is consistent with the concept used in the Dynamica Reuse Model.

4. Steady State Simulation Results

In addition to modeling transient relationships between *Productivity* and *Consumption Cost*, long-term steady state relationships were studied. Rather than take "snapshots" every year of a ten-year simulation, the Dynamica Reuse6 Model program was run for a series of 20 year simulations and values of variables taken at the end of each 20 year simulation. That is, data points were taken only when the software producing organization was at steady state.

TABLE 4: STEADY STATE *REUSE RATE* AND *PRODUCTIVITIES* VERSUS *CONSUMPTION COST*: VARYING *CONSUMPTION COST*

Reuse Rate (REUSE)	*Consumption Cost* (NMFRRU)	*Instantaneous Productivity* (DVPROD)	*Cumulative Productivity* (CDVPRD)
35.58	0.10	1.26	1.18
35.80	0.20	1.20	1.12
36.02	0.30	1.15	1.07
36.22	0.40	1.10	1.02
[illegible].43	0.50	1.05	0.98
[illegible].61	0.60	1.01	0.94
[illegible].03	0.70	0.96	0.89
36.86	0.80	0.93	0.86
36.98	0.90	0.90	0.83
33.56	1.00	0.86	0.81

In each of ten 20 year simulations, the variable *Consumption Cost* was changed in increments of 0.1, from an initial value of 0.1 in the first simulation to 1.0 in the tenth simulation. The values of the four variables, *Reuse Rate* (REUSE), *Consumption Cost* (NMFRRU), *Productivity* (DVPROD), and *Cumulative Productivity* (CDVPRD), were taken at the end of each 20 year simulation and are presented in Table 4 and plotted in Figure 8. The ten Dynamica Reuse6 Model simulation graphs are presented as Figures C5 through C14 in Appendix C. The tabulated results are presented as Tables C5 through C14 in Appendix C.

In Figure 8, both *Productivities* were plotted on the right Y-axis. *Cumulative Productivity* is represented by diamonds. *Productivity (Instantaneous)* is represented by triangles. Both *Productivities* have a clearly inverse relationship with *Consumption Cost,* just as *Productivity* (*Instantaneous*) had in the dynamic state simulations of Figures 3, 5, and 6. As the relative cost to consume or reuse components increases and approaches that of an all-new component, i.e., 1.0, both *Productivities* decrease at the same rate.

However, the result for the relationship between the long-term, steady state *Reuse Rate* and *Consumption Cost* is rather surprising. The steady state *Reuse Rate* does not have the same inverse relationship with *Consumption Cost* as in previous figures from dynamic state simulations. Rather, it stays almost constant.

The surprising independence of the steady state *Reuse Rate* and *Consumption Cost* demonstrated in Figure 8 can be best explained by examining Figure 9, which compares the results from two simulation runs in the same graph. In the two simulation runs, the value for *Consumption Cost* was set to be either 0.1 or 0.9. The plotted results show the variables *Reuse Rate* (REUSE) and *Cumulative Productivity* (CDVPRD) for the two runs.

At the end of the 20 year simulation runs, the two *Reuse Rates* converge at approximately the 36 percent level. An interesting note is that although the final steady state

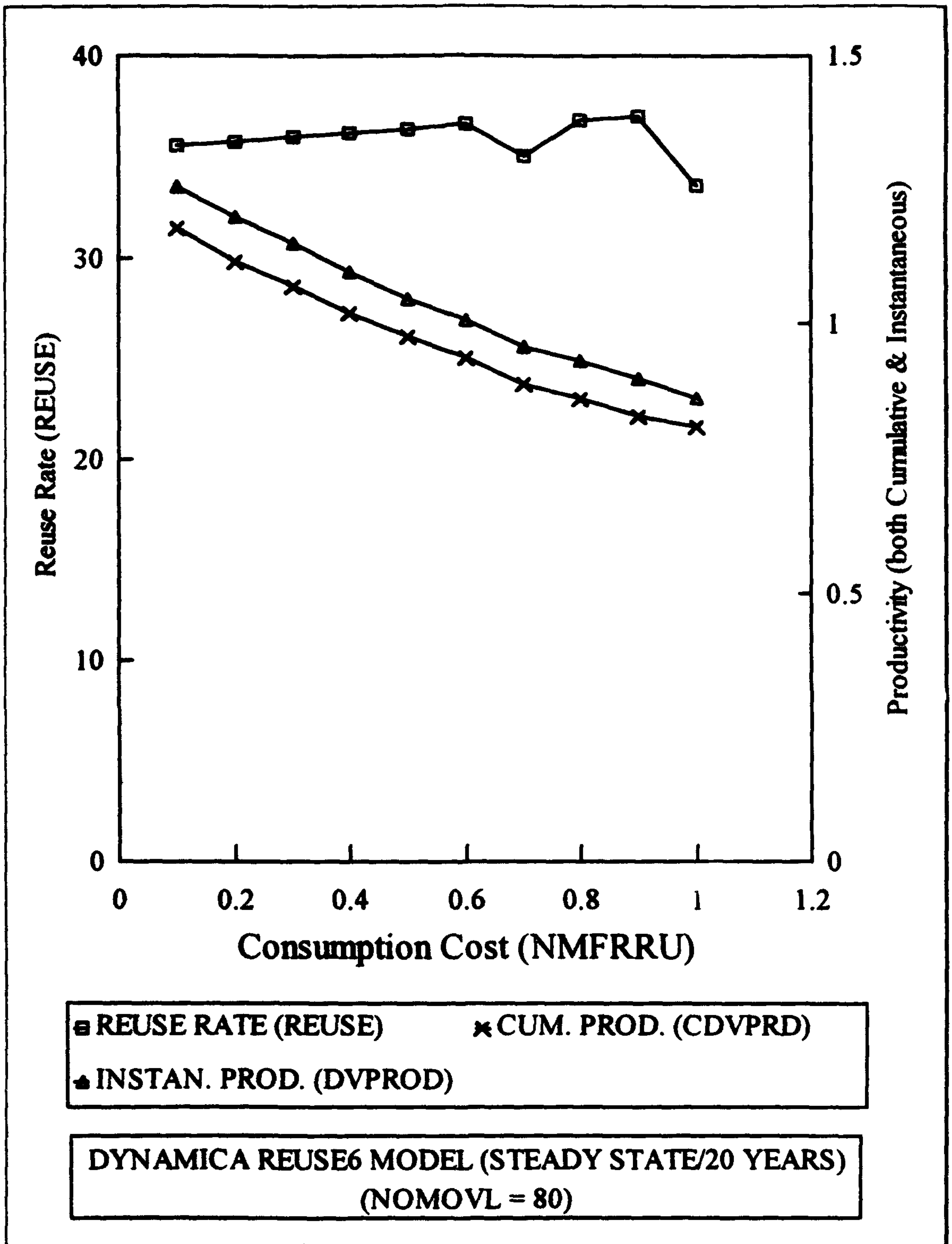


FIGURE 8: Steady State *Reuse Rate* and *Productivities* versus *Consumption Cost*: varying *Consumption Cost* (NMFRRU)

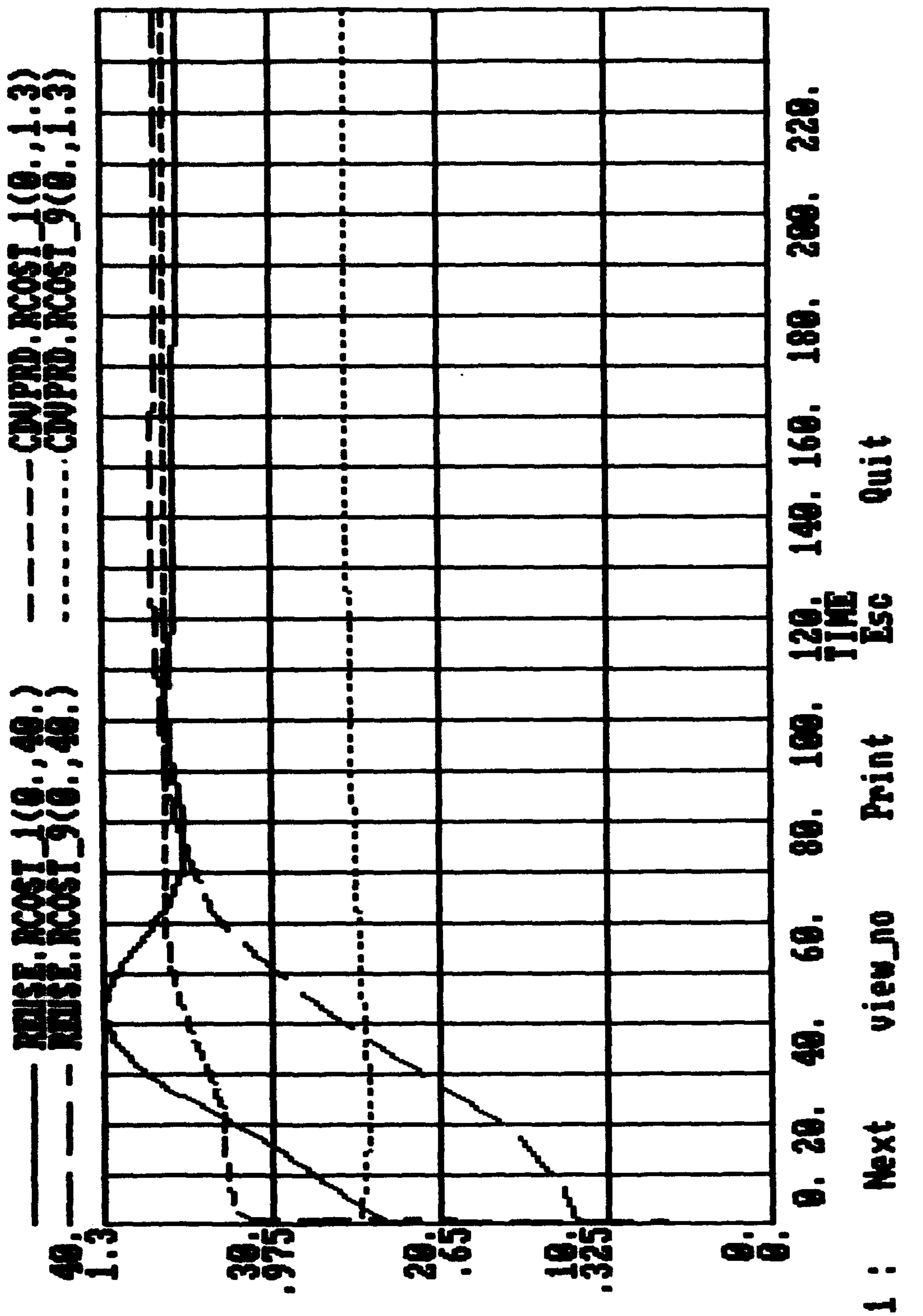


FIGURE 9: Steady State *Reuse Rate* and *Cumulative Productivity* at very high and low *Consumption Cost*

values are very close, their dynamic profiles early in the lifecycle are quite different. The *Reuse Rate* curve for *Consumption Cost* equal to 0.1 is cyclic, while the *Reuse Rate* curve for *Consumption Cost* equal to 0.9 increases at a nearly constant rate to steady state. Because in both cases *Consumption Cost* is less than development cost (i.e., less than 1), these results suggest that as long as *Consumption Cost* is not greater than development cost, over the long run the organization will achieve a steady state reuse level, which can be called "The Organization Reuse Potential."

Finally, note that the two lines representing *Cumulative Productivity* have steady state values of 1.18 and 0.83 (for *Consumption Cost* equal to 0.1 and 0.9, respectively). This represents a drop of nearly 30 percent in *Cumulative Productivity* at the higher *Consumption Cost* value of 0.9. Because *Productivity* is inversely related to *Unit Cost*, *Consumption Cost* is an indicator of the economic benefits of the software reuse process, that is, as *Consumption Cost* decreases, *Productivity* increases, and *Unit Cost* decreases (and vice versa). Correspondingly, *Reuse Rate* is not an indicator of the economic benefits of software reuse in the long-term.

D. PRODUCTION COST

1. Variables Studied: *Production Cost, Productivity* and *Reuse Rate*

Production Cost (NMEXTR) is the relative cost to produce a reusable component. It is the nominal fraction of development effort to produce a reusable component, and is a unitless number (development person-months per component to develop a reusable component, divided by development person-months per component to develop a non-reusable component). A *Production Cost* value of one means that the development cost to develop a reusable component equals the development cost to develop a non-reusable component. Typically, *Production Cost* values are greater than one. *Productivity* and *Reuse Rate* were discussed in Section IV.B.

2. Dynamic State Simulation Results

The Dynamica Reuse5 Model program was run for a series of five ten-year simulations. In each run the value of the variable *Production Cost* (NMEXTR) was set to one of the following values: 1, 2, 3, 4, and 5. Values of three variables, *Production Cost* (NMEXTR), *Productivity* (DVPROD), and *Reuse Rate* (REUSE), were taken at one-year intervals and are presented in two different plots (Figures 10 and 11) in order to emphasize relationships differently. In Appendix D, the Dynamica Reuse5 Model simulation graphs are presented as Figures D1 through D5; the tabulated results are presented as Tables D1 through D5.

Figures 10 and 11 (and corresponding Tables 5 and 6) present different ways of examining relationships using the same data. In Figure 10, *Productivity* is plotted versus *Reuse Rate* and grouped by *Production Cost* (NMEXTR) relative values of 1, 2, 3, 4, and 5. In Figure 11, *Productivity* is plotted versus *Production Cost* and grouped by *Reuse Rate* (REUSE) values of <10 (less than 10%), 10's (from 10 to less than 20%), 20's (from 20 to less than 30%), and 30's (30% and over).

TABLE 5: DYNAMIC STATE *PRODUCTIVITY* VERSUS *REUSE RATE*: VARYING *PRODUCTION COST*					
Reuse Rate (REUSE)	*Productivity* for *Production Cost* = 1	*Productivity* for *Production Cost* = 2	*Productivity* for *Production Cost* = 3	*Productivity* for *Production Cost* = 4	*Productivity* for *Production Cost* = 5
22.60	1.03				
35.34	1.23				
35.67	1.24				
36.08	1.24				
36.09	1.26				
36.15	1.25				
36.49	1.25				
36.50	1.27				
38.35	1.33				
39.27	1.29				
22.60		0.98			
28.08		1.02			
34.62		1.10			
35.96		1.16			
36.01		1.16			
36.45		1.18			
36.55		1.18			
36.65		1.19			
37.46		1.21			
38.98		1.23			
39.46		1.26			
22.60			0.96		
24.89			0.95		
28.42			0.95		
32.53			1.00		
36.14			1.09		
36.23			1.09		
36.37			1.10		
36.42			1.10		
36.53			1.10		
36.71			1.11		
37.29			1.12		
15.79				0.95	
17.23				0.96	
18.69				0.96	
19.99				0.96	
21.15				0.96	
22.21				0.96	
22.59				0.95	
22.60				0.95	
22.83				0.95	
22.98				0.97	
23.13				0.96	
9.77					0.90
10.48					0.91
11.31					0.91
12.25					0.92
13.37					0.92
14.81					0.94
16.60					0.97
18.46					0.98
20.04					0.96
21.44					0.96
22.60					0.94

TABLE 6: DYNAMIC STATE *PRODUCTIVITY* VERSUS *PRODUCTION COST*: VARYING *PRODUCTION COST*				
Production Cost (NMEXTR)	*Productivity* for *Reuse Rate* < 10	*Productivity* for *Reuse Rate* = 10's	*Productivity* for *Reuse Rate* = 20's	*Productivity* for *Reuse Rate* = 30's
1.00			1.03	1.26
1.00				1.29
1.00				1.32
1.00				1.33
1.00				1.27
1.00				1.24
1.00				1.24
1.00				1.25
1.00				1.25
1.00				1.23
2.00			0.98	1.10
2.00			1.02	1.23
2.00				1.26
2.00				1.21
2.00				1.16
2.00				1.16
2.00				1.18
2.00				1.18
3.00			0.95	1.00
3.00			0.96	1.09
3.00			0.96	1.12
3.00				1.11
3.00				1.09
3.00				1.10
3.00				1.10
3.00				1.10
4.00		0.96	0.95	
4.00		0.96	0.95	
4.00		0.96	0.95	
4.00		0.96	0.96	
4.00			0.97	
4.00			0.96	
4.00			0.96	
5.00	0.90	0.98	0.94	
5.00		0.97	0.96	
5.00		0.94	0.96	
5.00		0.92		
5.00		0.92		
5.00		0.91		
5.00		0.91		

In Figure 10, each group of points connected by a line represents one entire simulation run. The five simulations can be divided into two groups, based on the relationships between the variables. In the first group, for *Production Cost* values of 1, 2, or 3, *Productivity* is generally directly related to *Reuse Rate* values, that is, as *Reuse Rate* increases, so does *Productivity*. In addition, as *Production Cost* increases from 1 to 3, the values for *Productivity* decrease for equal *Reuse Rate* values.

In contrast, the second group of simulation runs with *Production Costs* of 4 or 5 have *Reuse Rates* and *Productivities* that are generally lower than the lowest corresponding values for the runs with *Production Costs* from 1 to 3. The two runs with higher *Production Costs* have *Reuse Rates* ranging from 9.77 to 23.13; the three runs with the lower *Production Costs* have *Reuse Rates* ranging from 22.6 to 39.46. Likewise, the two runs with higher *Production Costs* have *Productivities* ranging from 0.9 to 0.98; the three runs with the lower *Production Costs* have *Productivities* values ranging from 0.95 to 1.33.

In Figure 11, for each of the five values of the variable NMEXTR, *Productivity* values are plotted and grouped in three groups depending on *Reuse Rate* (each with its symbol). Each data point represents the value at the end of a one-year interval of the ten-year simulation.

In general, as *Production Cost* increases, *Productivity* decreases and converges to a narrow range when *Production Cost* is 4. When *Production Cost* is 5, this range spreads again, but still has the lowest *Productivity* values of any of the simulations.

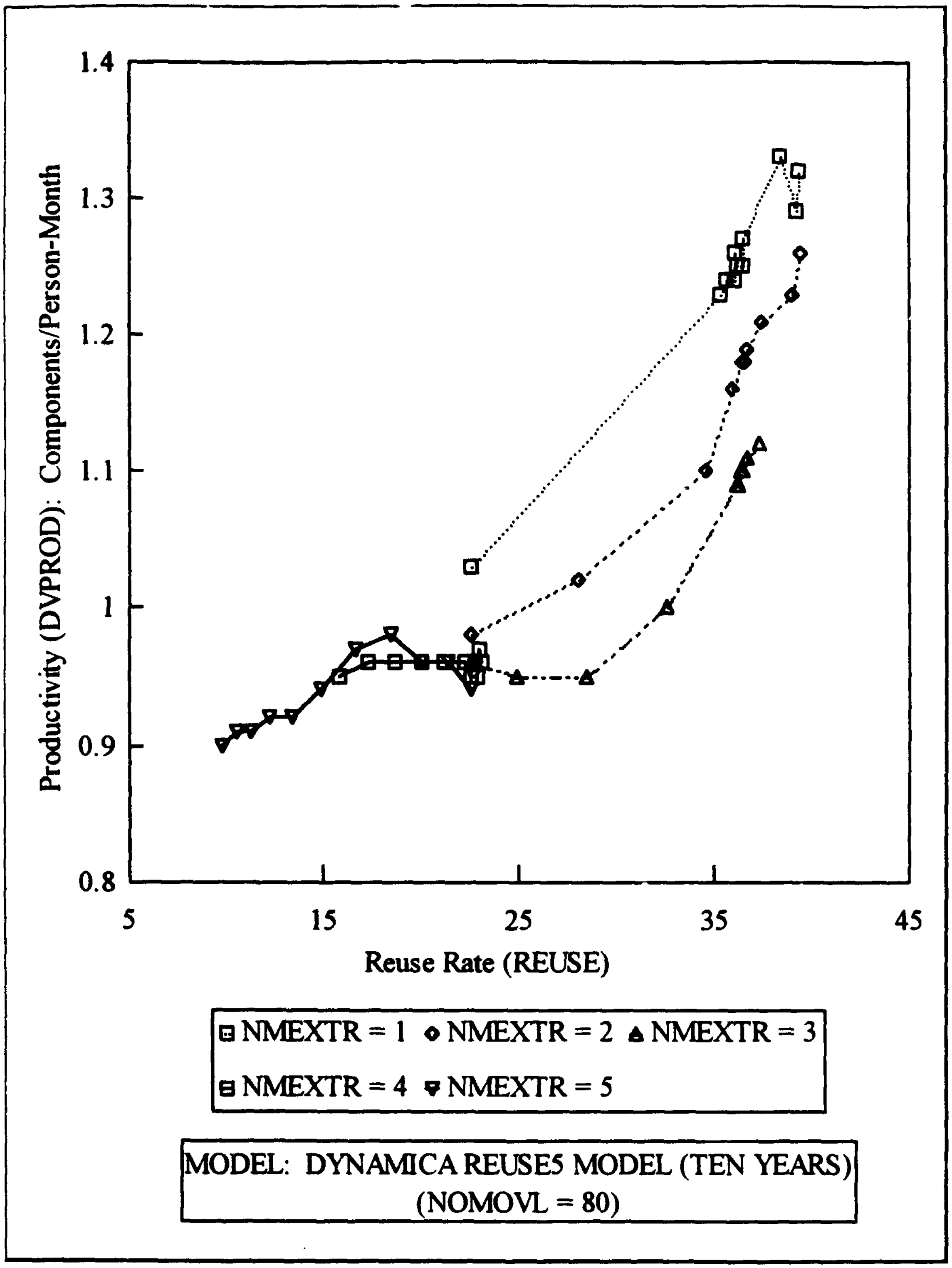


FIGURE 10: Dynamic State *Productivity* versus *Reuse Rate:* varying *Production Cost* (NMEXTR)

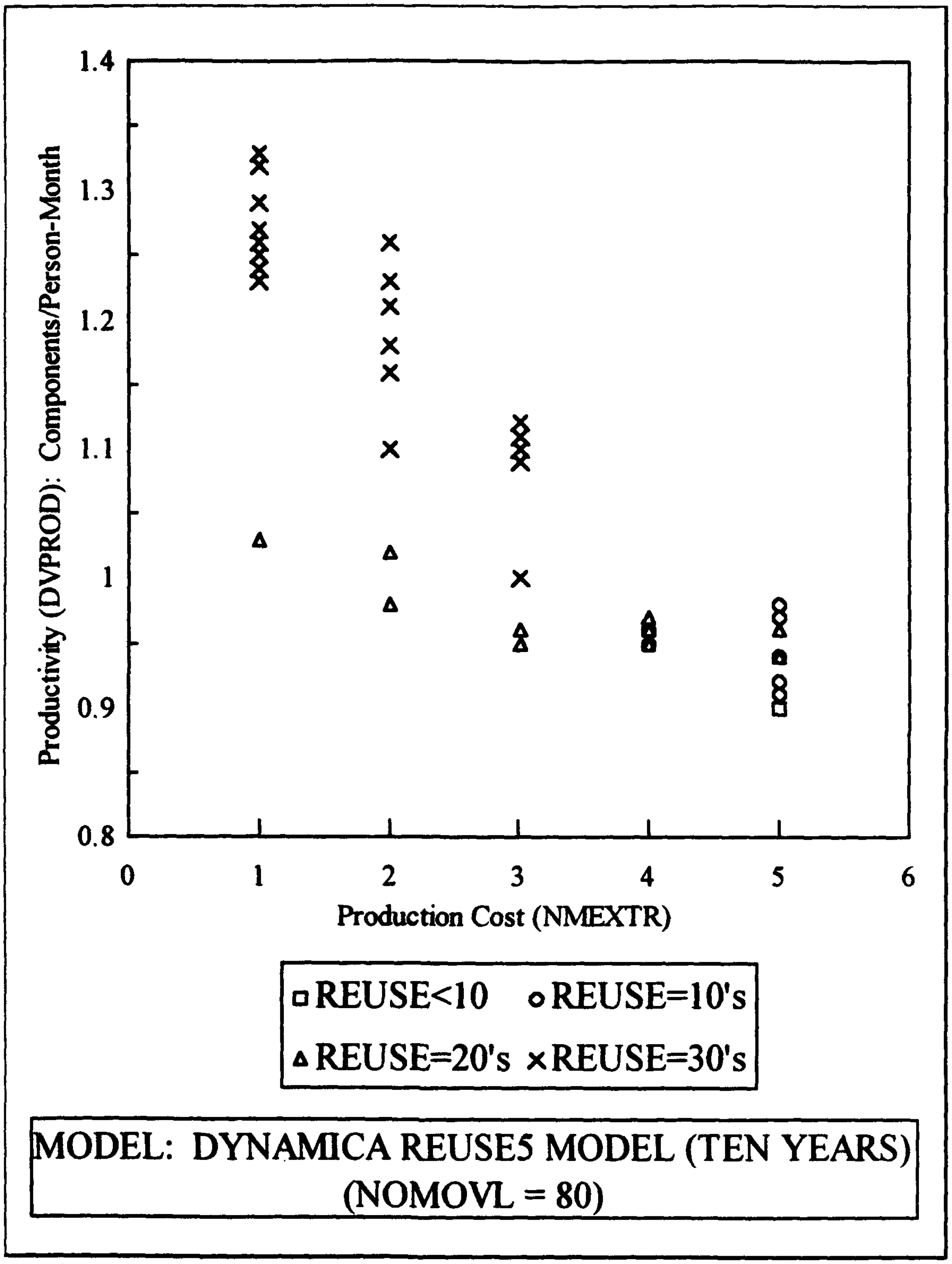


FIGURE 11: Dynamic State *Productivity* versus *Production Cost:* varying *Production Cost* (NMEXTR)

3. Steady State Simulation Results

The next set of simulation runs was designed to measure long term steady state relationships. The Dynamica Reuse5 Model program was used to produce 16 runs, from which only one data point was taken at the end of each 20 year simulation. That is, data points were taken only when the software producing organization was at steady state.

Each of 16 simulations had a value for *Production Cost* that was in the range from 1 to 5. The values of the three variables, *Productivity* (DVPROD), *Reuse Rate* (REUSE) and *Production Cost* (NMEXTR), were taken at the end of each 20 year simulation and are presented in Table 7 and plotted in Figure 12. The 16 Dynamica Reuse5 Model simulation graphs are presented as Figures D6 through D21 in Appendix D. The tabulated results are presented as Tables D6 through D21 in Appendix D.

TABLE 7: STEADY STATE *REUSE RATE* AND *PRODUCTIVITY* VERSUS *PRODUCTION COST*: VARYING *PRODUCTION COST*

Production Cost (NMEXTR)	*Reuse Rate* (REUSE)	*Productivity* (DVPROD)
1.00	35.92	1.26
1.25	35.99	1.23
1.50	35.40	1.20
1.75	35.80	1.19
2.00	35.91	1.17
2.25	36.04	1.15
2.50	36.19	1.13
2.75	36.01	1.11
3.00	35.79	1.09
3.25	35.49	1.07
3.50	34.96	1.05
3.75	26.42	0.99
4.00	16.66	0.95
4.25	12.69	0.92
4.50	11.20	0.91
5.00	8.75	0.90

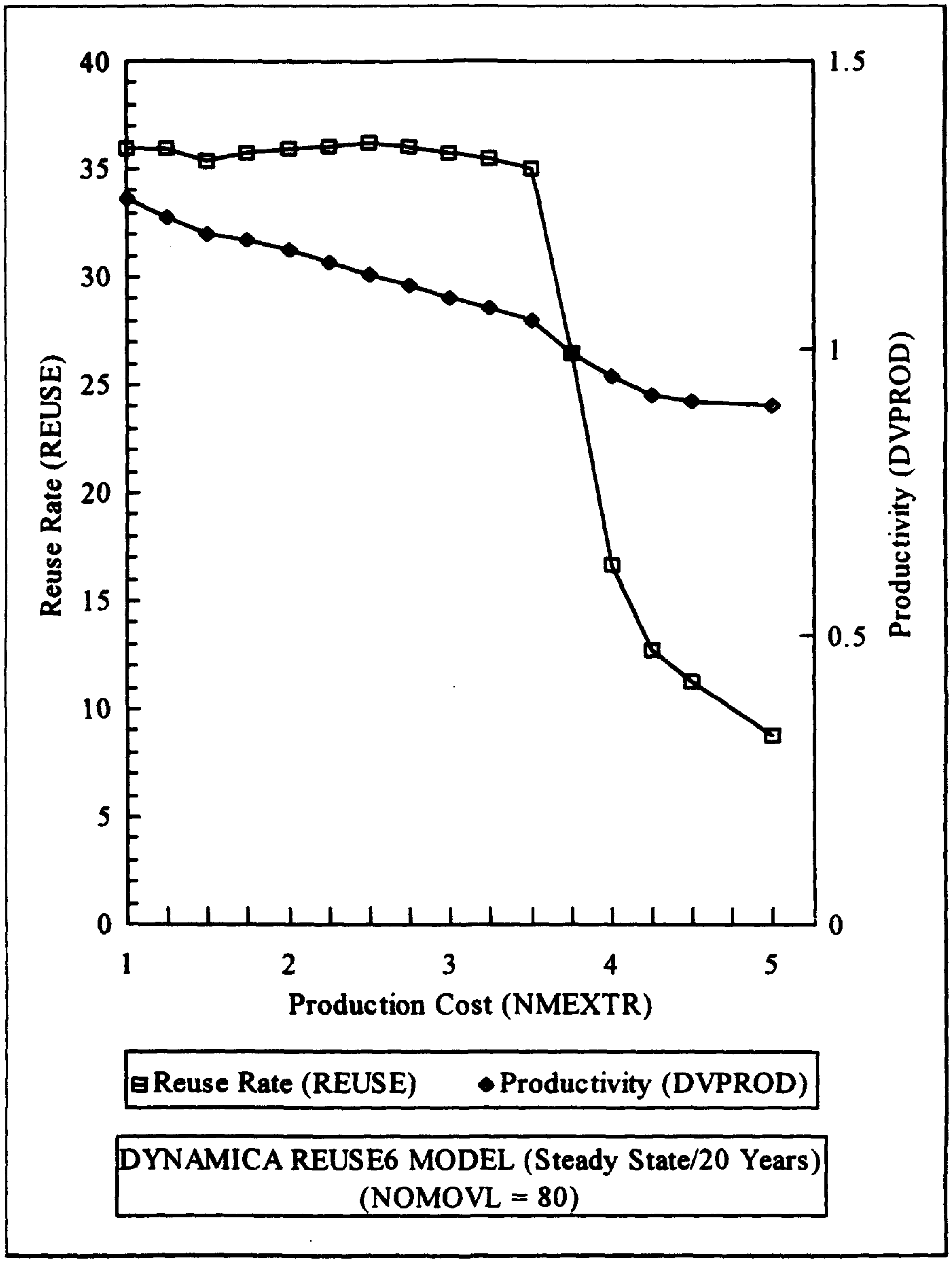


FIGURE 12: Steady State *Reuse Rate and Productivity* versus *Production Cost:* varying *Production Cost* (NMEXTR)

In Figure 12, *Productivity* varies inversely with *Production Cost*, that is, as *Production Cost* increases, *Productivity* decreases at a fairly constant rate. *Reuse Rate* values remain fairly constant for values of NMEXTR between 1 and 3.5, but for values of NMEXTR greater than 3, *Reuse Rate* has a pronounced inverse relationship with *Production Cost.* That is, as *Production Cost* increases above 3.5, *Reuse Rate* sharply declines. The main reason for this sharp decline in *Reuse Rate* is that very high *Production Costs* create a disincentive to create reusable components. Over the long term this leads to a depleted repository, which in turn causes *Reuse Rates* to drop. This is demonstrated in Figure 13, which plots the *Repository Size* for two 20 year simulations for NMEXTR equal to 2 or 5.

The short-term, dynamic state relationship between *Productivity* and *Production Cost* is like the long-term, constant state relationship, that is, as *Production Cost* decreases, *Productivity* increases. Like *Consumption Cost*, *Production Cost* is a good long-term as well as short-term indicator of the economic benefits of the software reuse process, that is, as *Production Cost* decreases, *Productivity* increases, and *Unit Cost* decreases.

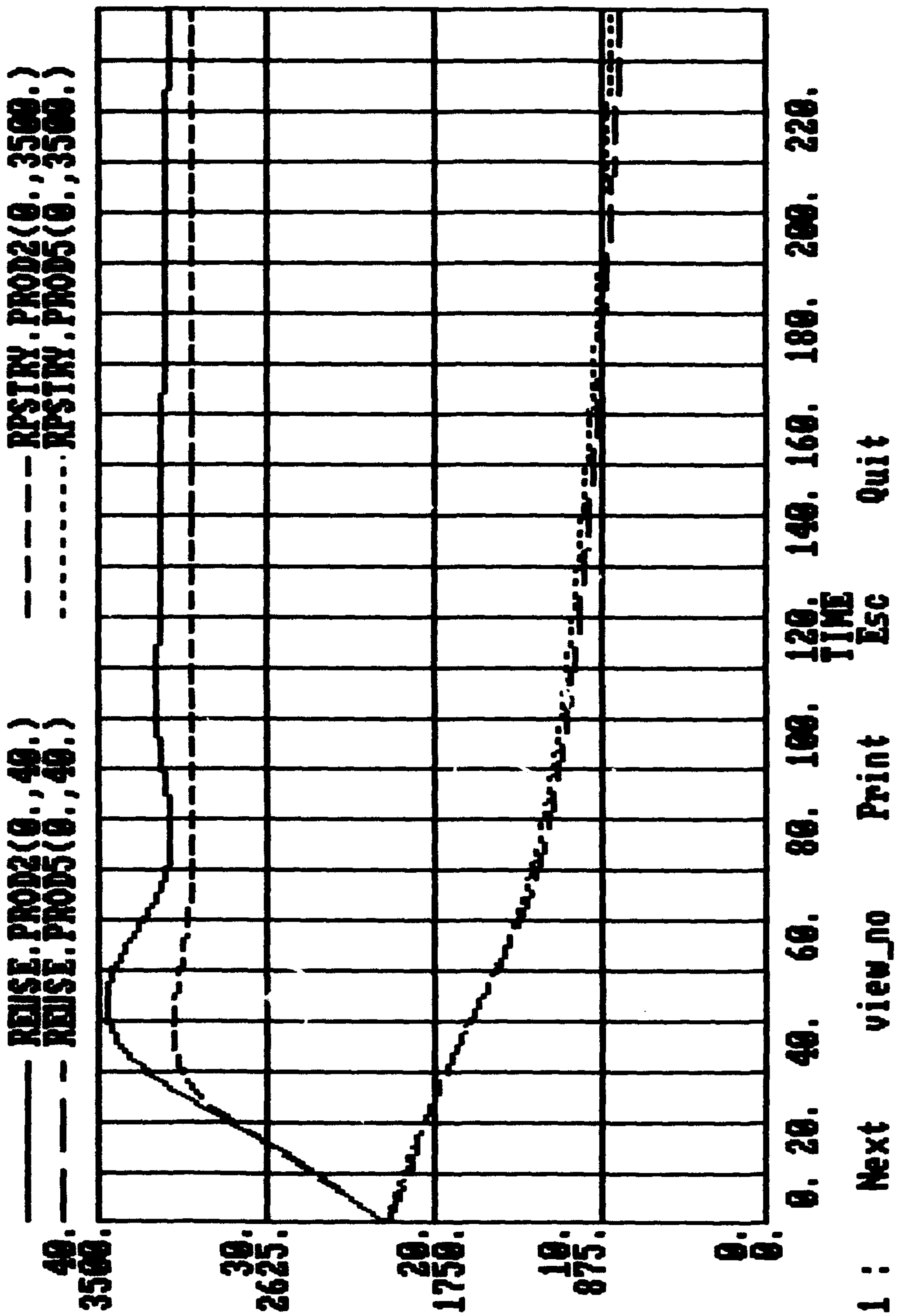


FIGURE 13: Steady State *Reuse Rate* and *Repository Size* at very high and low *Production Cost*

E. (EMPLOYEE) AVERAGE EMPLOYMENT

1. Variables Studied: *Average Employment*, *Productivity* and *Reuse Rate*

Average Employment (AVEMPT) is the average employment time in months for the workforce. It is related to the turnover rate for employees, i.e., a low *Average Employment* means the turnover rate is high, and vice versa. Typically, low employee turnover rates are associated with higher software productivities, that is, the longer the average employment time is for employees, the higher the software productivity is. The nominal or base line value for the Dynamica Reuse Model for *Average Employment* is 42 months. *Productivity* and *Reuse Rate* were discussed in Section IV.B.

2. Dynamic State Simulation Results

The Dynamica Reuse5 Model program was run for a series of three ten-year simulations. In each run the value of the variable *Average Employment* (AVEMPT) was set to one of the following values: 21, 42 and 84 months. Values of three variables, *Average Employment* (AVEMPT), *Productivity* (DVPROD), and *Reuse Rate* (REUSE), were taken at one-year intervals and are presented in Table 8 and Figure 14. In Appendix E, the Dynamica Reuse5 Model simulation graphs are presented as Figures E1 through E3; the tabulated results are presented as Tables E1 through E3.

In Figure 14, all three variables are directly related to each other. The highest *Average Employment* of 84 months results in the largest values of *Productivity* and *Reuse Rate* obtained in any of the simulations so far, 1.93 components/person-month and 65.27% respectively, compared to 1.26 and 39.46% in previous simulations. Unlike the roughly parallel lines in Figures 5 and 11, these simulation run lines appear to merge.

TABLE 8: DYNAMIC STATE *PRODUCTIVITY* VERSUS *REUSE RATE*: VARYING *AVERAGE EMPLOYMENT*			
Reuse Rate (REUSE)	*Productivity* at *Average Employment* = 84	*Productivity* at *Average Employment* = 42	*Productivity* at *Average Employment* = 21
30.43	1.11		
42.13	1.30		
43.56	1.34		
43.58	1.32		
45.98	1.43		
47.21	1.45		
51.57	1.51		
53.14	1.60		
64.07	1.87		
65.27	1.93		
28.08		1.02	
34.62		1.10	
35.96		1.16	
36.01		1.16	
36.45		1.18	
36.55		1.18	
36.64		1.19	
37.46		1.21	
38.98		1.23	
39.46		1.26	
25.02			0.90
29.24			0.95
31.72			1.02
31.76			1.02
31.78			1.02
31.83			1.02
31.83			1.02
32.03			1.03
32.19			1.03
32.55			1.04

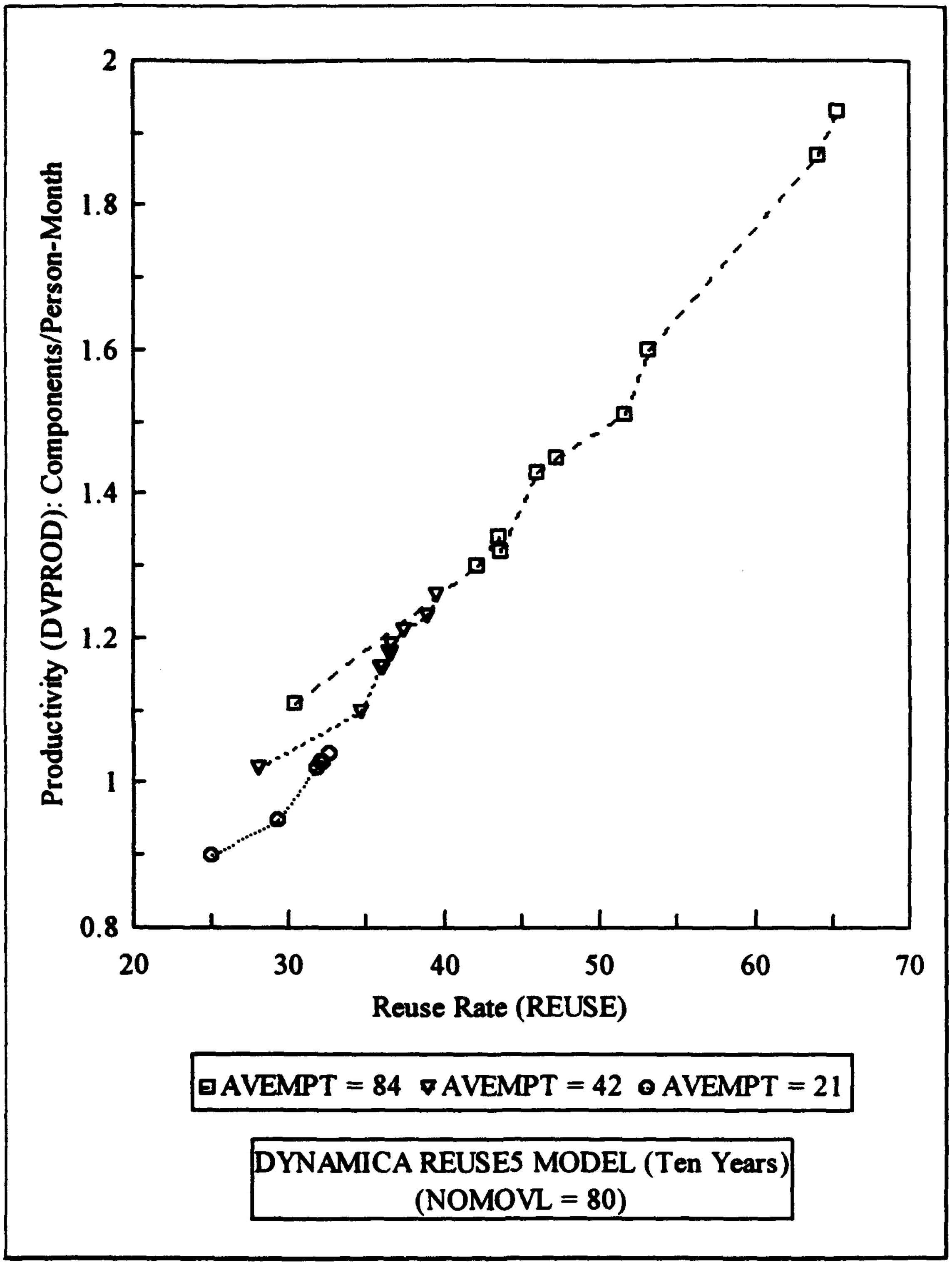


FIGURE 14: Dynamic State *Productivity* versus *Reuse Rate*: varying *Average Employment* (AVEMPT)

F. REUSABLE COMPONENT RETIREMENT AGE

1. Variables Studied: *Retirement Age, Repository Size, Productivity* and *Reuse Rate*

Retirement Age (NMRCLF) is the average life in months of a reusable component before it is retired from the repository. A low *Retirement Age* means the rate of retiring reusable components from the repository will be high, and vice versa. The nominal or base value for the Dynamica Reuse5 Model program is 60 months. *Productivity* and *Reuse Rate* were discussed in Section IV.B.1 *Repository Size* was discussed in Section IV.E.1.

2. Dynamic State Simulation Results

The Dynamica Reuse5 Model program was run for a series of three ten-year simulations. In each run the value of the variable *Retirement Age* (NMRCLF) was set to one of the following values: 30, 60 and 120 months. Values of three variables, *Retirement Age* (NMRCLF), *Productivity* (DVPROD), and *Reuse Rate* (REUSE), were taken at one-year intervals and are presented in Table 9 and Figure 15. In Appendix F, the Dynamica Reuse5 Model simulation graphs are presented as Figures F1 through F3; the tabulated results are presented as Tables F1 through F3.

In general in Figure 15, all three variables are directly related to each other, that is, as *Retirement Age* increases, so does *Productivity* and *Reuse Rate*. The highest *Retirement Age* of 120 months results in the greatest gains in *Productivity* and *Reuse*. Unlike previous graphs of *Productivity* versus *Reuse Rate* for dynamic state systems (Figures 5 and 11), the lines representing the individual simulation runs are not in parallel, but converge. In other words, for a given value for *Reuse Rate*, the corresponding *Productivity* will be virtually the same no matter what the assigned value for *Retirement Age* was for that run.

TABLE 9: DYNAMIC STATE *PRODUCTIVITY* VERSUS *REUSE RATE*: VARYING *RETIREMENT AGE*

Reuse Rate (REUSE)	*Productivity* for *Retirement Age* = 30	*Productivity* for *Retirement Age* = 60	*Productivity* for *Retirement Age* = 120
23.56	0.98		
25.43	0.97		
28.03	1.00		
30.31	1.04		
31.43	1.05		
31.93	1.04		
32.27	1.06		
32.45	1.05		
32.58	1.06		
32.73	1.06		
28.08		1.02	
34.62		1.10	
35.96		1.16	
36.01		1.16	
36.45		1.18	
36.55		1.18	
36.65		1.19	
37.46		1.21	
38.98		1.23	
39.46		1.26	
30.63			1.05
38.77			1.24
38.78			1.21
39.06			1.24
39.25			1.25
39.29			1.25
39.56			1.26
40.17			1.28
42.16			1.32
42.29			1.34

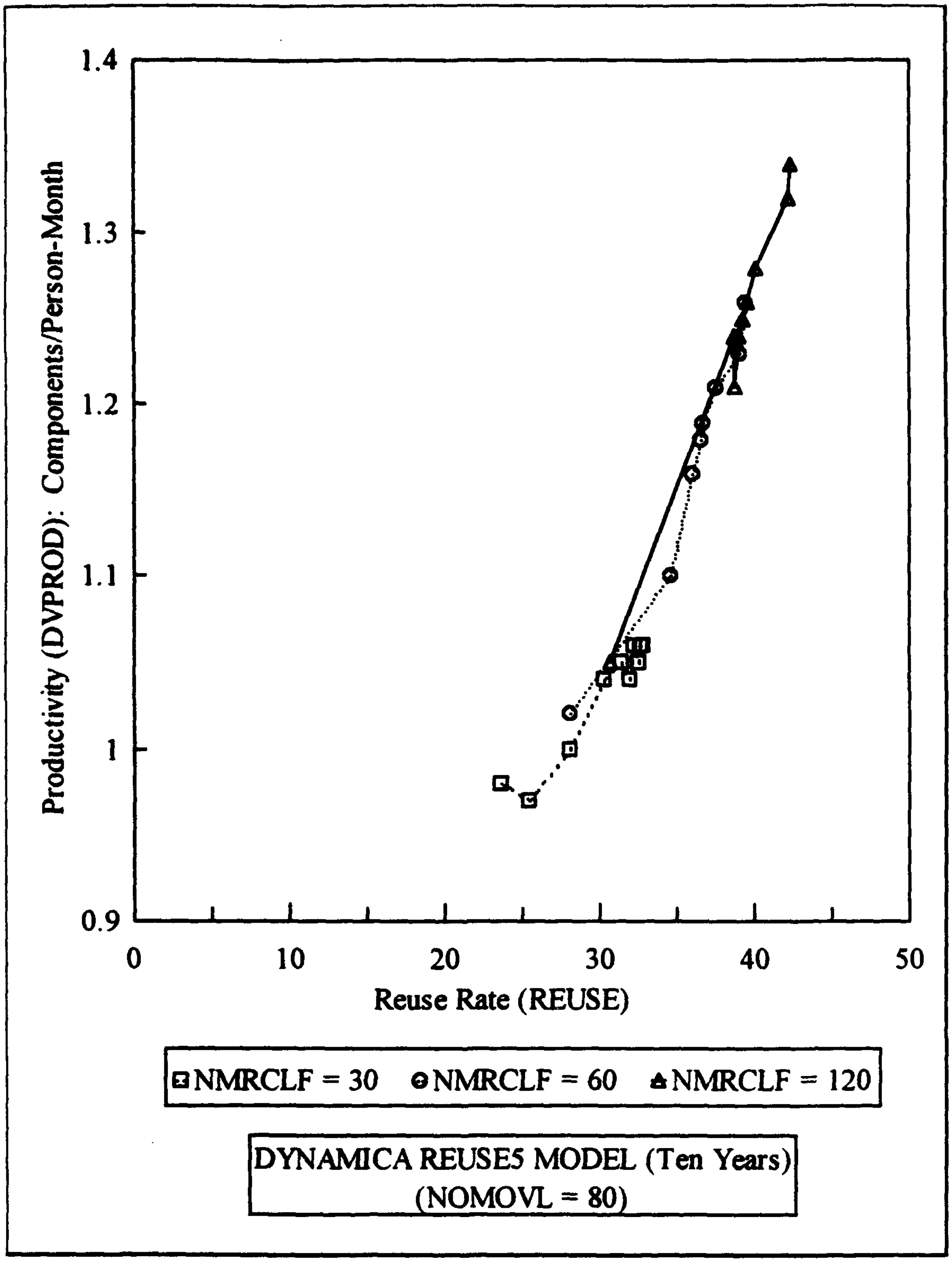


FIGURE 15: Dynamic State *Productivity* versus *Reuse Rate:* varying *Retirement Age* (NMRCLF)

3. Steady State Simulation Results

The Dynamica Reuse6 Model was used to produce a series of 17 steady state simulations in which the variable *Retirement Age* (NMRCLF) ranged from 20 to 180 months. The values of three variables, *Productivity* (DVPROD), *Reuse Rate* (REUSE), and *Repository Size* (RPSTRY), were taken at the end of each 20 year simulation and are presented in Table 10, and corresponding Figures 16 and 17, respectively. The 17 Dynamica Reuse6 Model simulation graphs are presented as Figures F4 through F20 in Appendix F. The tabulated results are presented as Tables F4 through F20 in Appendix F.

TABLE 10: STEADY STATE *REUSE RATE, PRODUCTIVITY* AND *REPOSITORY SIZE* VERSUS *RETIREMENT AGE*: VARYING *RETIREMENT AGE*

Retirement Age (NMRCLF)	*Reuse Rate* (REUSE)	*Productivity* (DVPROD)	*Repository Size* (RPSTRY)
20.00	25.30	0.95	2,228.00
30.00	31.69	1.04	2,729.00
40.00	33.59	1.11	2,865.00
50.00	34.92	1.15	2,958.00
60.00	35.91	1.17	3,026.00
70.00	36.70	1.19	3,080.00
80.00	37.33	1.21	3,123.00
90.00	37.85	1.22	3,159.00
100.00	38.27	1.24	3,208.00
110.00	38.62	1.25	3,255.00
120.00	38.92	1.25	3,296.00
130.00	39.19	1.26	3,332.00
140.00	39.42	1.27	3,364.00
150.00	39.63	1.28	3,392.00
160.00	39.82	1.28	3,418.00
170.00	39.98	1.29	3,440.00
180.00	40.13	1.29	3,461.00

In Figure 16, both *Reuse Rate* and *Productivity* increase with increasing *Retirement Age*, just as in the dynamic state organization relationships depicted in Figure 15. Clearly, the longer a reusable component is available in a repository, the longer it is available for reuse, and the higher the resulting productivity.

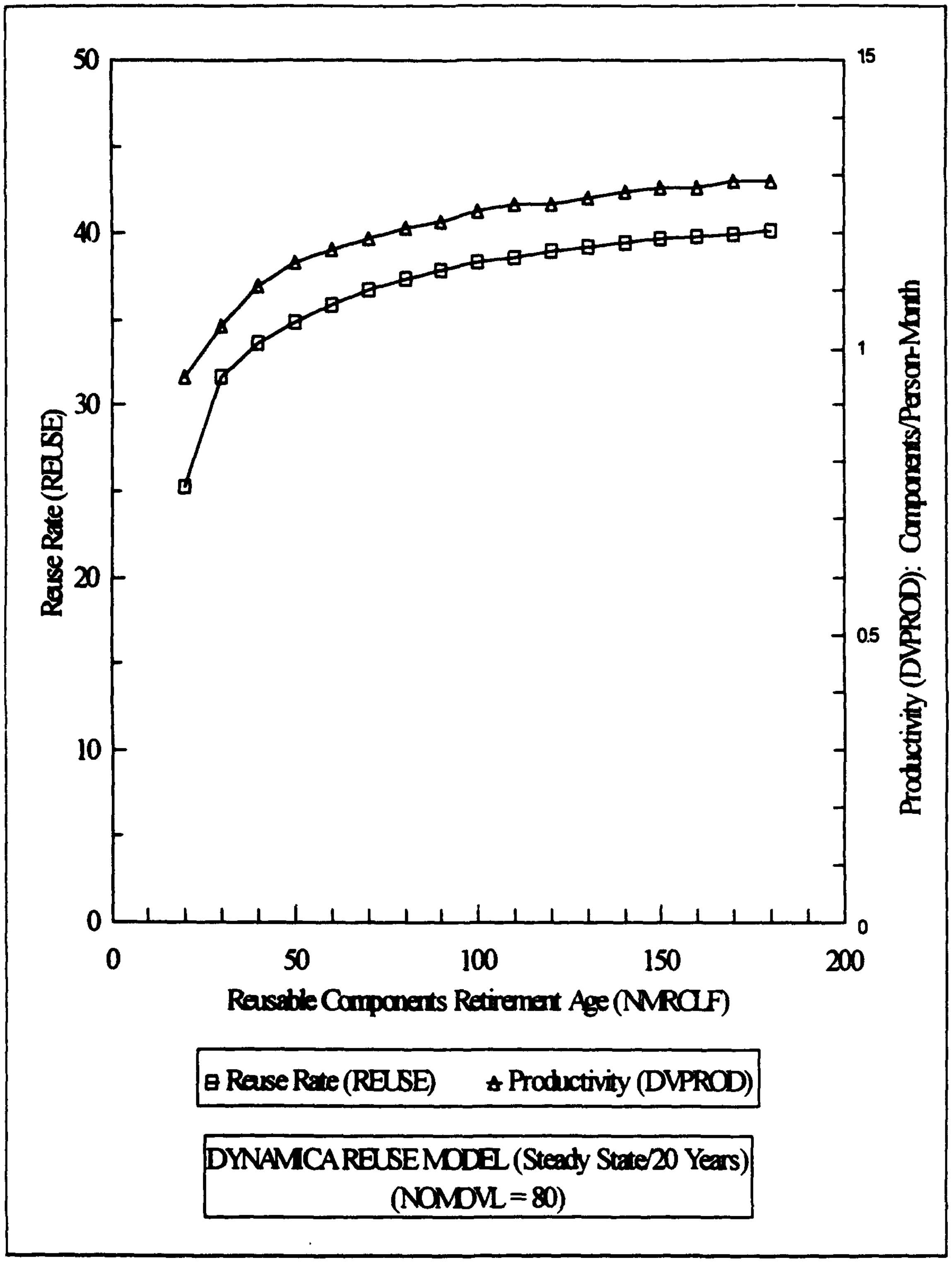


FIGURE 16: Steady State *Reuse Rate* and *Productivity* versus *Retirement Age*: varying *Retirement Age* (NMRCLF)

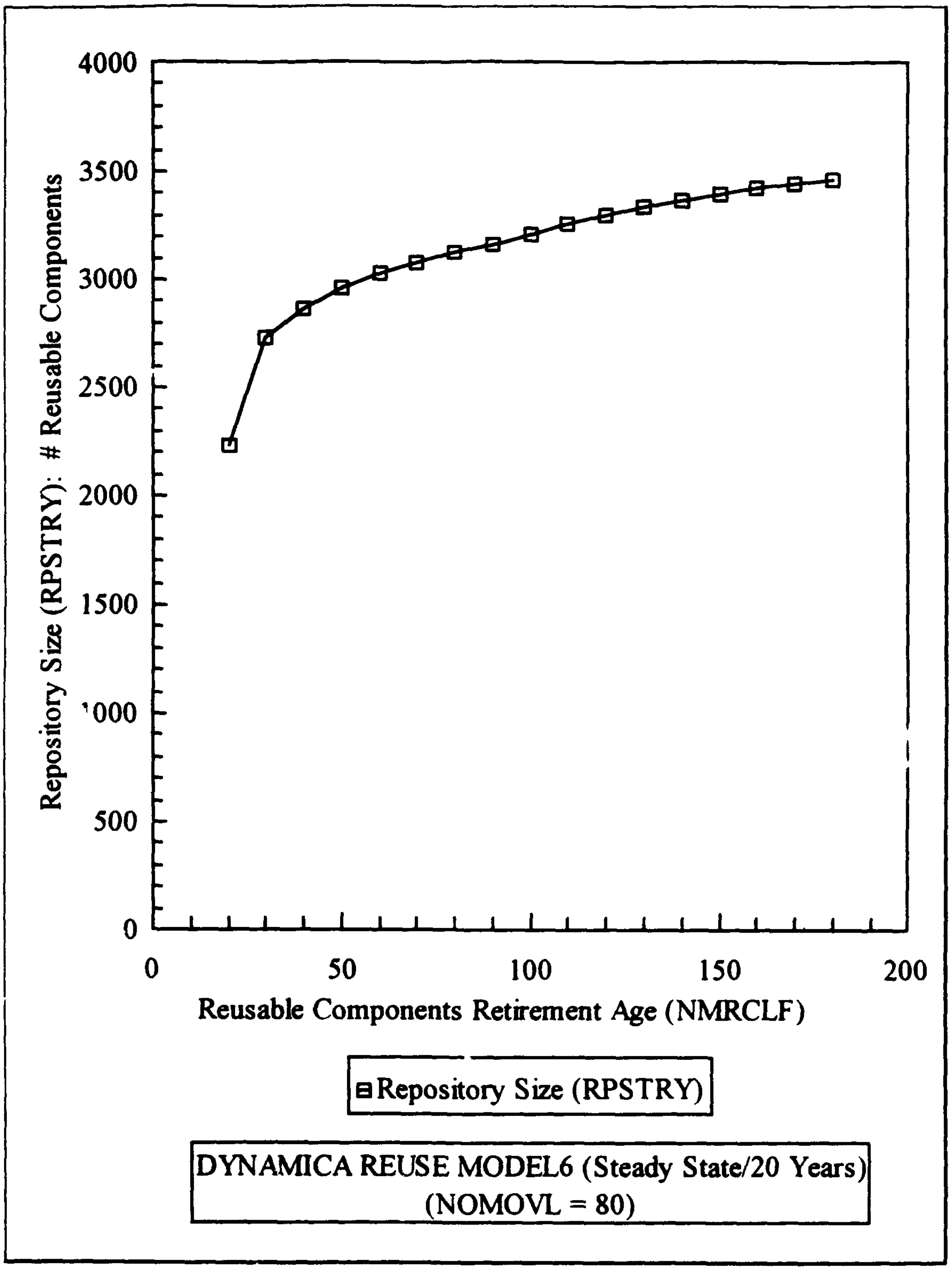


FIGURE 17: Steady State *Repository Size* versus *Retirement Age*: varying *Retirement Age* (NMRCLF)

In Figure 17, *Repository Size* shows the same type of direct relationship with *Retirement Age* as *Productivity* and *Reuse Rate*, that is, increasing *Retirement Age* is associated with increasing numbers of reusable components in the organization's repository. As the rate of retiring reusable components decreases, the numbers of reusable components in the repository increases.

V. CONCLUSIONS

A. RESULTS

Until now the study of reuse economics has been without a model that is both comprehensive and long-term. Models published in the literature offer simplified "snapshots," providing limited information about a few factors in the reuse process, after completion of applications or projects. In contrast, the Dynamica Reuse Model provides information about a wide array of important technical and managerial factors at any time during the reuse process. The Dynamica Reuse Model provides information about the reuse process during an organization's entire life span, from the early dynamic state in the initial reuse process to the later steady state at equilibrium.

Results suggest that the long-term steady state relationships between variables may be different from those in the short-term. This, for example, was demonstrated by the study of the relationship between *Reuse Rate* and *Consumption Cost*. In the short term (Figures 5 and 6), *Reuse Rate* and *Consumption Cost* demonstrate a strong inverse relationship, in contrast to the constant relationship demonstrated between the long-term, steady state *Reuse Rate* and *Consumption Cost* in Figure 8. Although it is counter-intuitive, long-term *Reuse Rate*, but not short-term *Reuse Rate*, appears to be insensitive to *Consumption Cost* values.

Unlike the economic models presented in Chapter III, the Dynamica Reuse Model is able to isolate one factor and examine its effects and what affects it, in a dynamic environment that simulates the diversity and comprehensiveness of an entire software producing organization. For example, simulations presented in this thesis demonstrate the strong direct relationships between *Productivity* and variables such as *Average Employment* time of employees in Figure 14 and reusable component *Retirement Age* in Figures 15 and 16. Managers of software development organizations and government contractors with these organizations should consider organizational factors such as these when making decisions about the relative health of a particular software development organization.

The managerial implications of the Dynamica Reuse Model are dual. As with other integrative, system dynamics models, "the first and primary purpose of the model is to enhance our understanding of the software development process". . .and the second purpose is "to make predictions about the general process by which software systems are developed" (Abdel-Hamid, 1990). These capabilities will enable the Dynamica Reuse Model to serve as a management support tool.

For example, a manager knowing values for such variables as *Consumption Cost*, *Production Cost*, and *Average Employment* time of employees will be able to use the model to make knowledgeable predictions about organizational *Reuse Rate*, as opposed to making intuitive or "gut-feel" estimates without the model. Because software planning impacts on *Productivity* at different *Reuse Rate* levels, the model can be used to support an organization's software estimation tools for cost and schedule. Of course, this initial stage of the Dynamica Reuse Model will require validation and customization to a particular organization before use as a management support tool.

B. SUGGESTED FUTURE AREAS OF STUDY

1. Longitudinal Studies

The Dynamica Reuse Model simulates a large software producing organization using software reuse as an organization-wide process. A possible next step is to conduct longitudinal empirical studies of actual software producing organizations to assess the accuracy of the model's projections.

2. Software Reuse Between Organizations

The Dynamica Reuse Model simulates a single software producing organization engaged in software reuse solely as an internal process. The organization is not obtaining or supplying reusable components with other organizations as is done in the DoD environment (Cruickshank and Gaffney, 1991; Green, 1992; Menke, 1993; Schwartz, April 1993; and Schwartz, May 1993). An interesting next step is to extend the model to simulate multiple software producing organizations engaged in the process of software reuse as a group.

APPENDIX A: GLOSSARY

Application Overlap: Dynamica Reuse Model variable named NOMOVL. Defines a nominal degree of overlap between the simulated organization's software systems. Software systems developed from very similar domains will have a high degree of overlap; those developed from very different domains will have a lower degree of overlap Default NOMOVL is 60% overlap, however, for this thesis, NOMOVL was reset to 80% overlap for all simulations.

AVEMPT: See *Average Employment.*

Average Employment: Dynamica Reuse Model variable named AVEMPT. The average employment time in months equivalent to 25% turnover of the workforce. Default AVEMPT is 42 months.

CDVPRD: See *Cumulative Productivity.*

Consumption Cost: Dynamica Reuse Model variable named NMFRRU. The relative cost of reusing a component, i.e., the nominal fraction of development effort to reuse a component, and is a unitless fraction (development person-months per component to reuse divided by development person-months per component to develop an all-new component). A value of one is the same relative cost as the cost to use an all-new component. A value of less than one means it costs less to reuse a component in an application than to use an all-new component. Default NMFRRU is 0.25.

Cumulative Productivity: Dynamica Reuse Model variable named CDVPRD. The cumulative development productivity expressed in components per person-month. Whereas *Productivity* is the instantaneous average development productivity at a single point in time, *Cumulative Productivity* is the accumulated development productivity up to the time it is measured. It is the average productivity at time *t*.

DVPMPC: See *Unit Cost.*

Dynamic state: Refers to a system not in equilibrium.

NMEXTR: See *Production Cost.*

NMFRRU: See *Consumption Cost.*

NMRCLF: See *Retirement Age.*

NOMOVL: See *Application Overlap.*

Production Cost: Dynamica Reuse Model variable named NMEXTR. The relative cost to produce a reusable component. It is the fraction of development effort to produce a reusable component, and is a unitlesss number (development person-months per reusable component to develop divided by development person-months per component to develop a non-reusable component). Default NMEXTR is 2.

Productivity: Dynamica Reuse Model variable named DVPROD. A measure of how productive the development process is, and is expressed as components per person-month. *Productivity* is the inverse of *Unit Cost.*

Repository Size: Dynamica Reuse Model variable named RPSTRY. Represents the number of reusable components in the organization's reusable component repository.

Retirement Age: Dynamica Reuse Model variable named NMRCLF. The average life in months of a reusable component at the time it is retired. Also related to the repository retirement rate, i.e., a low *Retirement Age* implies a high rate of retiring components from the reusable component repository, and vice versa. Default NMRCLF is 60 months.

Reuse Rate: Dynamica Reuse Model variable named REUSE. The rate at which components are reused, and is expressed as the number of reusable components used divided by the total number of components, i.e., both reusable and new.

REUSE: See *Reuse Rate*.

RPSTRY: See *Repository Size*.

Steady state: Refers to a system in equilibrium.

Unit Cost: **Dynamica Reuse Model variable named DVPMPC. The average cost to develop a software component, and is expressed as person-months per component. *Unit Cost* is the inverse of *Productivity*.**

APPENDIX B: UNIT COST DOCUMENTATION

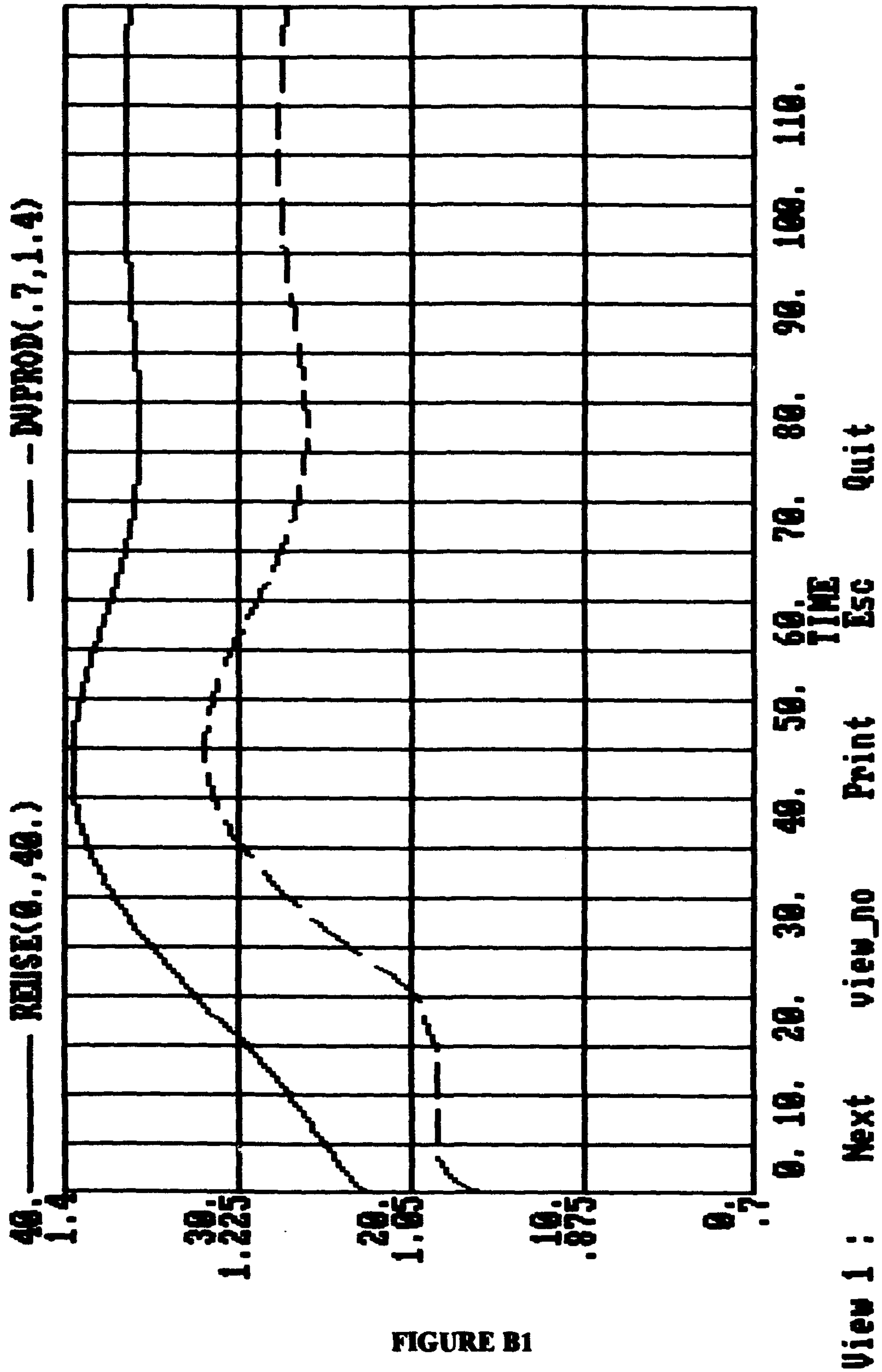


FIGURE B1

TABLE B1

Model = REUSE5; Run = UNITCOST.RSL; Change = NONE

Years	DVPROD	REUSE	DVPMPC	AVGUSE
0.00	0.98	22.60	1.02	2.00
1.00	1.02	28.08	0.98	2.76
2.00	1.10	34.62	0.91	3.38
3.00	1.23	38.98	0.81	4.25
4.00	1.26	39.46	0.80	5.07
5.00	1.21	37.46	0.83	5.64
6.00	1.16	35.96	0.86	6.02
7.00	1.16	36.01	0.86	6.32
8.00	1.18	36.55	0.85	6.59
9.00	1.19	36.65	0.84	6.82
10.00	1.18	36.45	0.85	7.00

APPENDIX C: CONSUMPTION COST DOCUMENTATION

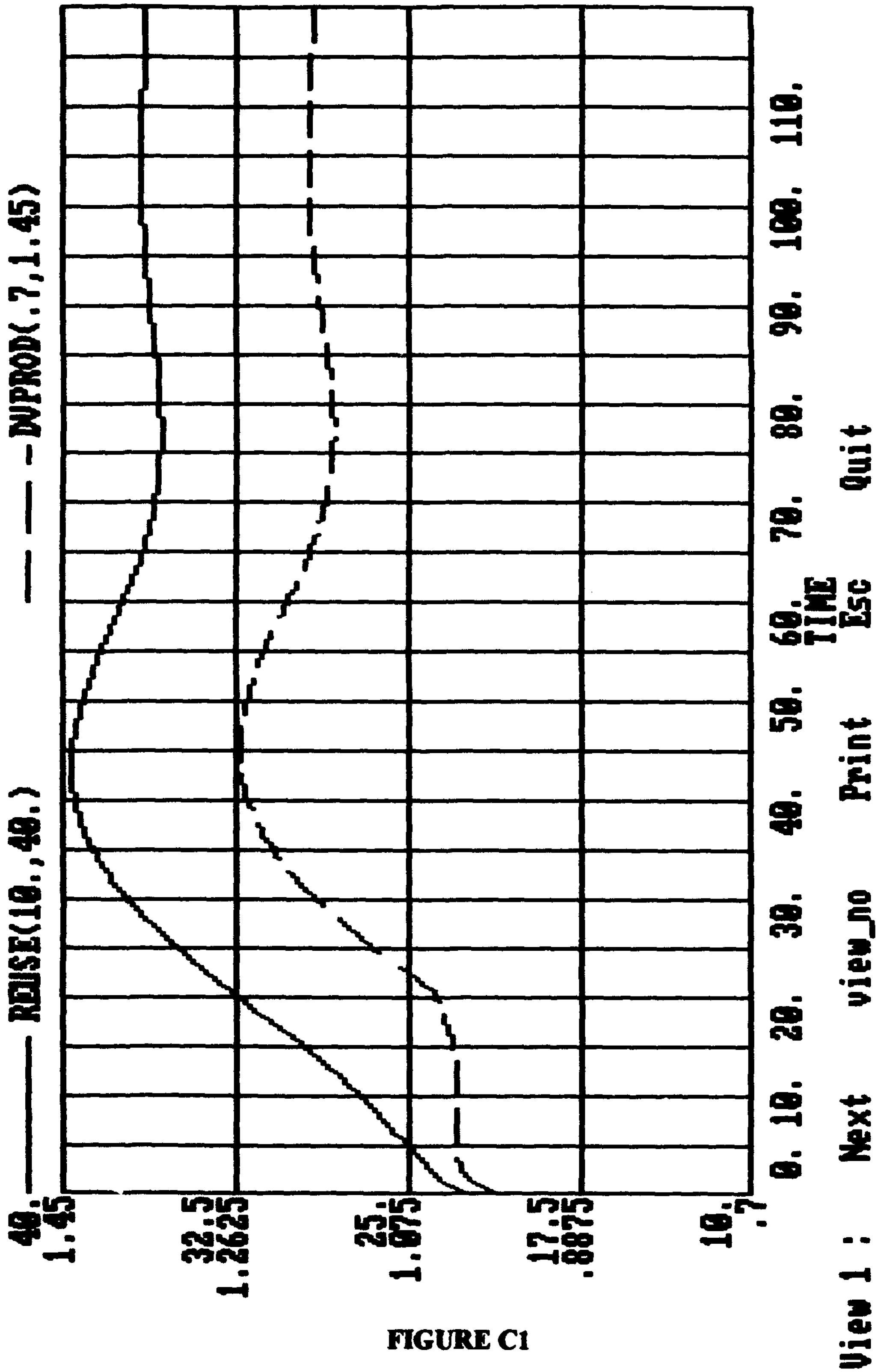


FIGURE C1

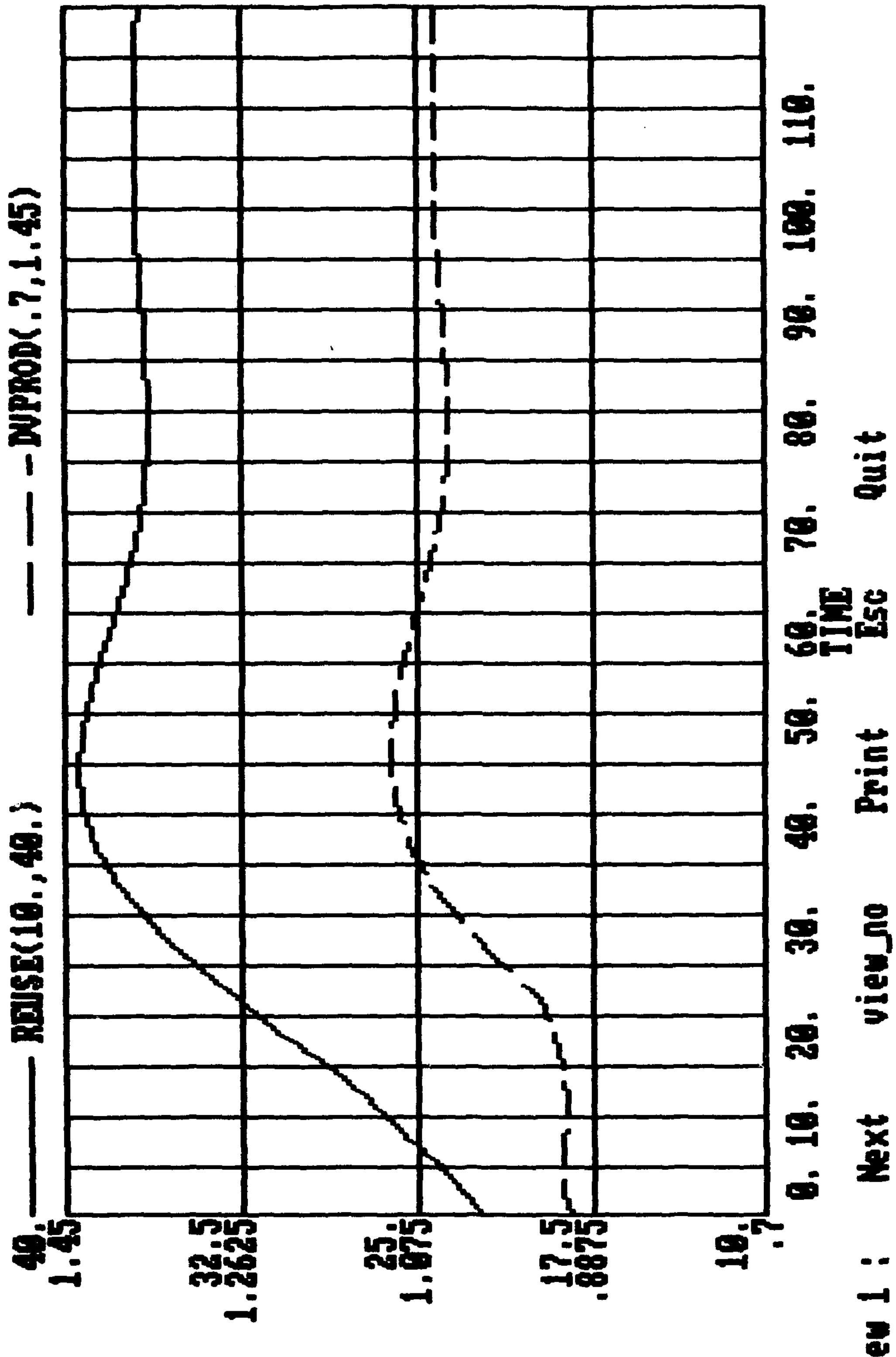

REUSE(10.,40.)
DUPROD(.7,1.45)
40.
1.45
32.5
1.2625
25.
1.075
17.5
.8875
10.
.7
0. 10. 20. 30. 40. 50. 60. 70. 80. 90. 100. 110.
TIME
View 1 :
Next
view_no
Print
Esc
Quit

FIGURE C2

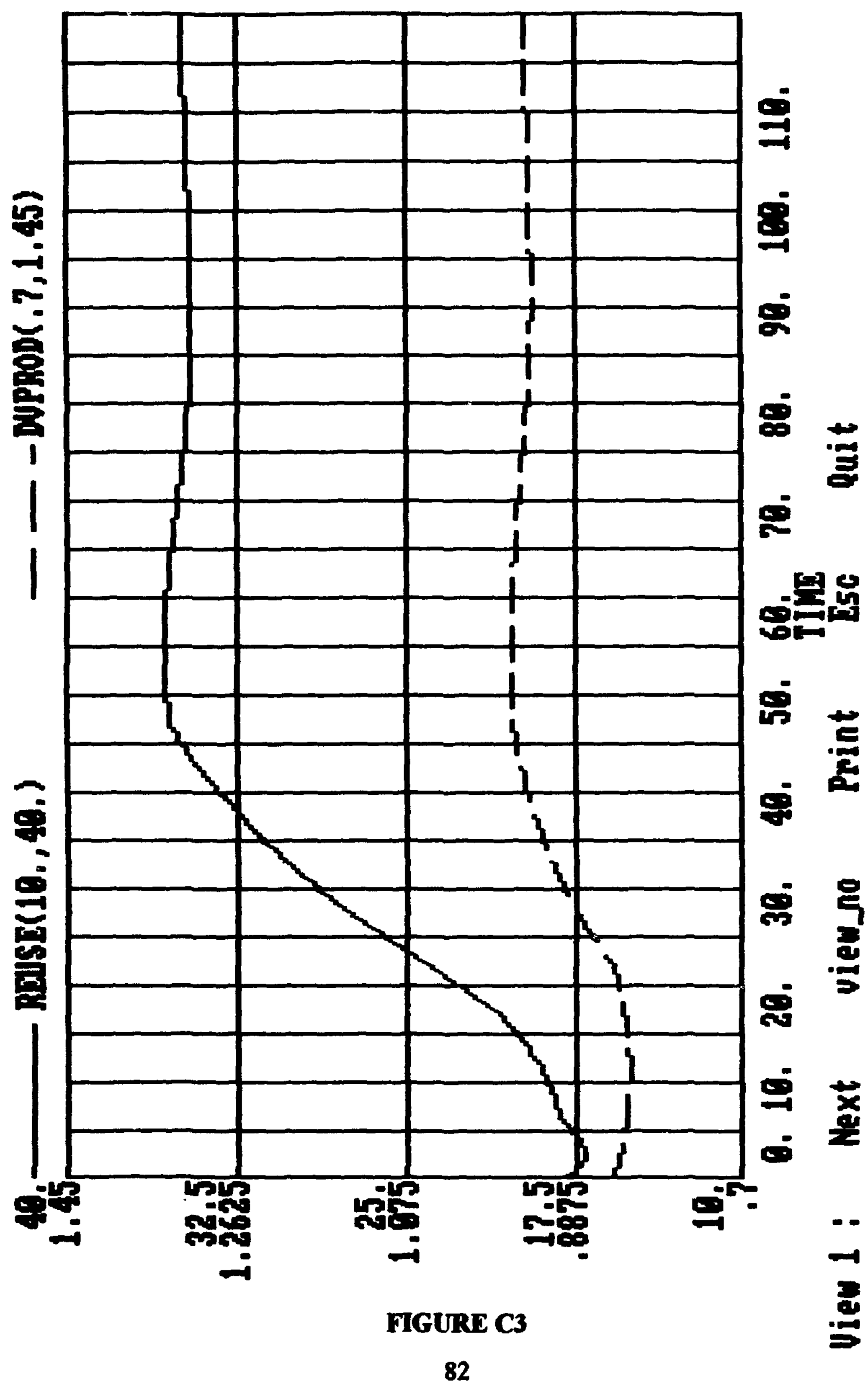

REUSE(10.,40.)
DUPROD(.7,1.45)
40.
1.45
32.5
1.2625
25.
1.075
17.5
.8875
10.
.7
0. 10. 20. 30. 40. 50. 60. 70. 80. 90. 100. 110.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE C3

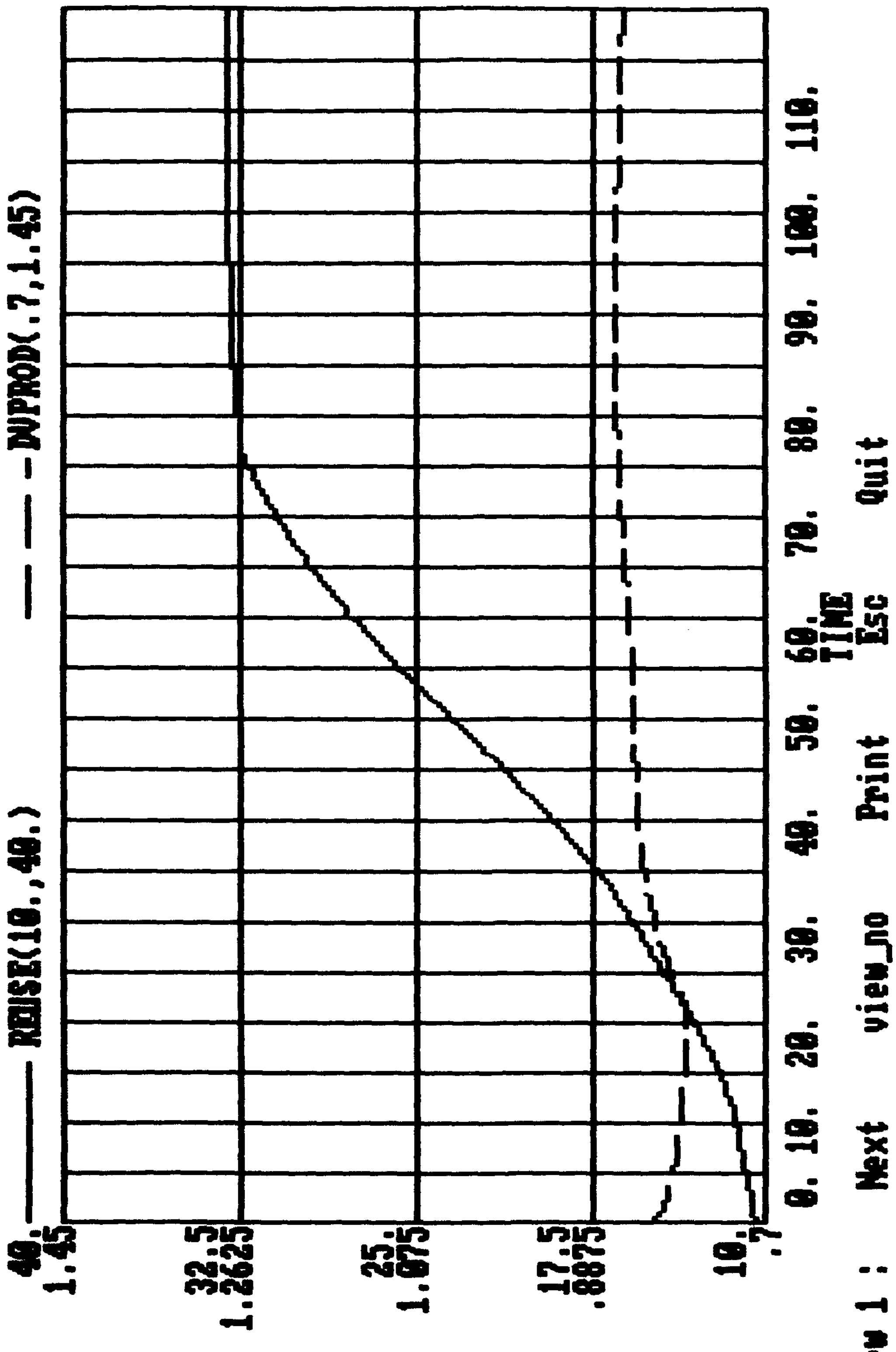

REUSE(10.,40.)
DUPROD(.7,1.45)
40.
1.45
32.5
1.2625
25.
1.075
17.5
.8875
10.
.7
0. 10. 20. 30. 40. 50. 60. 70. 80. 90. 100. 110.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE C4

TABLE C1

Model = REUSE5; Run = RCOST225.RSL; Change = NMFRRU = 0.25

Years	DVPROD	REUSE	NMFRRU	AVGUSE
0.00	0.98	22.60	0.25	2.00
1.00	1.02	28.08	0.25	2.76
2.00	1.10	34.62	0.25	3.38
3.00	1.23	38.98	0.25	4.25
4.00	1.26	39.46	0.25	5.07
5.00	1.21	37.46	0.25	5.64
6.00	1.16	35.96	0.25	6.02
7.00	1.16	36.01	0.25	6.32
8.00	1.18	36.55	0.25	6.59
9.00	1.19	36.65	0.25	6.82
10.00	1.18	36.45	0.25	7.00

TABLE C2

Model = REUSE5; Run = RCOST250.RSL; Change = NMFRRU = 0.50

Years	DVPROD	REUSE	NMFRRU	AVGUSE
0.00	0.91	22.09	0.50	2.00
1.00	0.92	27.33	0.50	2.72
2.00	0.97	34.01	0.50	3.32
3.00	1.08	38.60	0.50	4.14
4.00	1.10	39.34	0.50	4.96
5.00	1.08	37.95	0.50	5.55
6.00	1.05	36.71	0.50	5.97
7.00	1.04	36.62	0.50	6.30
8.00	1.05	36.99	0.50	6.59
9.00	1.06	37.10	0.50	6.83
10.00	1.05	36.96	0.50	7.02

TABLE C3

Model = REUSE5; Run = RCOST275.RSL; Change = NMFRRU = 0.75

Years	DVPROD	REUSE	NMFRRU	AVGUSE
0.00	0.85	17.94	0.75	2.00
1.00	0.83	19.17	0.75	2.42
2.00	0.86	25.35	0.75	2.78
3.00	0.93	31.79	0.75	3.41
4.00	0.96	35.57	0.75	4.17
5.00	0.96	35.68	0.75	4.82
6.00	0.95	35.08	0.75	5.32
7.00	0.94	34.59	0.75	5.70
8.00	0.94	34.59	0.75	6.01
9.00	0.94	34.84	0.75	6.27
10.00	0.94	34.98	0.75	6.50

TABLE C4

Model = REUSE5; Run = RCOST210.RSL; Change = NMFRRU = 1.0

Years	DVPROD	REUSE	NMFRRU	AVGUSE
0.00	0.82	10.50	1.00	2.00
1.00	0.79	11.58	1.00	2.13
2.00	0.80	14.14	1.00	2.24
3.00	0.83	17.67	1.00	2.50
4.00	0.84	22.57	1.00	2.91
5.00	0.85	27.74	1.00	3.44
6.00	0.86	31.47	1.00	4.02
7.00	0.86	32.80	1.00	4.56
8.00	0.86	33.03	1.00	5.02
9.00	0.86	33.05	1.00	5.39
10.00	0.86	33.13	1.00	5.70

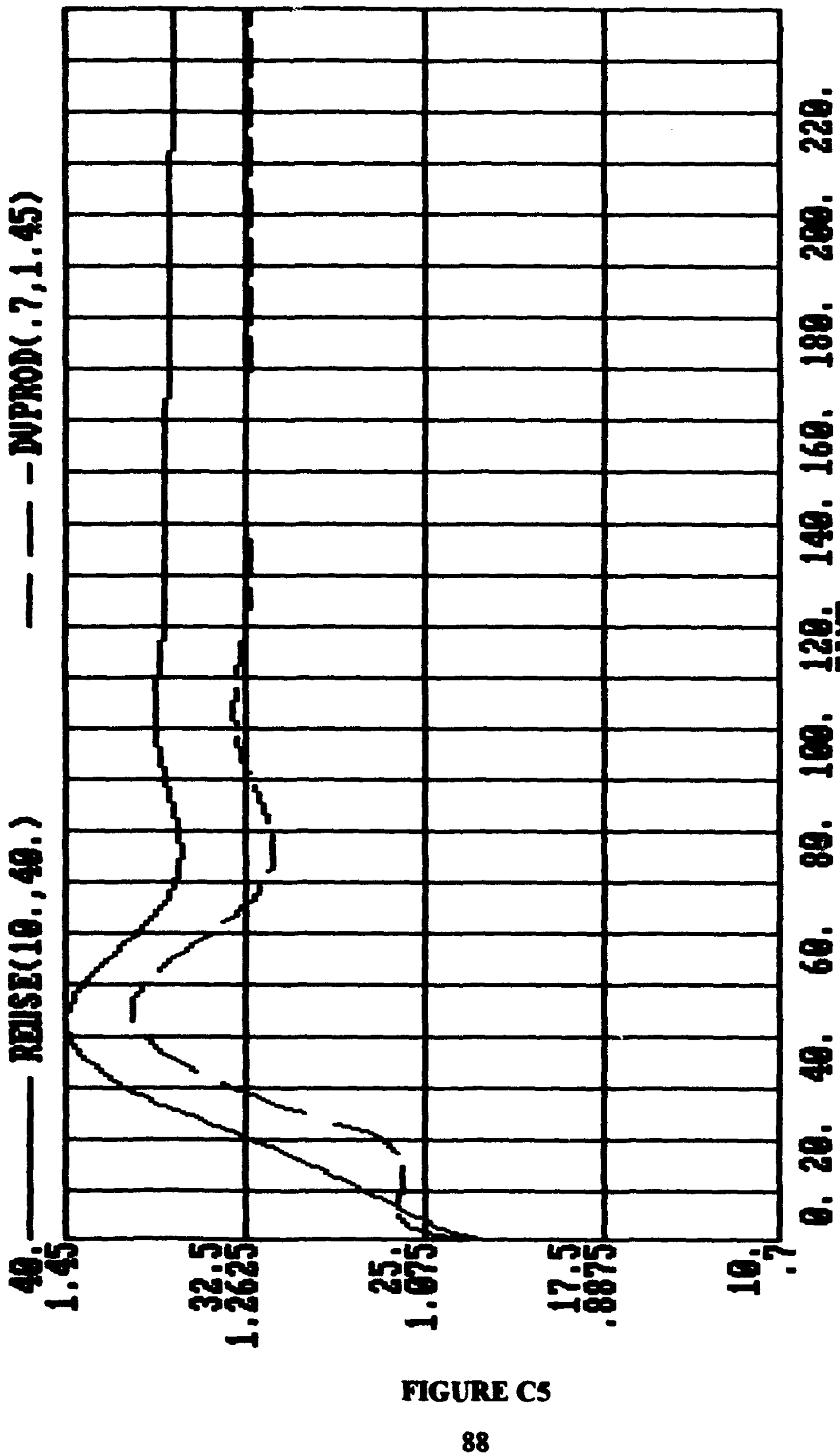

REUSE(10.,40.)
DUPROD(.7,1.45)
40.
1.45
32.5
1.2625
25.
1.075
17.5
.8875
10.
.7
0. 20. 40. 60. 80. 100. 120. 140. 160. 180. 200. 220.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE C5

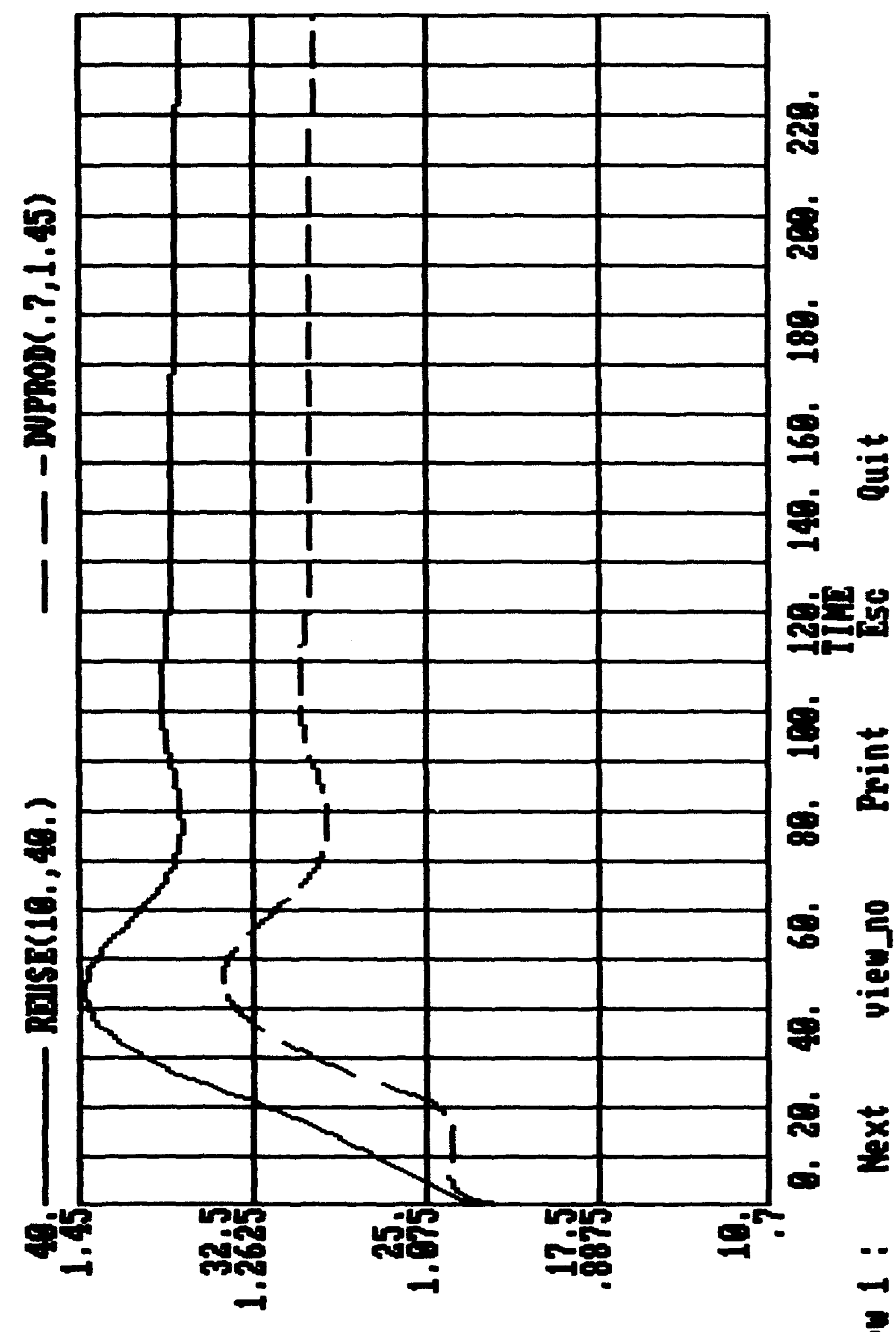

REUSE(10.,40.)
INVPROD(.7,1.45)
40.
1.45
32.5
1.2625
25.
1.075
17.5
.8875
10.
.7
0.
20.
40.
60.
80.
100.
120.
140.
160.
180.
200.
220.
TIME
View 1 :
Next
view_no
Print
Esc
Quit

FIGURE C6

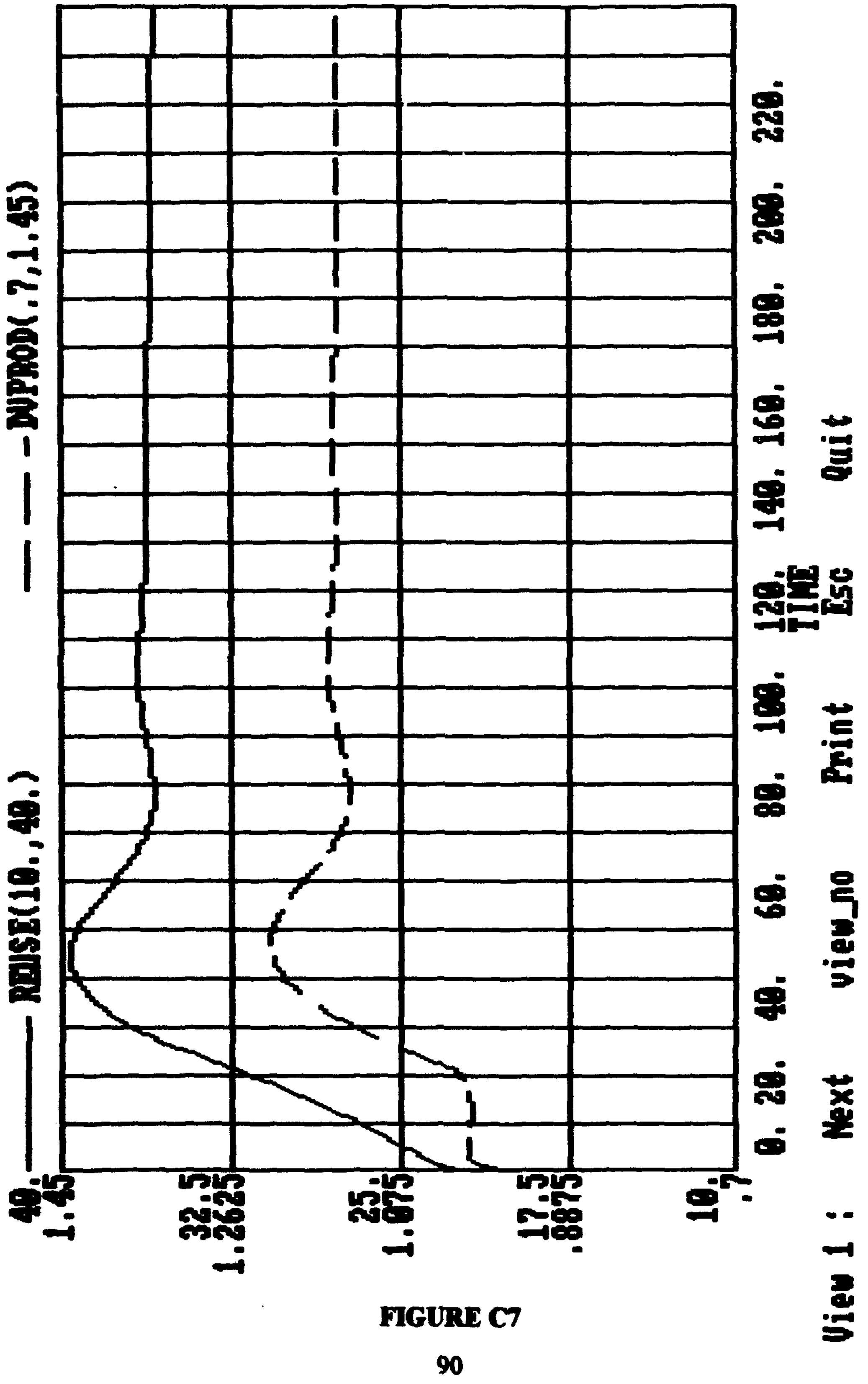

REUSE(10.,40.)
DUPROD(.7,1.45)
40.
1.45
32.5
1.2625
25.
1.075
17.5
.8875
10.
.7
0. 20. 40. 60. 80. 100. 120. 140. 160. 180. 200. 220.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE C7

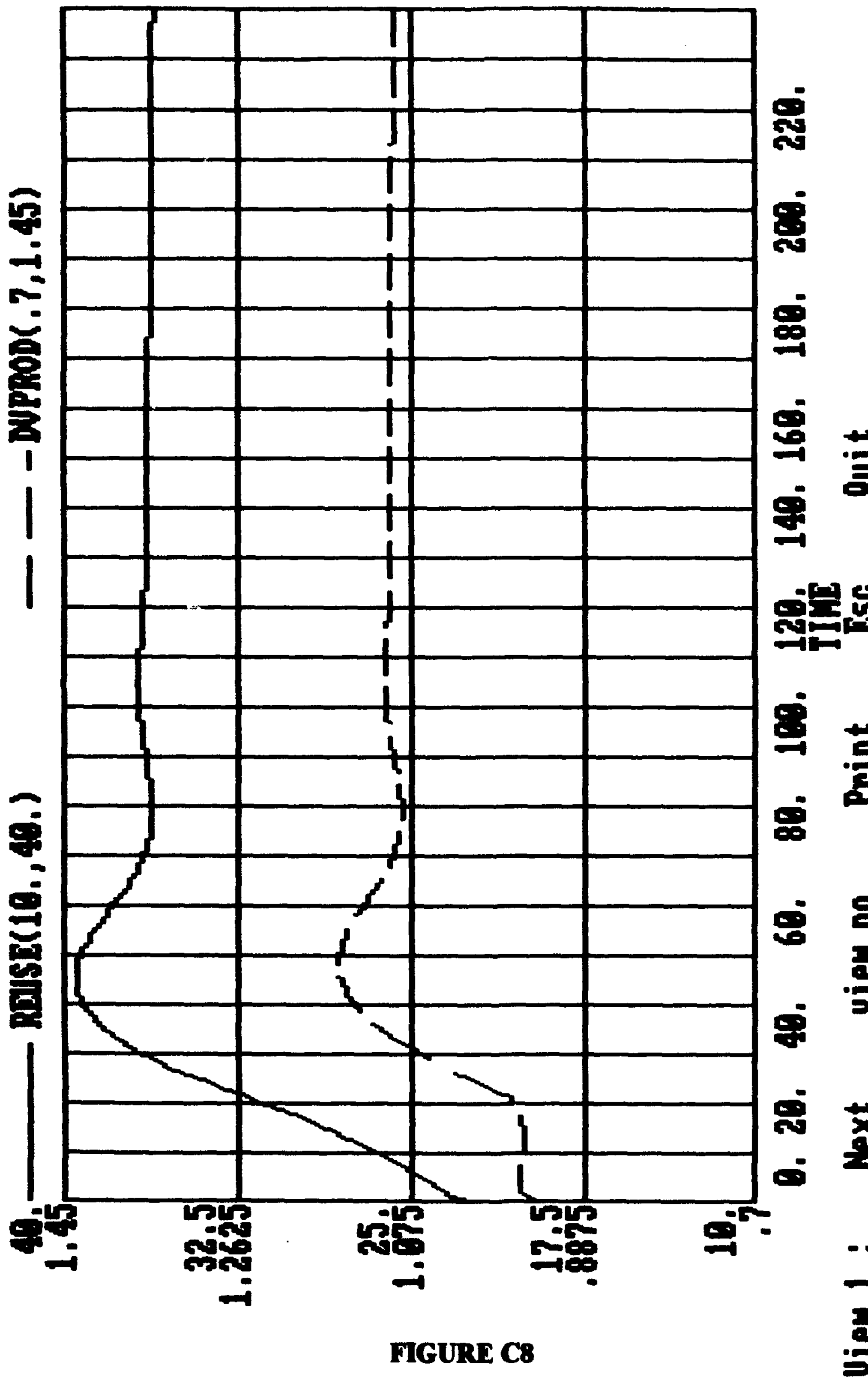
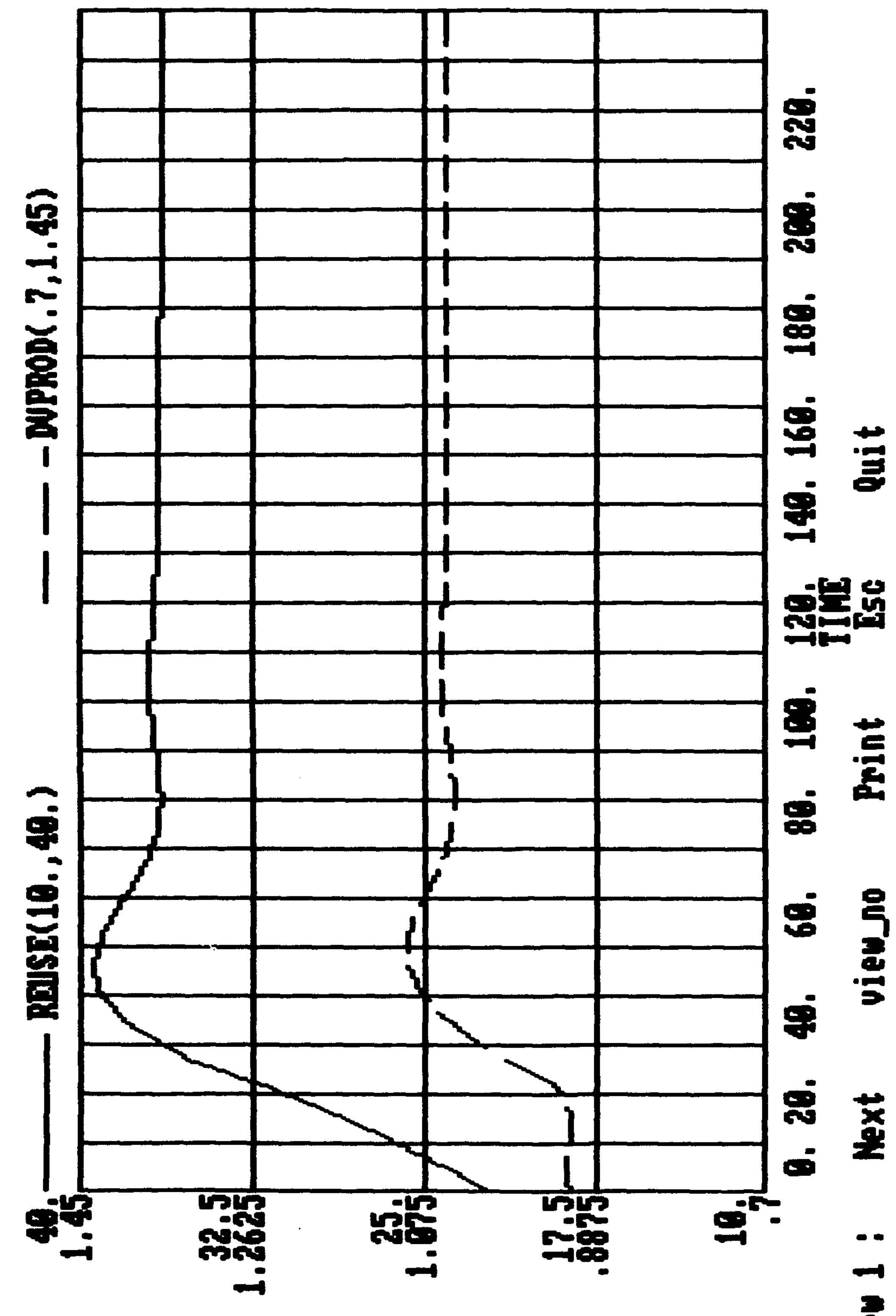

REUSE(10.,40.)
DUPROD(.7,1.45)
40.
1.45
32.5
1.2625
25.
1.075
17.5
.8875
10.
.7
0. 20. 40. 60. 80. 100. 120. 140. 160. 180. 200. 220.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE C8



REUSE(10.,40.)
DUPROD(.7,1.45)
40.
1.45
32.5
1.2625
25.
1.075
17.5
.8875
10.
.7
0. 20. 40. 60. 80. 100. 120. 140. 160. 180. 200. 220.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE C9

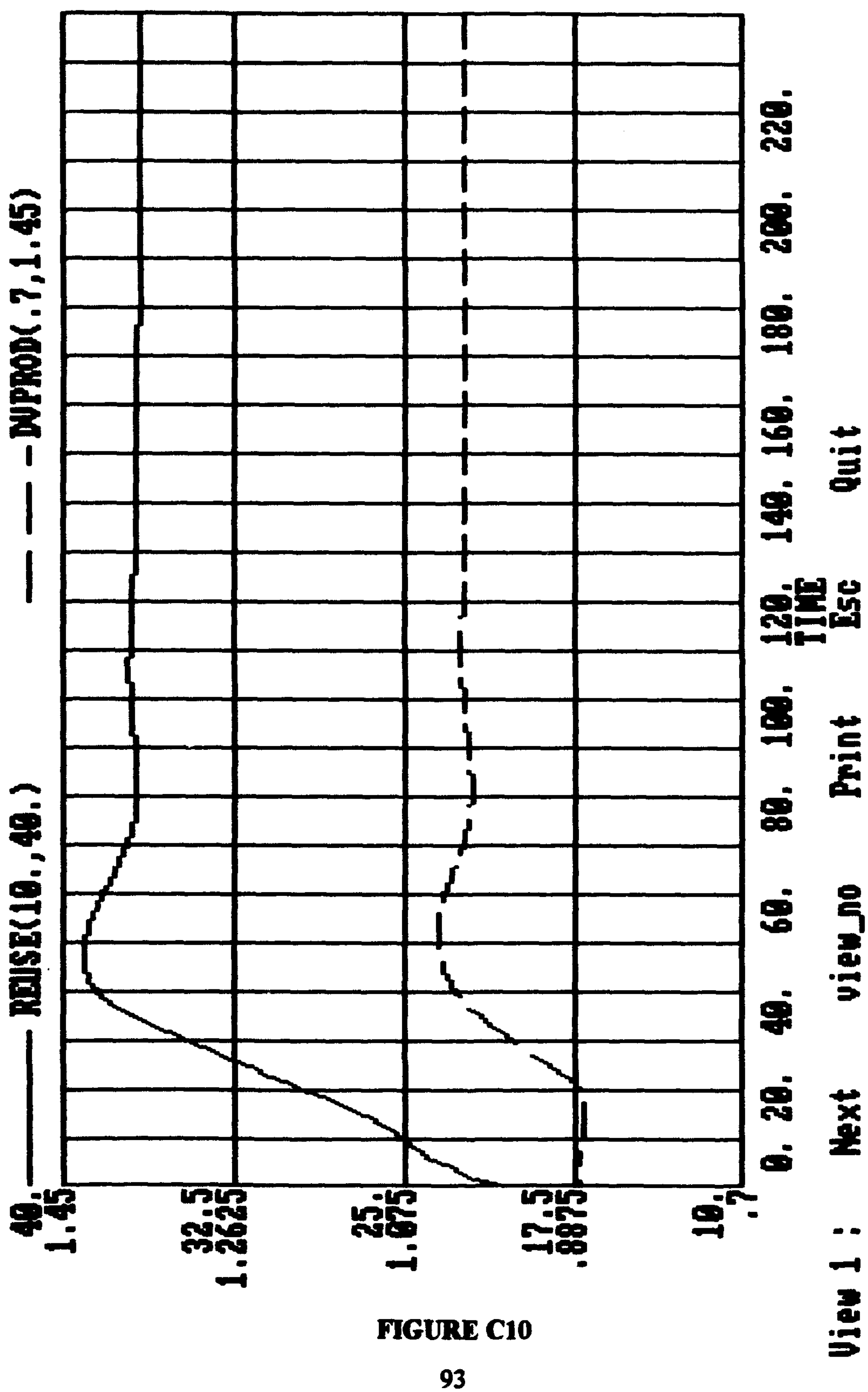

REUSE(10.,40.)
DUPROD(.7,1.45)
40.
1.45
32.5
1.2625
25.
1.075
17.5
.8875
10.
.7
0.
20.
40.
60.
80.
100.
120.
140.
160.
180.
200.
220.
TIME
View 1 :
Next
view_no
Print
Esc
Quit

FIGURE C10

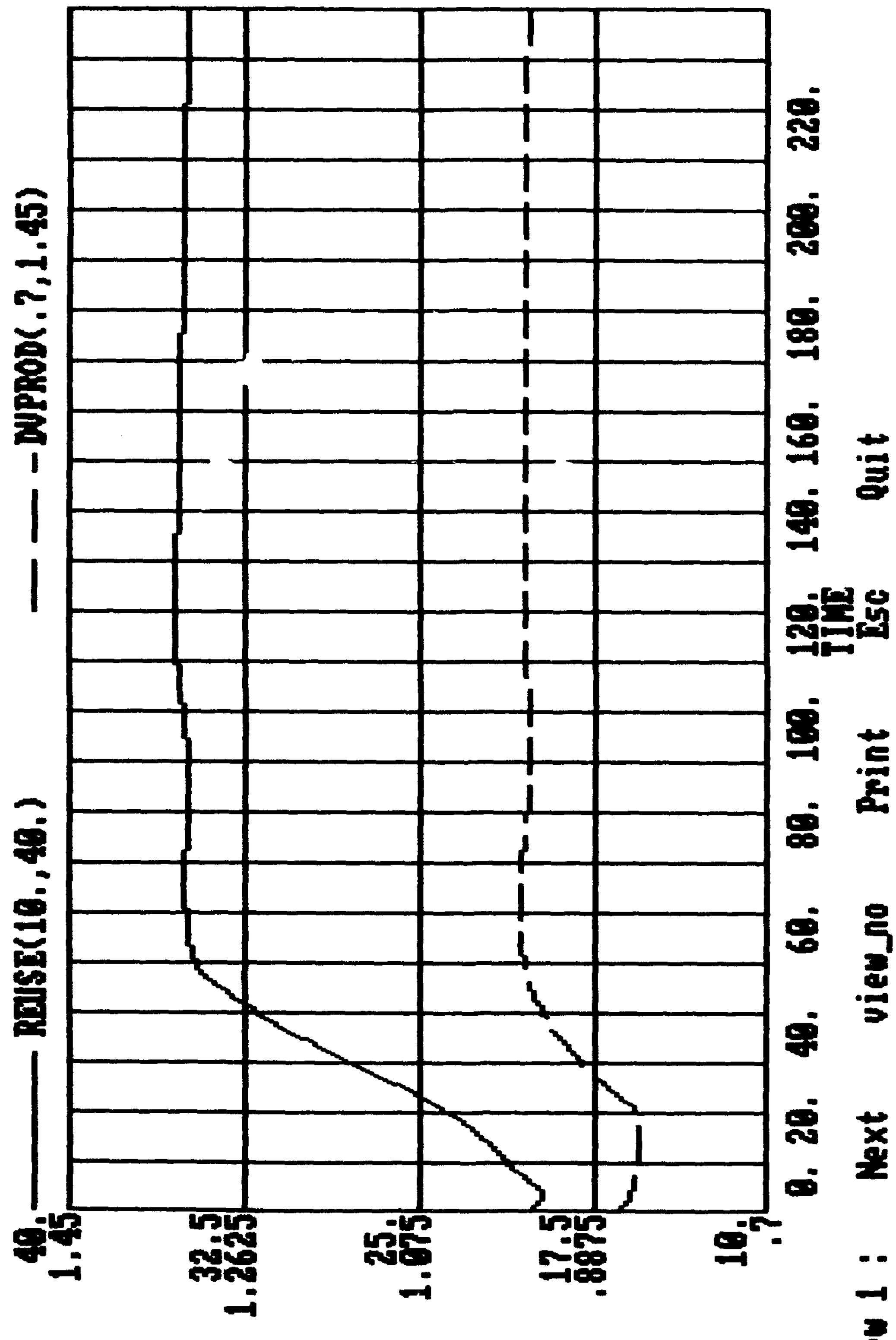
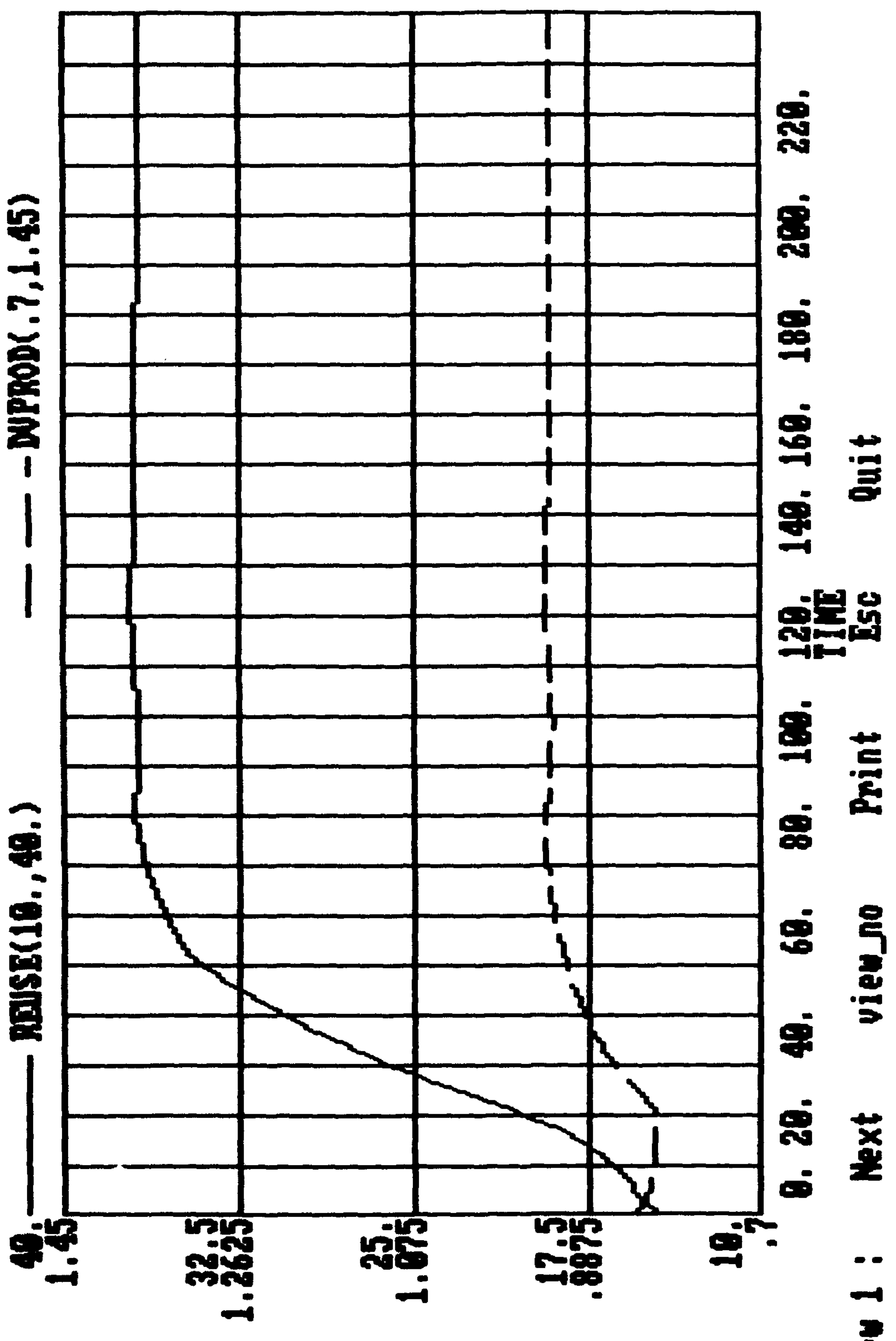

REUSE(10.,40.)
DUPROD(.7,1.45)
40.
1.45
32.5
1.2625
25.
1.075
17.5
.8875
10.
.7
0. 20. 40. 60. 80. 100. 120. 140. 160. 180. 200. 220.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE C11



REUSE(10.,40.)
DUPROD(.7,1.45)
40.
1.45
32.5
1.2625
25.
1.075
17.5
.8875
10.
.7
0.
20.
40.
60.
80.
100.
120.
140.
160.
180.
200.
220.
TIME
View 1 :
Next
view_no
Print
Esc
Quit

FIGURE C12

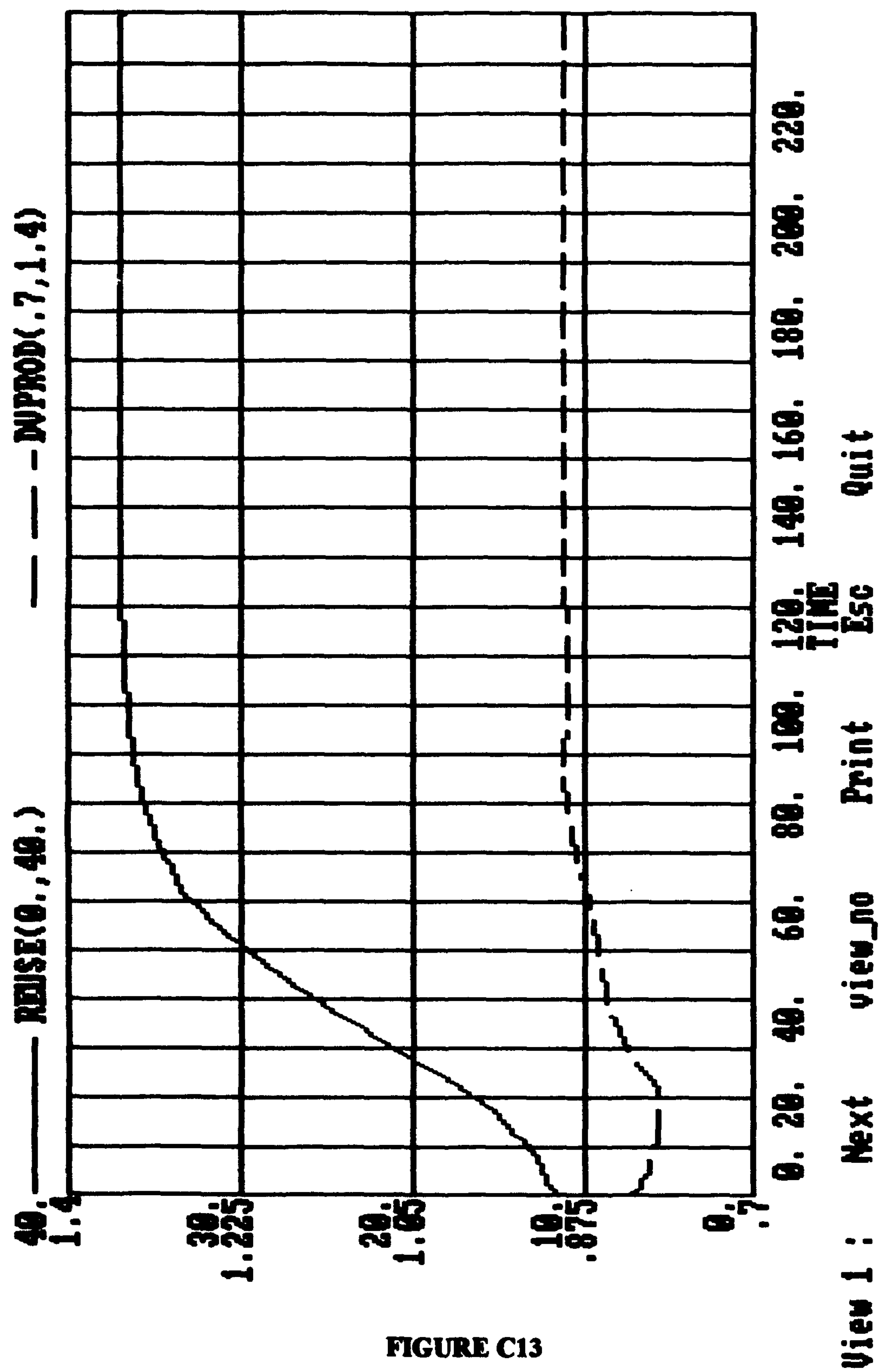

REUSE(0.,40.)
DUPROD(.7,1.4)
40.
1.4
30.
1.225
20.
1.05
10.
.875
0.
.7
0.
20.
40.
60.
80.
100.
120.
140.
160.
180.
200.
220.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE C13

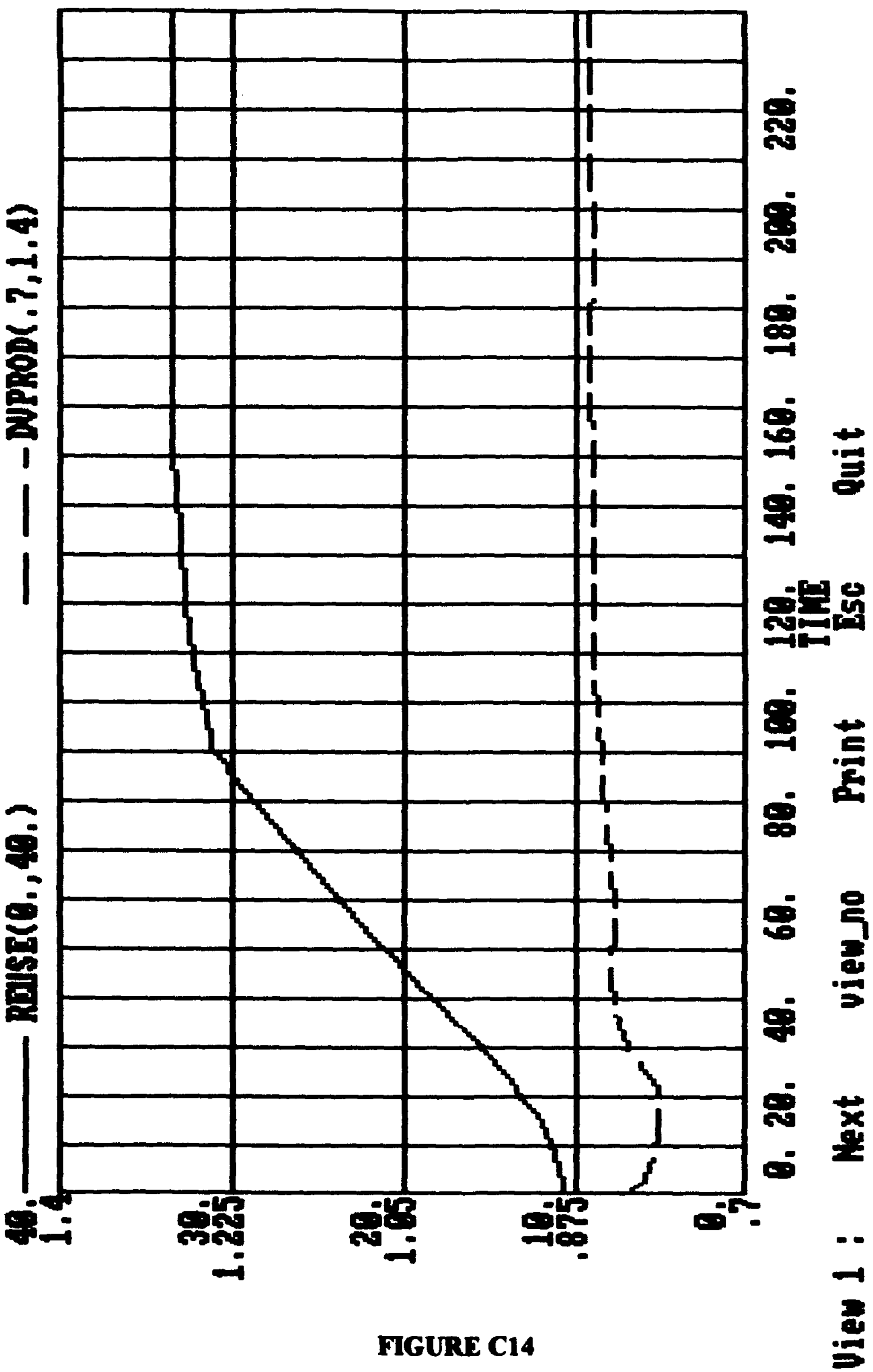

REUSE(0.,40.)
DUPROD(.7,1.4)
40.
1.4
30.
1.225
20.
1.05
10.
.875
0.
.7
0.
20.
40.
60.
80.
100.
120.
140.
160.
180.
200.
220.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE C14

TABLE C5

Model = REUSE6; Run = RCOST_1.RSL; Change = NMFRRU = 0.1

Years	DVPROD	REUSE	NMFRRU	CDVPRD
0.00	1.03	22.60	0.10	0.00
1.00	1.10	28.48	0.10	1.07
2.00	1.19	34.67	0.10	1.08
3.00	1.34	39.34	0.10	1.13
4.00	1.38	39.68	0.10	1.17
5.00	1.30	37.10	0.10	1.18
6.00	1.24	35.36	0.10	1.18
7.00	1.24	35.55	0.10	1.18
8.00	1.27	36.21	0.10	1.18
9.00	1.27	36.27	0.10	1.18
10.00	1.26	35.98	0.10	1.18
11.00	1.26	35.87	0.10	1.18
12.00	1.26	35.92	0.10	1.18
13.00	1.26	35.91	0.10	1.18
14.00	1.26	35.81	0.10	1.18
15.00	1.26	35.73	0.10	1.18
16.00	1.26	35.70	0.10	1.18
17.00	1.26	35.69	0.10	1.18
18.00	1.26	35.65	0.10	1.18
19.00	1.26	35.61	0.10	1.18
20.00	1.26	35.58	0.10	1.18

TABLE C6

Model = REUSE6; Run = RCOST_2.RSL; Change = NMFRRU = 0.2

Years	DVPROD	REUSE	NMFRRU	CDVPRD
0.00	0.99	22.60	0.20	0.00
1.00	1.04	28.14	0.20	1.02
2.00	1.12	34.39	0.20	1.03
3.00	1.26	39.08	0.20	1.07
4.00	1.29	39.63	0.20	1.10
5.00	1.24	37.45	0.20	1.12
6.00	1.19	35.78	0.20	1.12
7.00	1.19	35.83	0.20	1.12
8.00	1.21	36.39	0.20	1.12
9.00	1.21	36.47	0.20	1.12
10.00	1.20	36.22	0.20	1.12
11.00	1.20	36.10	0.20	1.12
12.00	1.20	36.13	0.20	1.12
13.00	1.20	36.12	0.20	1.12
14.00	1.20	36.04	0.20	1.12
15.00	1.20	35.96	0.20	1.12
16.00	1.20	35.93	0.20	1.12
17.00	1.20	35.91	0.20	1.12
18.00	1.20	35.87	0.20	1.12
19.00	1.20	35.83	0.20	1.12
20.00	1.20	35.80	0.20	1.12

TABLE C7

Model = REUSE6; Run = RCOST_3.RSL; Change = NMFRRU = 0.3

Years	DVPROD	REUSE	NMFRRU	CDVPRD
0.00	0.96	22.60	0.30	0.00
1.00	1.00	27.79	0.30	0.97
2.00	1.06	34.13	0.30	0.98
3.00	1.18	38.85	0.30	1.01
4.00	1.22	39.57	0.30	1.04
5.00	1.18	37.73	0.30	1.06
6.00	1.14	36.17	0.30	1.07
7.00	1.14	36.11	0.30	1.07
8.00	1.15	36.57	0.30	1.07
9.00	1.16	36.65	0.30	1.07
10.00	.15	36.44	0.30	1.07
11.00	15	36.32	0.30	1.07
12.00	15	36.33	0.30	1.07
13.00	15	36.33	0.30	1.07
14.00	1.15	36.25	0.30	1.07
15.00	1.15	36.18	0.30	1.07
16.00	1.15	36.14	0.30	1.07
17.00	1.15	36.12	0.30	1.07
18.00	1.15	36.08	0.30	1.07
19.00	1.15	36.05	0.30	1.07
20.00	1.15	36.02	0.30	1.07

TABLE C8

Model = REUSE6; Run = RCOST_4.RSL; Change = NMFRRU = 0.4

Years	DVPROD	REUSE	NMFRRU	CDVPRD
0.00	0.94	22.60	0.40	0.00
1.00	0.95	27.46	0.40	0.93
2.00	1.00	33.89	0.40	0.94
3.00	1.12	38.64	0.40	0.97
4.00	1.15	39.50	0.40	0.99
5.00	1.13	37.97	0.40	1.01
6.00	1.09	36.51	0.40	1.02
7.00	1.09	36.37	0.40	1.02
8.00	1.10	36.75	0.40	1.02
9.00	1.10	36.84	0.40	1.02
10.00	1.10	36.65	0.40	1.02
11.00	1.10	36.53	0.40	1.02
12.00	1.10	36.53	0.40	1.02
13.00	1.10	36.52	0.40	1.02
14.00	1.10	36.45	0.40	1.02
15.00	1.10	36.38	0.40	1.02
16.00	1.10	36.35	0.40	1.02
17.00	1.10	36.32	0.40	1.02
18.00	1.10	36.29	0.40	1.02
19.00	1.10	36.25	0.40	1.02
20.00	1.10	36.22	0.40	1.02

TABLE C9

Model = REUSE6; Run = RCOST_5.RSL; Change = NMFRRU = 0.5

Years	DVPROD	REUSE	NMFRRU	CDVPRD
0.00	0.91	22.09	0.50	0.00
1.00	0.91	27.15	0.50	0.90
2.00	0.96	33.66	0.50	0.90
3.00	1.06	38.45	0.50	0.92
4.00	1.09	39.43	0.50	0.95
5.00	1.08	38.17	0.50	0.96
6.00	1.05	36.84	0.50	0.97
7.00	1.04	36.64	0.50	0.97
8.00	1.05	36.95	0.50	0.97
9.00	1.06	37.03	0.50	0.97
10.00	1.05	36.87	0.50	0.98
11.00	1.05	36.75	0.50	0.98
12.00	1.05	36.73	0.50	0.98
13.00	1.05	36.72	0.50	0.98
14.00	1.05	36.66	0.50	0.98
15.00	1.05	36.60	0.50	0.98
16.00	1.05	36.56	0.50	0.98
17.00	1.05	36.53	0.50	0.98
18.00	1.05	36.50	0.50	0.98
19.00	1.05	36.46	0.50	0.98
20.00	1.05	36.43	0.50	0.98

TABLE C10

Model = REUSE6; Run = RCOST_6.RSL; Change = NMFRRU = 0.6

Years	DVPROD	REUSE	NMFRRU	CDVPRD
0.00	0.88	20.91	0.60	0.00
1.00	0.88	25.81	0.60	0.86
2.00	0.91	31.67	0.60	0.86
3.00	1.00	37.44	0.60	0.88
4.00	1.04	39.18	0.60	0.90
5.00	1.03	38.34	0.60	0.92
6.00	1.01	37.14	0.60	0.92
7.00	1.00	36.82	0.60	0.93
8.00	1.01	37.06	0.60	0.93
9.00	1.01	37.17	0.60	0.93
10.00	1.01	37.05	0.60	0.93
11.00	1.01	36.93	0.60	0.93
12.00	1.01	36.90	0.60	0.93
13.00	1.01	36.88	0.60	0.93
14.00	1.01	36.84	0.60	0.94
15.00	1.01	36.78	0.60	0.94
16.00	1.01	36.73	0.60	0.94
17.00	1.01	36.70	0.60	0.94
18.00	1.01	36.67	0.60	0.94
19.00	1.01	36.64	0.60	0.94
20.00	1.01	36.61	0.60	0.94

TABLE C11

Model = REUSE6; Run = RCOST_7.RSL; Change = NMFRRU = 0.7

Years	DVPROD	REUSE	NMFRRU	CDVPRD
0.00	0.86	20.07	0.70	0.00
1.00	0.84	21.69	0.70	0.83
2.00	0.87	25.40	0.70	0.83
3.00	0.93	30.60	0.70	0.84
4.00	0.96	34.47	0.70	0.86
5.00	0.97	35.08	0.70	0.87
6.00	0.96	35.11	0.70	0.88
7.00	0.96	34.99	0.70	0.88
8.00	0.96	35.14	0.70	0.88
9.00	0.96	35.45	0.70	0.89
10.00	0.96	35.56	0.70	0.89
11.00	0.96	35.50	0.70	0.89
12.00	0.96	35.40	0.70	0.89
13.00	0.96	35.35	0.70	0.89
14.00	0.96	35.31	0.70	0.89
15.00	0.96	35.26	0.70	0.89
16.00	0.96	35.20	0.70	0.89
17.00	0.96	35.15	0.70	0.89
18.00	0.96	35.11	0.70	0.89
19.00	0.96	35.07	0.70	0.89
20.00	0.96	35.03	0.70	0.89

TABLE C12

Model = REUSE6; Run = RCOST_8.RSL; Change = NMFRRU = 0.8

Years	DVPROD	REUSE	NMFRRU	CDVPRD
0.00	0.84	14.79	0.80	0.00
1.00	0.81	16.96	0.80	0.81
2.00	0.83	22.65	0.80	0.80
3.00	0.88	29.03	0.80	0.81
4.00	0.91	33.56	0.80	0.82
5.00	0.92	35.71	0.80	0.83
6.00	0.93	36.69	0.80	0.84
7.00	0.93	36.99	0.80	0.85
8.00	0.93	36.88	0.80	0.85
9.00	0.93	37.02	0.80	0.85
10.00	0.93	37.18	0.80	0.86
11.00	0.93	37.16	0.80	0.86
12.00	0.93	37.07	0.80	0.86
13.00	0.93	37.02	0.80	0.86
14.00	0.93	37.01	0.80	0.86
15.00	0.93	36.99	0.80	0.86
16.00	0.93	36.96	0.80	0.86
17.00	0.93	36.93	0.80	0.86
18.00	0.93	36.90	0.80	0.86
19.00	0.93	36.88	0.80	0.86
20.00	0.93	36.86	0.80	0.86

TABLE C13

Model = REUSE6; Run = RCOST_9.RSL; Change = NMFRRU = 0.9

Years	DVPROD	REUSE	NMFRRU	CDVPRD
0.00	0.83	11.67	0.90	0.00
1.00	0.80	13.84	0.90	0.79
2.00	0.81	18.24	0.90	0.79
3.00	0.85	24.04	0.90	0.79
4.00	0.86	29.03	0.90	0.80
5.00	0.87	32.96	0.90	0.81
6.00	0.89	34.85	0.90	0.81
7.00	0.89	36.02	0.90	0.82
8.00	0.89	36.56	0.90	0.82
9.00	0.89	36.82	0.90	0.82
10.00	0.89	37.04	0.90	0.83
11.00	0.90	37.20	0.90	0.83
12.00	0.90	37.18	0.90	0.83
13.00	0.90	37.11	0.90	0.83
14.00	0.90	37.07	0.90	0.83
15.00	0.90	37.06	0.90	0.83
16.00	0.90	37.05	0.90	0.83
17.00	0.90	37.03	0.90	0.83
18.00	0.90	37.01	0.90	0.83
19.00	0.90	36.99	0.90	0.83
20.00	0.90	36.98	0.90	0.83

TABLE C14

Model = REUSE6; Run = RCOST_10.RSL; Change = NMFRRU = 1.0

Years	DVPROD	REUSE	NMFRRU	CDVPRD
0.00	0.82	10.50	1.00	0.00
1.00	0.79	11.58	1.00	0.79
2.00	0.80	14.07	1.00	0.78
3.00	0.83	17.13	1.00	0.79
4.00	0.84	20.61	1.00	0.79
5.00	0.84	23.73	1.00	0.80
6.00	0.84	26.81	1.00	0.80
7.00	0.85	29.74	1.00	0.80
8.00	0.85	31.54	1.00	0.80
9.00	0.86	32.28	1.00	0.80
10.00	0.86	32.77	1.00	0.80
11.00	0.86	33.05	1.00	0.81
12.00	0.86	33.37	1.00	0.81
13.00	0.86	33.59	1.00	0.81
14.00	0.86	33.65	1.00	0.81
15.00	0.86	33.63	1.00	0.81
16.00	0.86	33.61	1.00	0.81
17.00	0.86	33.61	1.00	0.81
18.00	0.86	33.60	1.00	0.81
19.00	0.86	33.58	1.00	0.81
20.00	0.86	33.56	1.00	0.81

APPENDIX D: PRODUCTION COST DOCUMENTATION

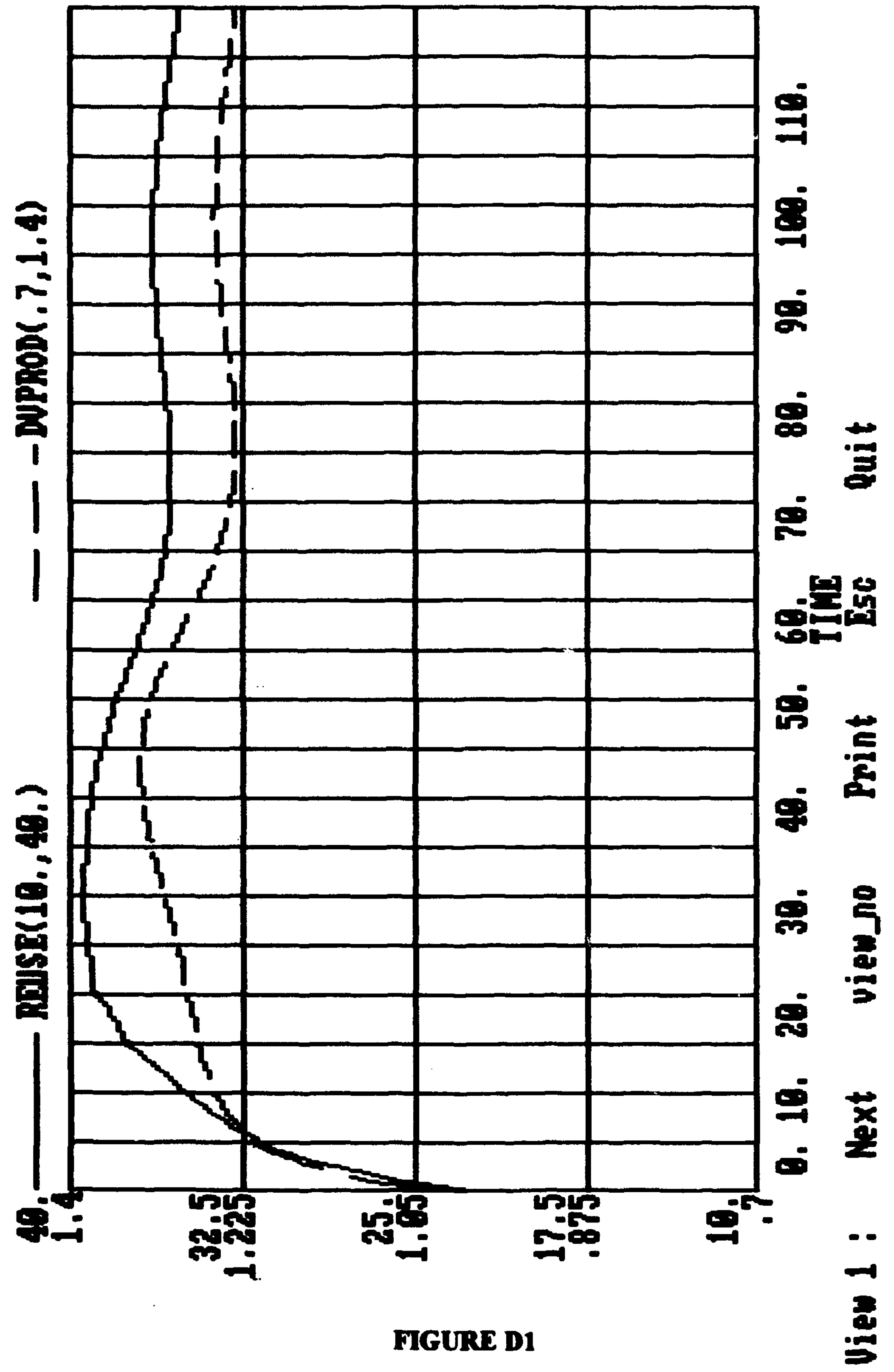
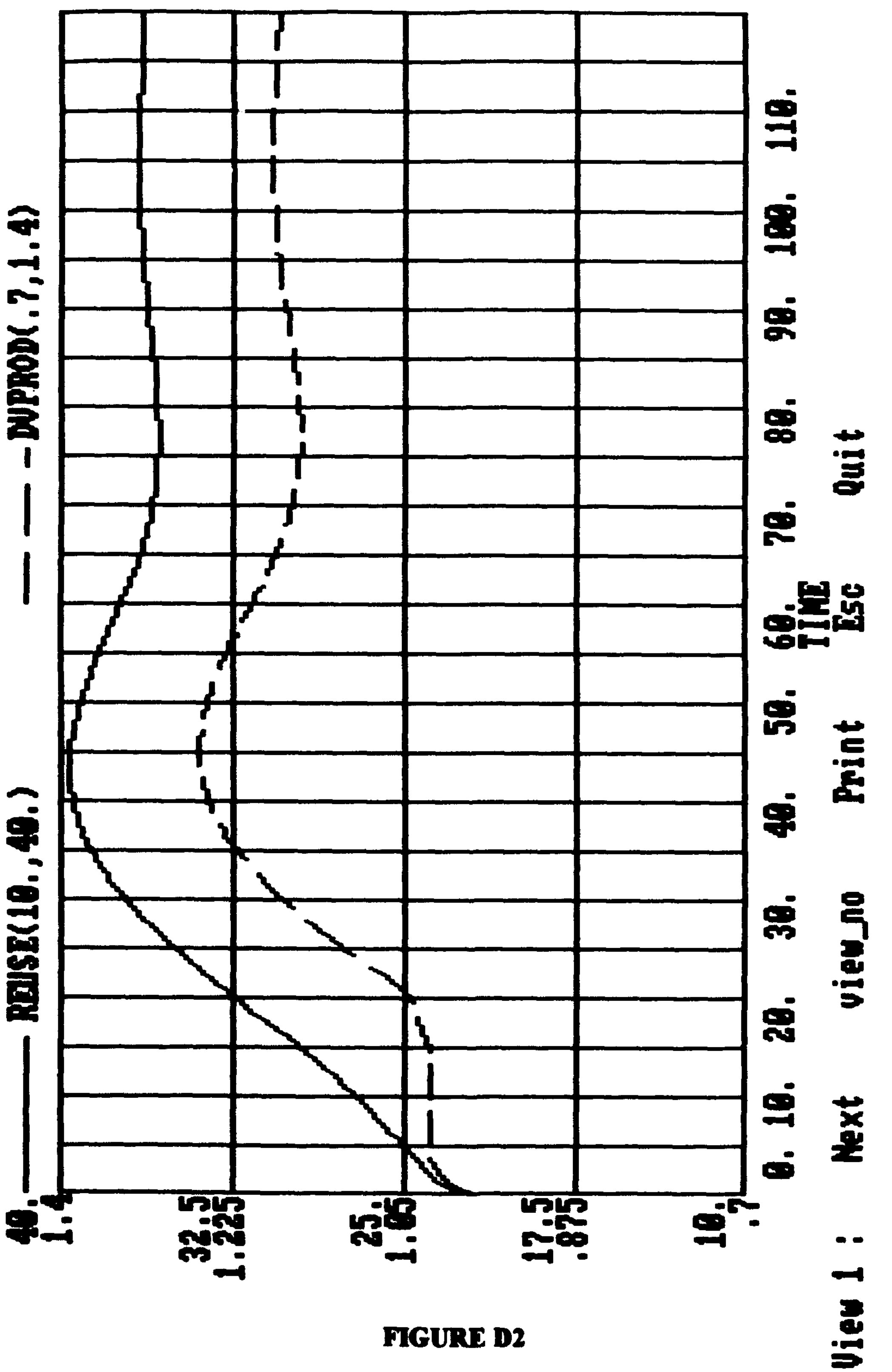


FIGURE D1



REUSE(10.,40.)
DUPROD(.7,1.4)
40.
1.4
32.5
1.225
25.
1.05
17.5
.875
10.
.7
0. 10. 20. 30. 40. 50. 60. 70. 80. 90. 100. 110.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE D2

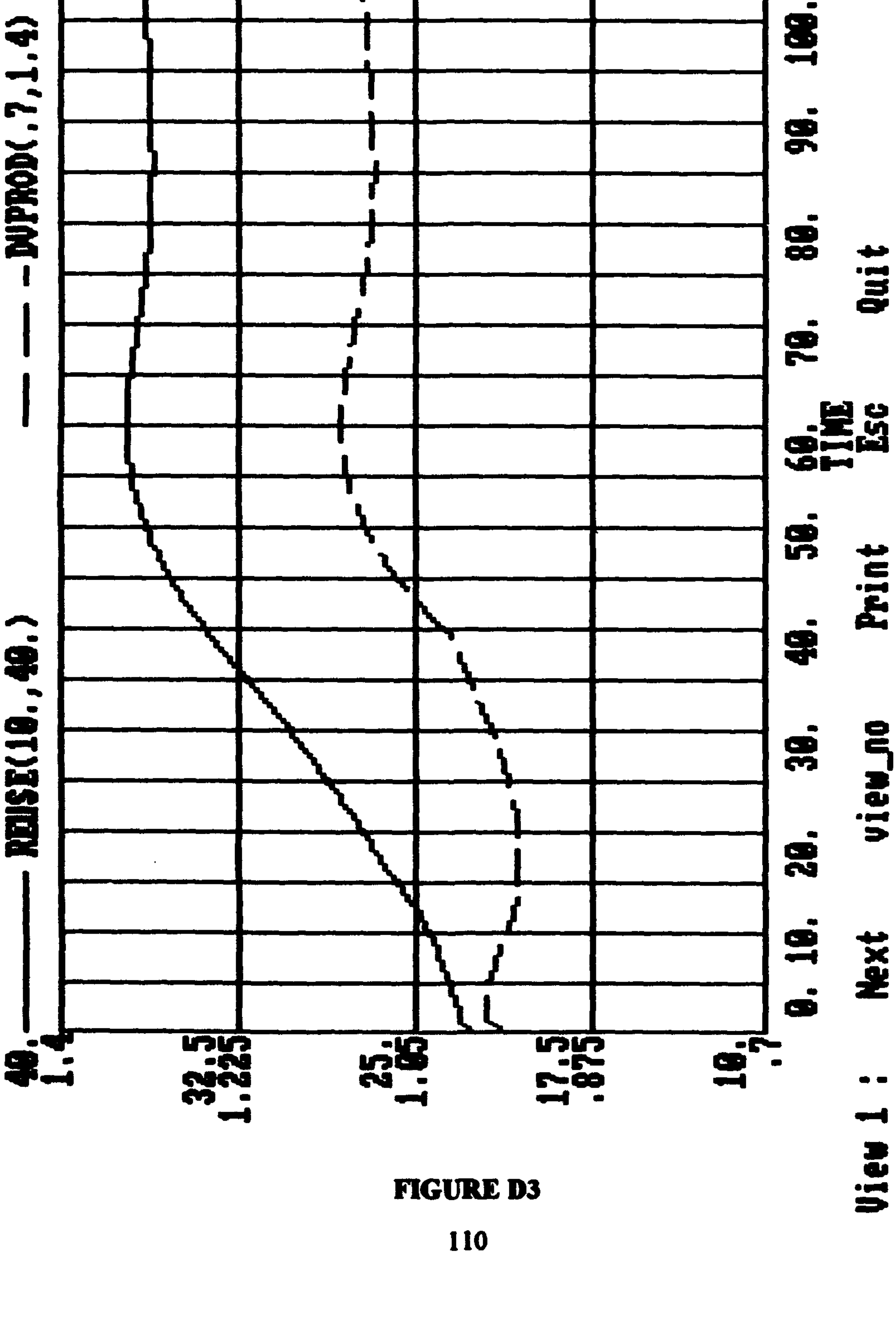

REUSE(10.,40.)
DUPROD(.7,1.4)
40.
1.4
32.5
1.225
25.
1.05
17.5
.875
10.
.7
0. 10. 20. 30. 40. 50. 60. 70. 80. 90. 100. 110.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE D3

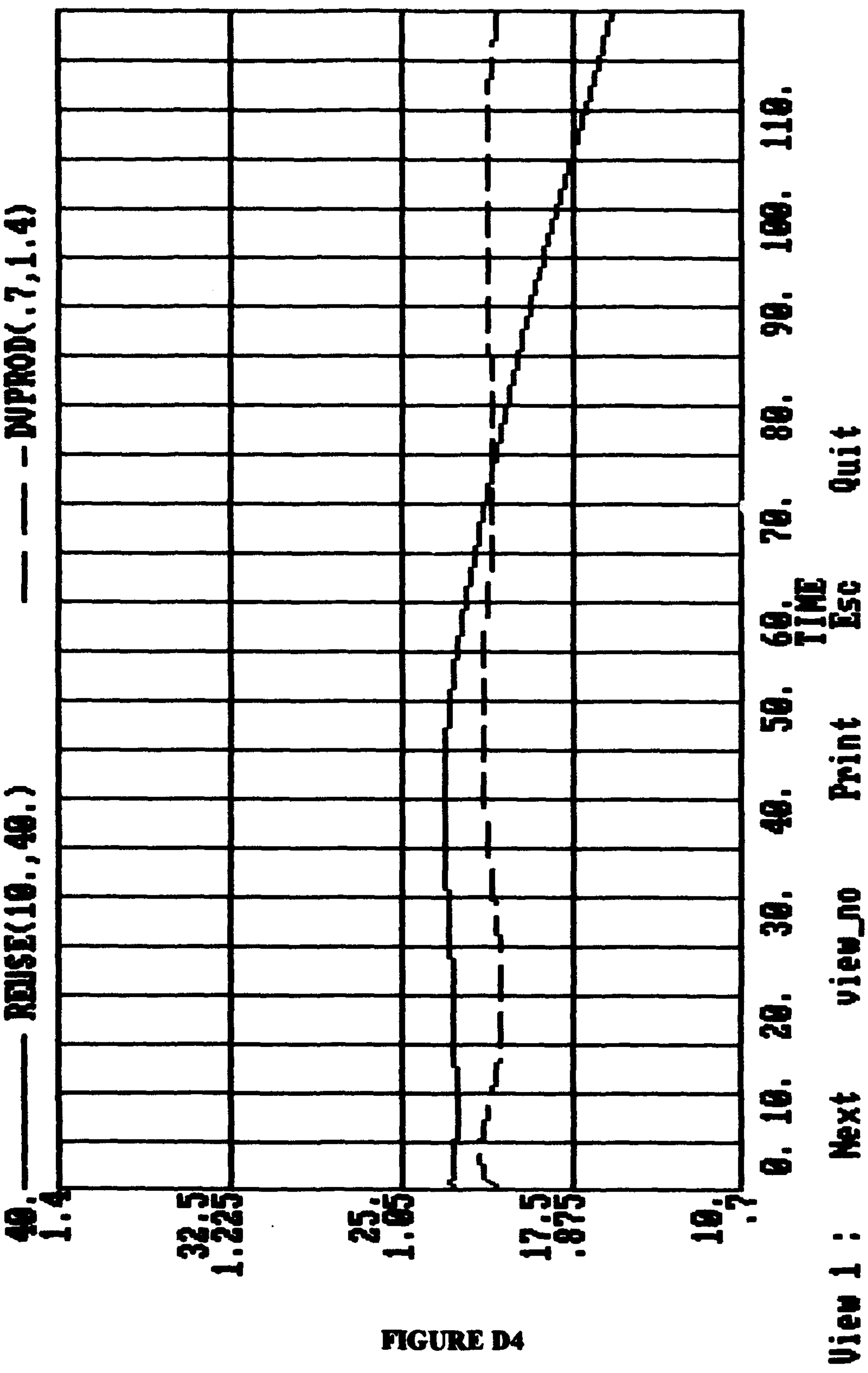

REUSE(10.,40.)
DVPROD(.7,1.4)
40.
1.4
32.5
1.225
25.
1.05
17.5
.875
10.
.7
0.
10.
20.
30.
40.
50.
60.
70.
80.
90.
100.
110.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE D4

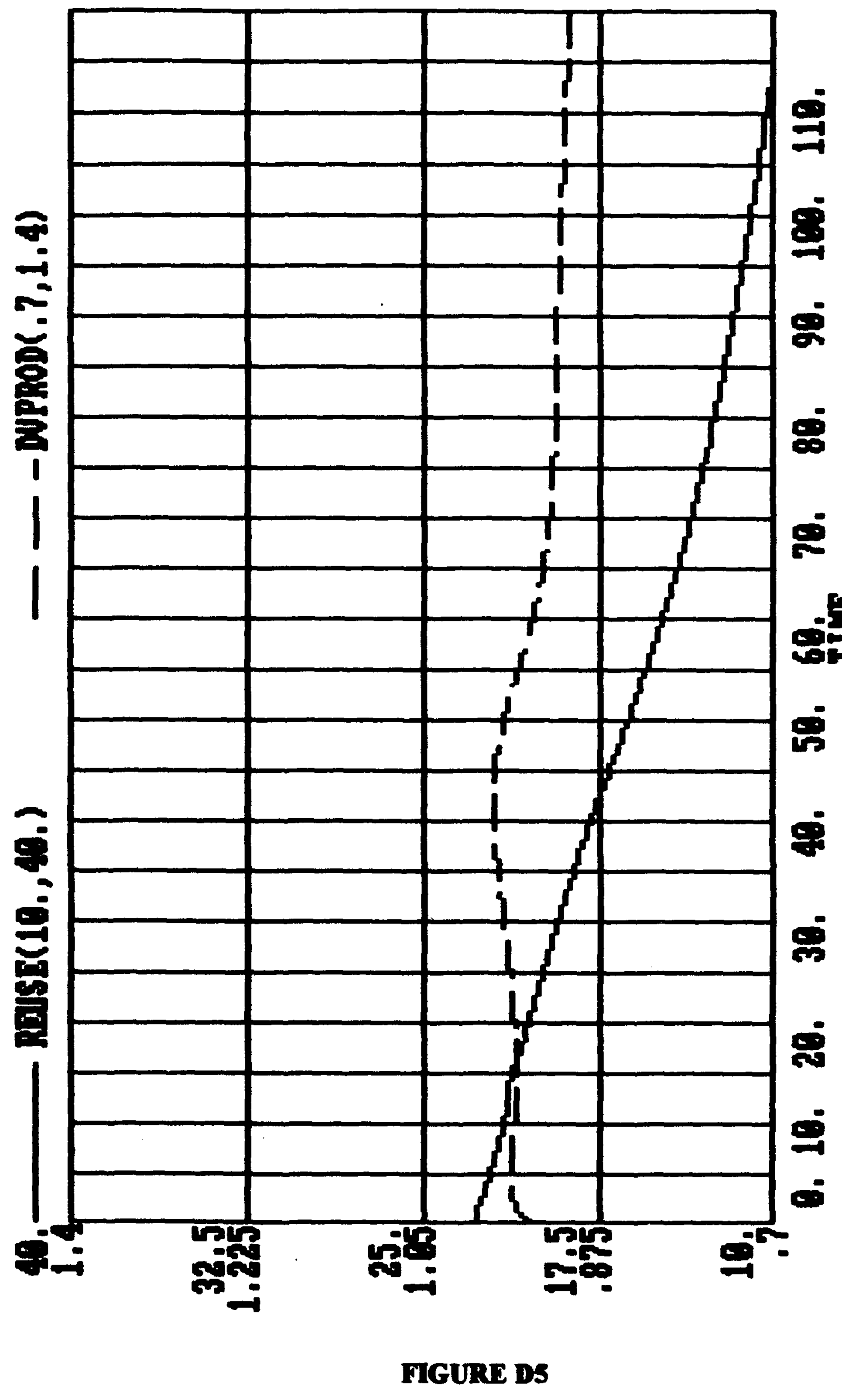

REUSE(10.,40.)
DUPROD(.7,1.4)
40.
1.4
32.5
1.225
25.
1.05
17.5
.875
10.
.7
0. 10. 20. 30. 40. 50. 60. 70. 80. 90. 100. 110.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE D5

TABLE D1

Model = REUSE6; Run = PCOST1.RSL; Change = NMEXTR = 1.00

Years	DVPROD	REUSE	NMEXTR	AVGUSE
0.00	1.03	22.60	1.00	2.00
1.00	1.26	36.09	1.00	2.72
2.00	1.29	39.27	1.00	3.78
3.00	1.32	39.37	1.00	4.74
4.00	1.33	38.35	1.00	5.41
5.00	1.27	36.50	1.00	5.85
6.00	1.24	35.67	1.00	6.15
7.00	1.24	36.08	1.00	6.42
8.00	1.25	36.49	1.00	6.68
9.00	1.25	36.15	1.00	6.93
10.00	1.23	35.34	1.00	7.16

TABLE D2

Model = REUSE6; Run = PCOST2.RSL; Change = NMEXTR = 2.00

Years	DVPROD	REUSE	NMEXTR	AVGUSE
0.00	0.98	22.60	2.00	2.00
1.00	1.02	28.08	2.00	2.76
2.00	1.10	34.62	2.00	3.38
3.00	1.23	38.98	2.00	4.25
4.00	1.26	39.46	2.00	5.07
5.00	1.21	37.46	2.00	5.64
6.00	1.16	35.96	2.00	6.02
7.00	1.16	36.01	2.00	6.32
8.00	1.18	36.55	2.00	6.59
9.00	1.19	36.65	2.00	6.82
10.00	1.18	36.45	2.00	7.00

TABLE D3

Model = REUSE6; Run = PCOST3.RSL; Change = NMEXTR = 3.00

Years	DVPROD	REUSE	NMEXTR	AVGUSE
0.00	0.96	22.60	3.00	2.00
1.00	0.95	24.89	3.00	2.86
2.00	0.95	28.42	3.00	3.49
3.00	1.00	32.53	3.00	4.03
4.00	1.09	36.14	3.00	4.59
5.00	1.12	37.29	3.00	5.20
6.00	1.11	36.71	3.00	5.70
7.00	1.09	36.23	3.00	6.08
8.00	1.10	36.37	3.00	6.40
9.00	1.10	36.53	3.00	6.66
10.00	1.10	36.42	3.00	6.88

TABLE D4

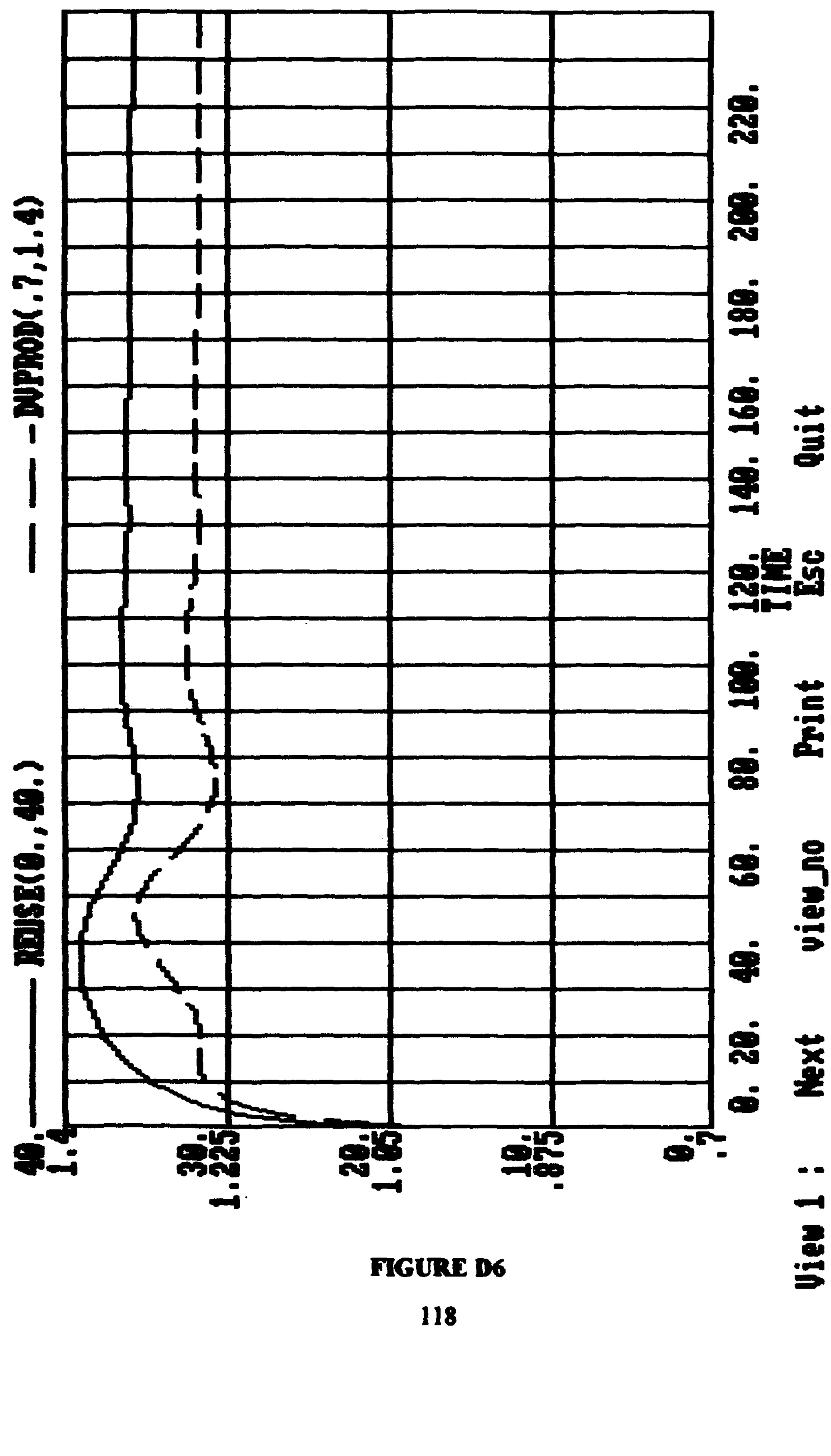
Model = REUSE6; Run = PCOST4.RSL; Change = NMEXTR = 4.00

Years	DVPROD	REUSE	NMEXTR	AVGUSE
0.00	0.95	22.60	4.00	2.00
1.00	0.95	22.59	4.00	3.01
2.00	0.95	22.83	4.00	3.83
3.00	0.96	23.13	4.00	4.53
4.00	0.97	22.98	4.00	5.13
5.00	0.96	22.21	4.00	5.67
6.00	0.96	21.15	4.00	6.19
7.00	0.96	19.99	4.00	6.70
8.00	0.96	18.69	4.00	7.21
9.00	0.96	17.23	4.00	7.71
10.00	0.95	15.75	4.00	8.18

TABLE D5

Model = REUSE6; Run = PCOST5.RSL; Change = NMEXTR = 5.00

Years	DVPROD	REUSE	NMEXTR	AVGUSE
0.00	0.94	22.60	5.00	2.00
1.00	0.96	21.44	5.00	3.09
2.00	0.96	20.04	5.00	4.09
3.00	0.98	18.46	5.00	5.03
4.00	0.97	16.60	5.00	5.90
5.00	0.94	14.81	5.00	6.64
6.00	0.92	13.37	5.00	7.25
7.00	0.92	12.25	5.00	7.77
8.00	0.91	11.31	5.00	8.19
9.00	0.91	10.48	5.00	8.51
10.00	0.90	9.77	5.00	8.74



REUSE(0.,40.)
DUPROD(.7,1.4)
40.
1.4
30.
1.225
20.
1.05
10.
.875
0.
.7
0. 20. 40. 60. 80. 100. 120. 140. 160. 180. 200. 220.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE D6

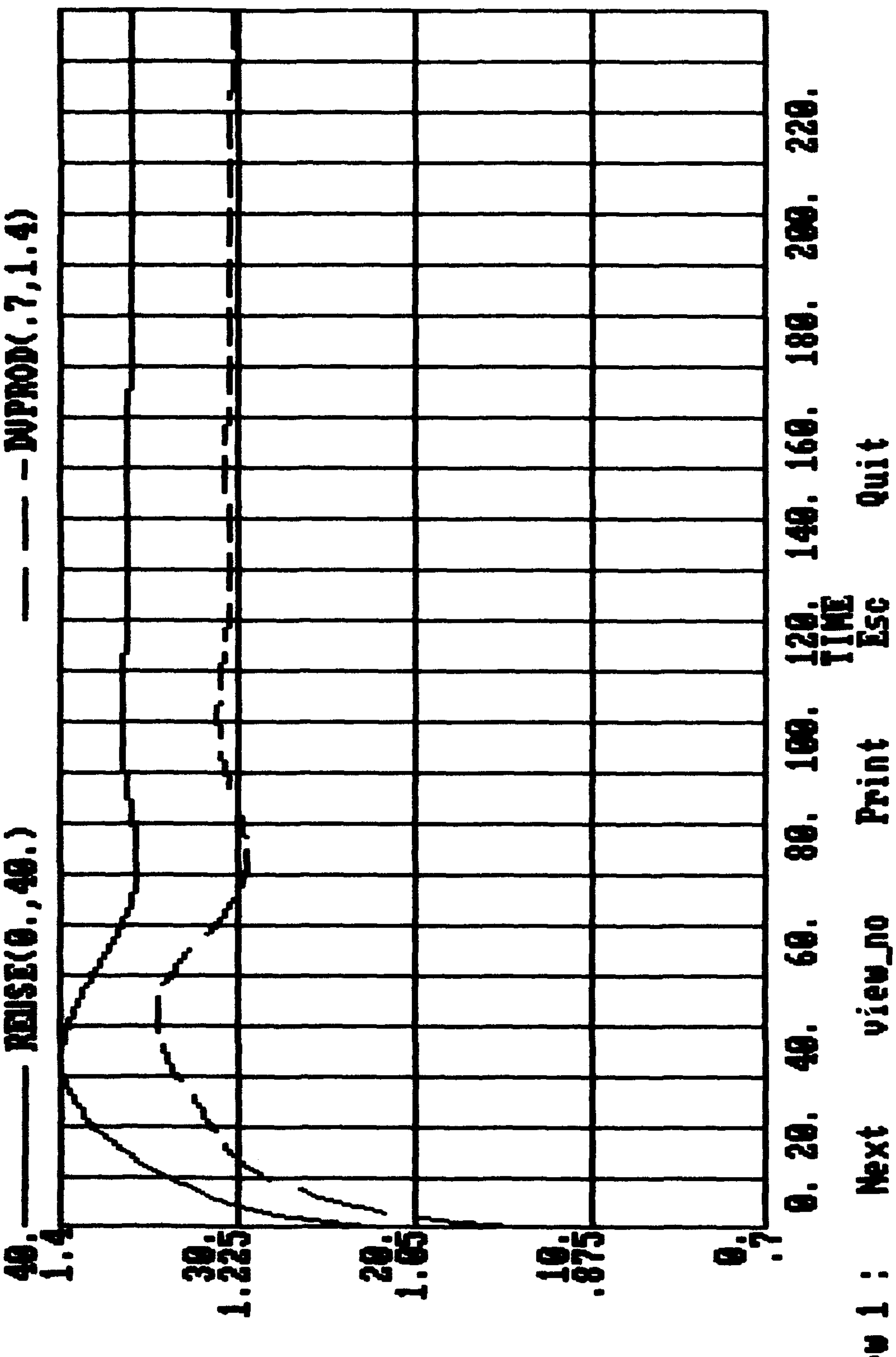

REUSE(0.,40.)
DUPROD(.7,1.4)
40.
1.4
30.
1.225
20.
1.05
10.
.875
0.
.7
0. 20. 40. 60. 80. 100. 120. 140. 160. 180. 200. 220.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE D7

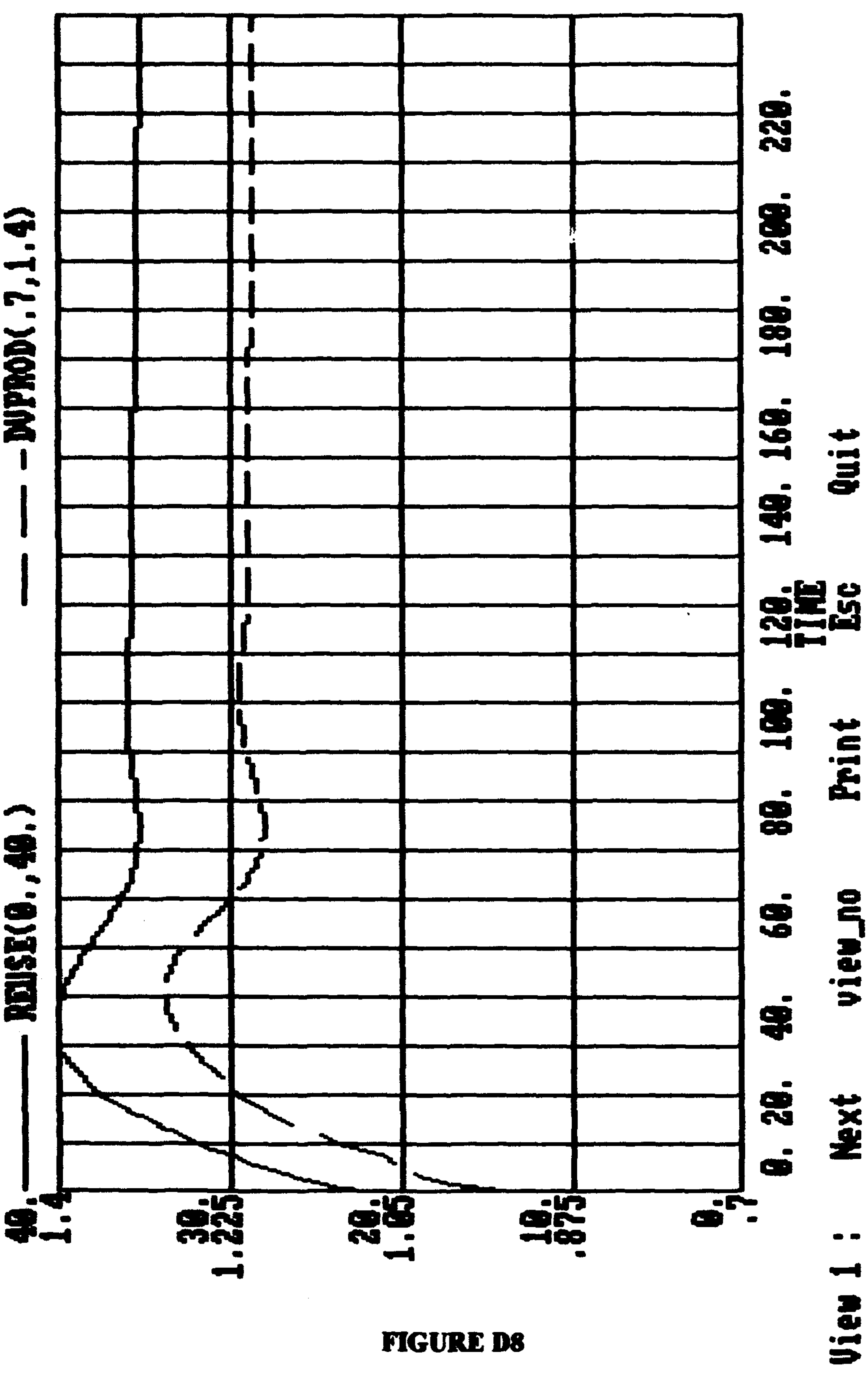

REUSE(0.,40.)
DUPROD(.7,1.4)
40.
1.4
30.
1.225
20.
1.05
10.
.875
0.
.7
0. 20. 40. 60. 80. 100. 120. 140. 160. 180. 200. 220.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE D8

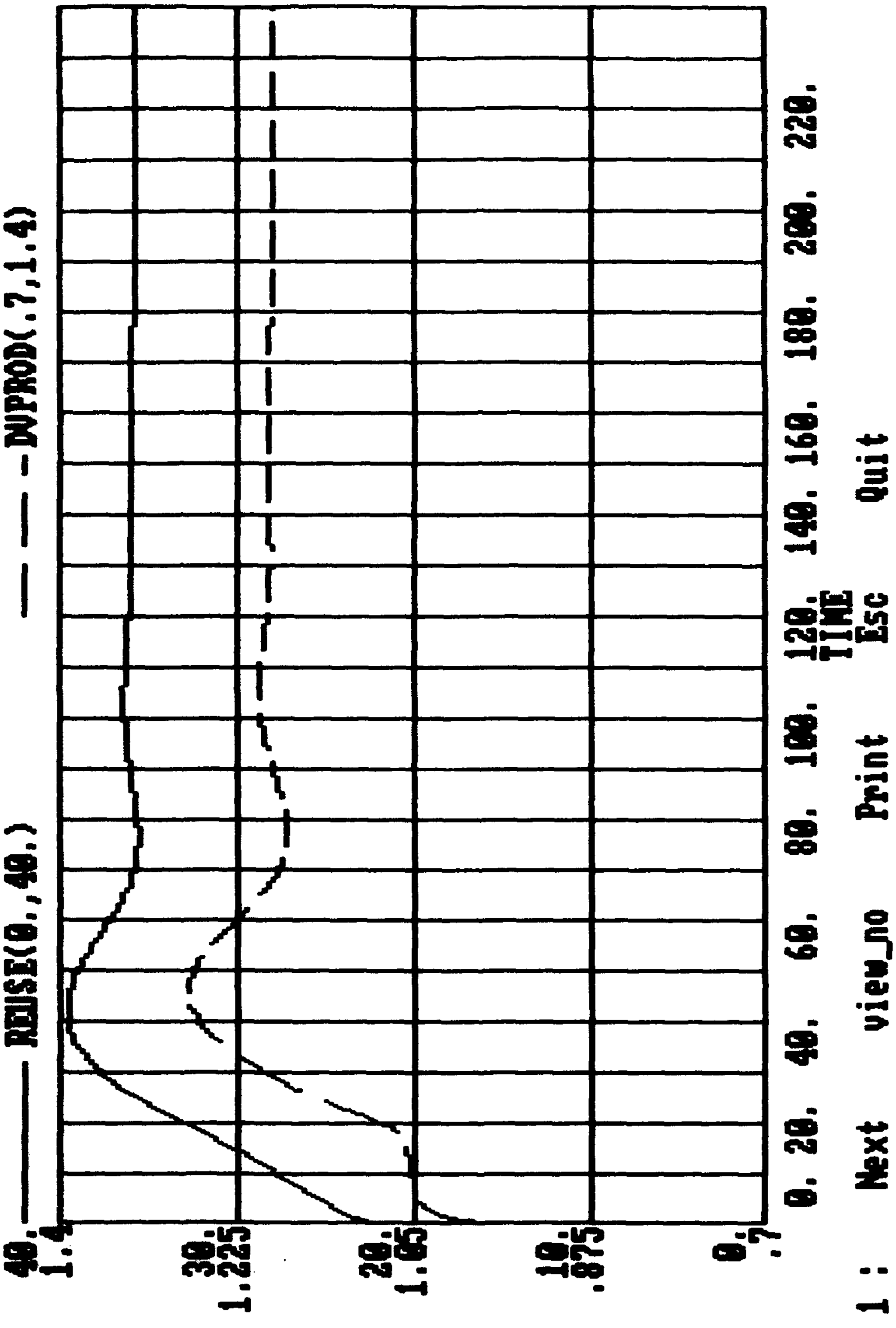

REUSE(0.,40.)
DUPROD(.7,1.4)
40.
1.4
30.
1.225
20.
1.05
10.
.875
0.
.7
0. 20. 40. 60. 80. 100. 120. 140. 160. 180. 200. 220.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE D9

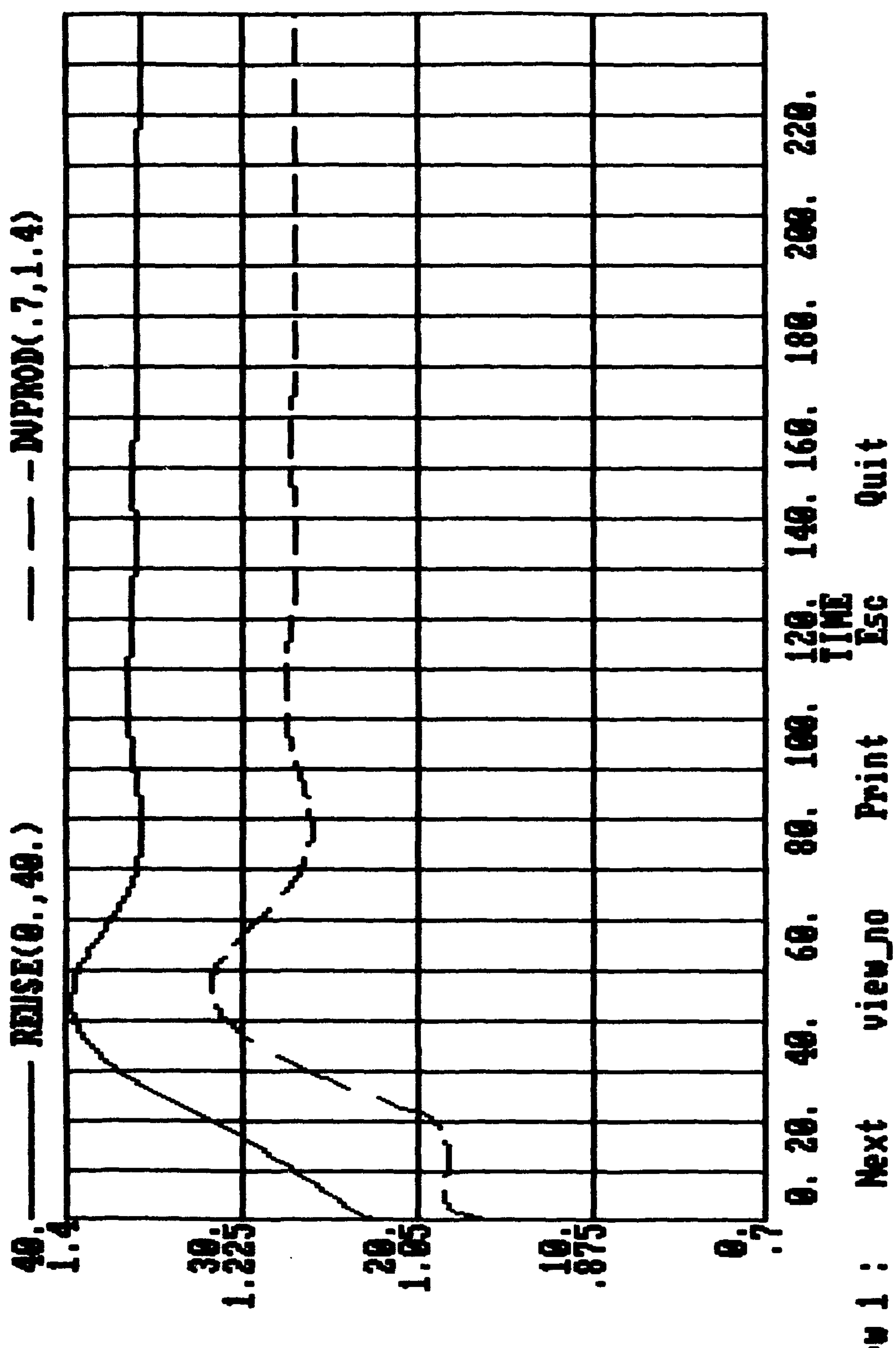

REUSE(0.,40.)
DUPROD(.7,1.4)
40. 1.4
30. 1.225
20. 1.05
10. .875
0. .7
0. 20. 40. 60. 80. 100. 120. 140. 160. 180. 200. 220.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE D10

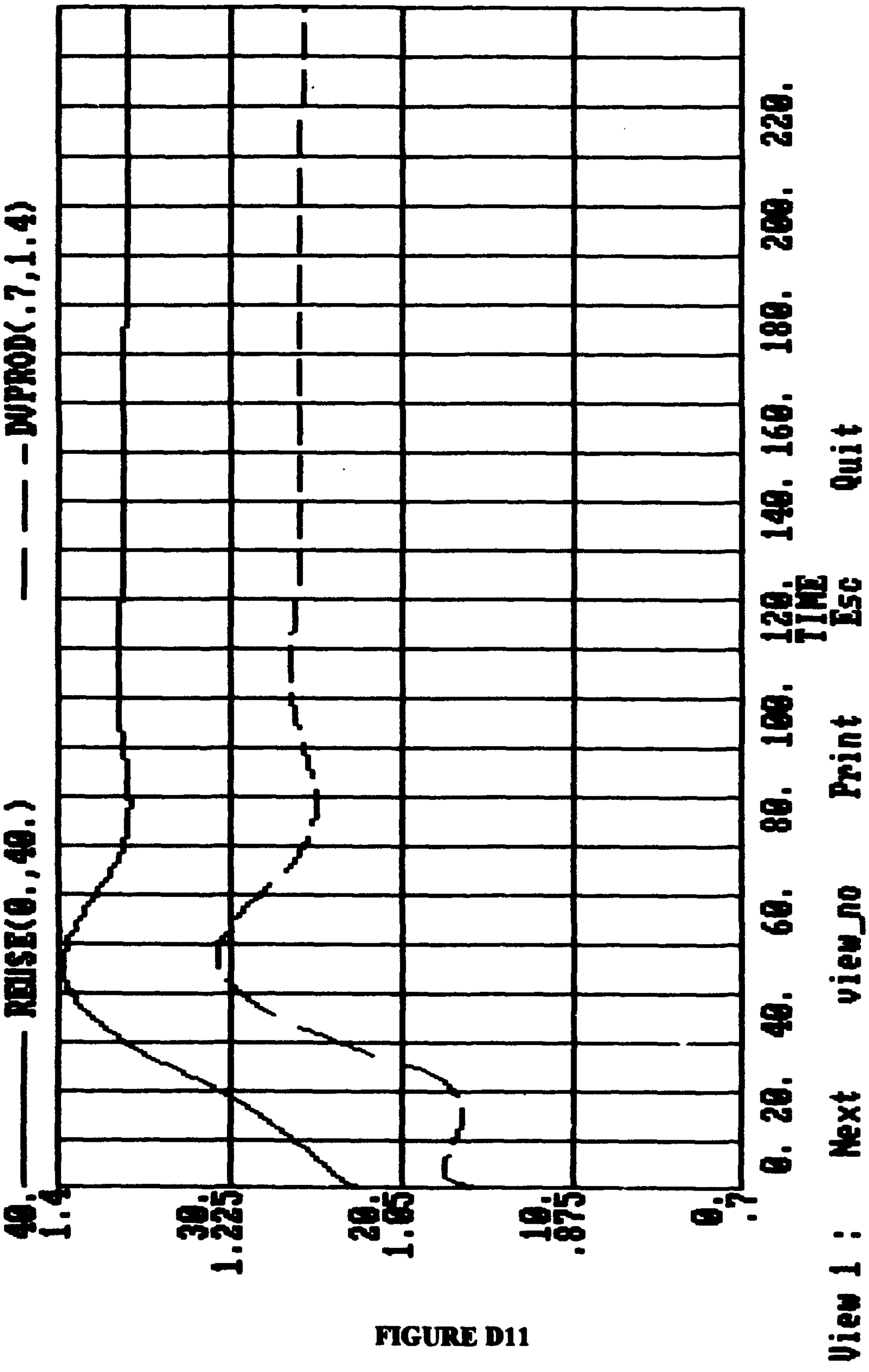

REUSE(0.,40.)
DUPROD(.7,1.4)
40. 1.4
30. 1.225
20. 1.05
10. .875
0. .7
0. 20. 40. 60. 80. 100. 120. 140. 160. 180. 200. 220.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE D11

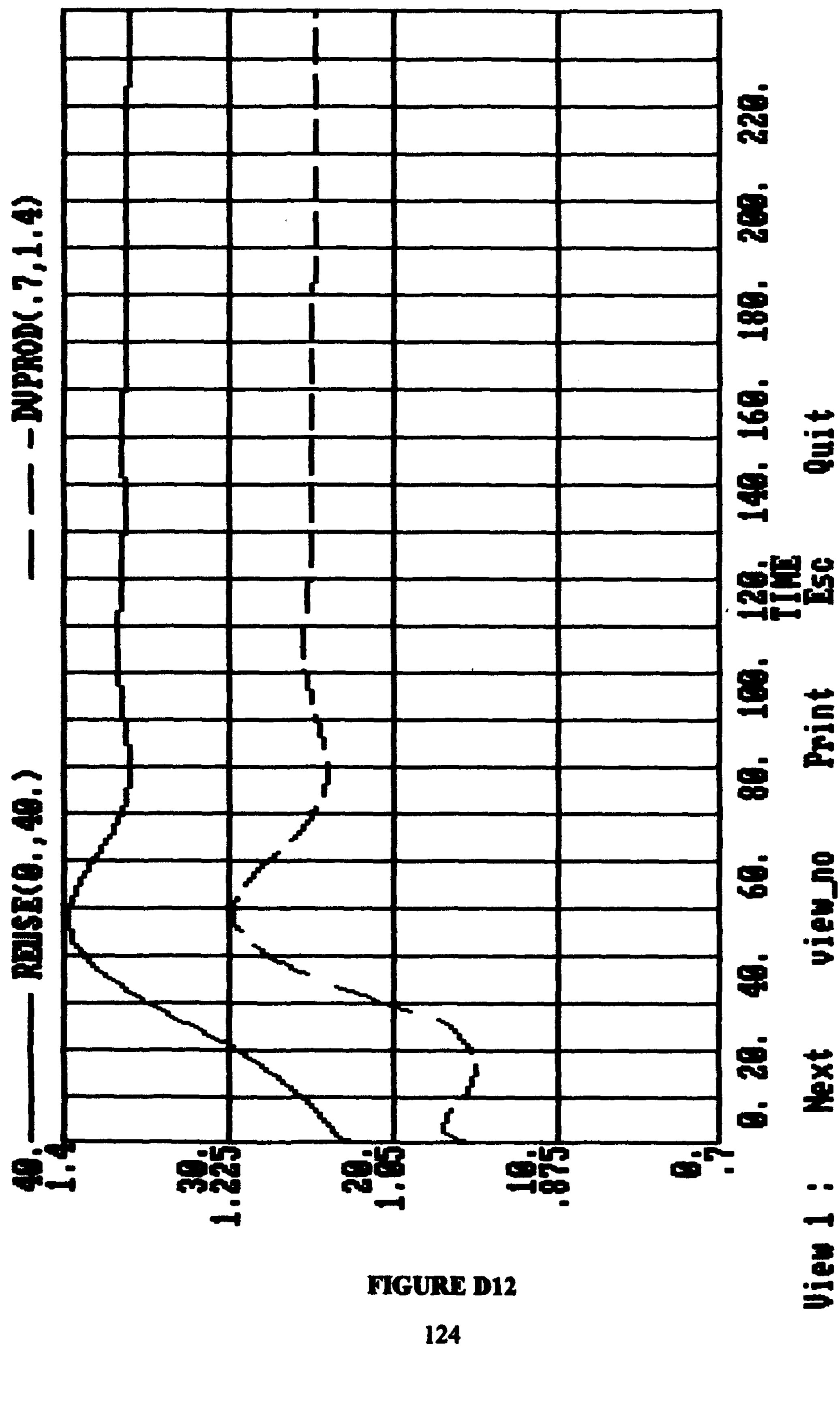

REUSE(0.,40.)
DUPROD(.7,1.4)
40.
1.4
30.
1.225
20.
1.05
10.
.875
0.
.7
0. 20. 40. 60. 80. 100. 120. 140. 160. 180. 200. 220.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE D12

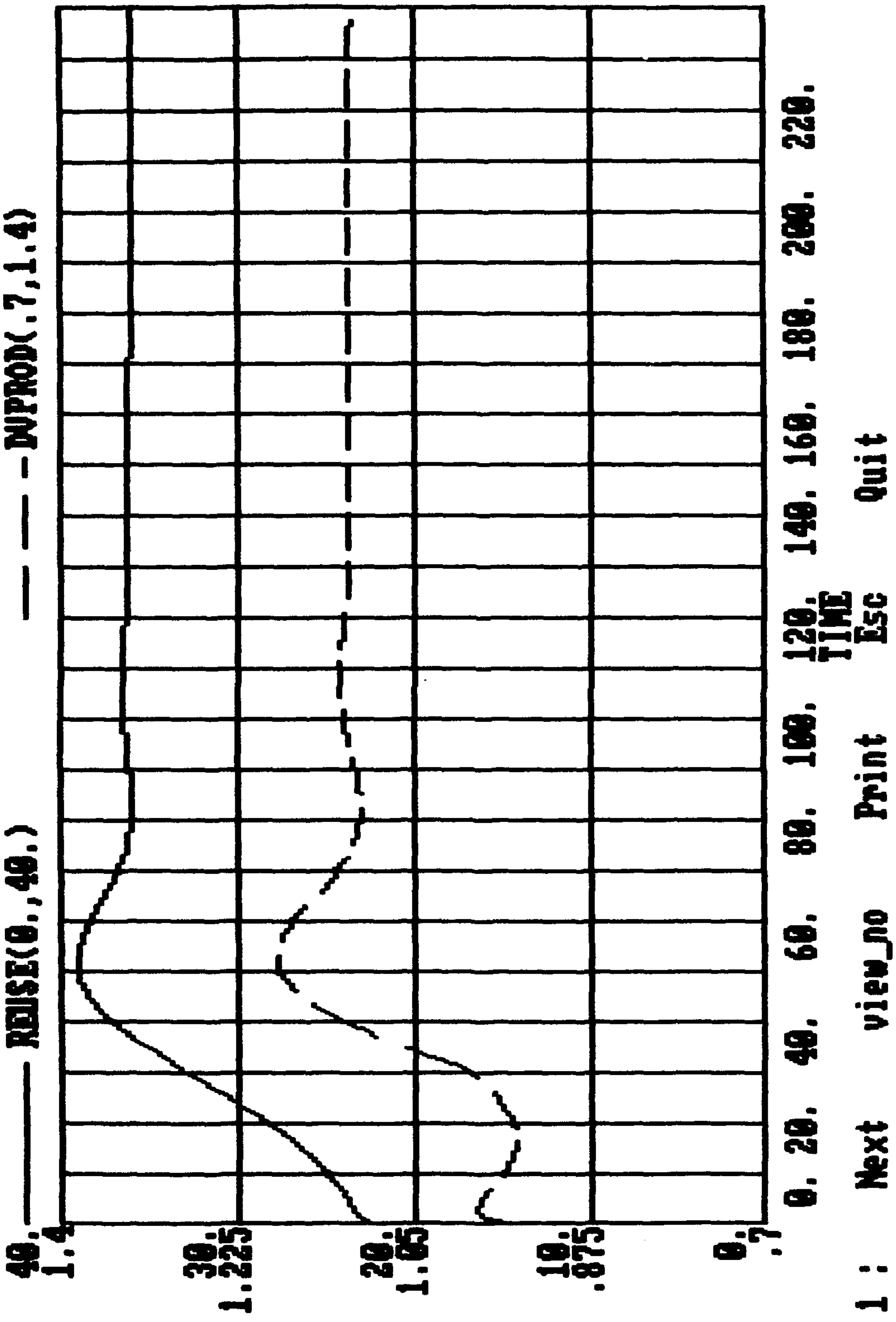

REUSE(0.,40.)
DVPROD(.7,1.4)
40.
1.4
30.
1.225
20.
1.05
10.
.875
0.
.7
0. 20. 40. 60. 80. 100. 120. 140. 160. 180. 200. 220.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE D13

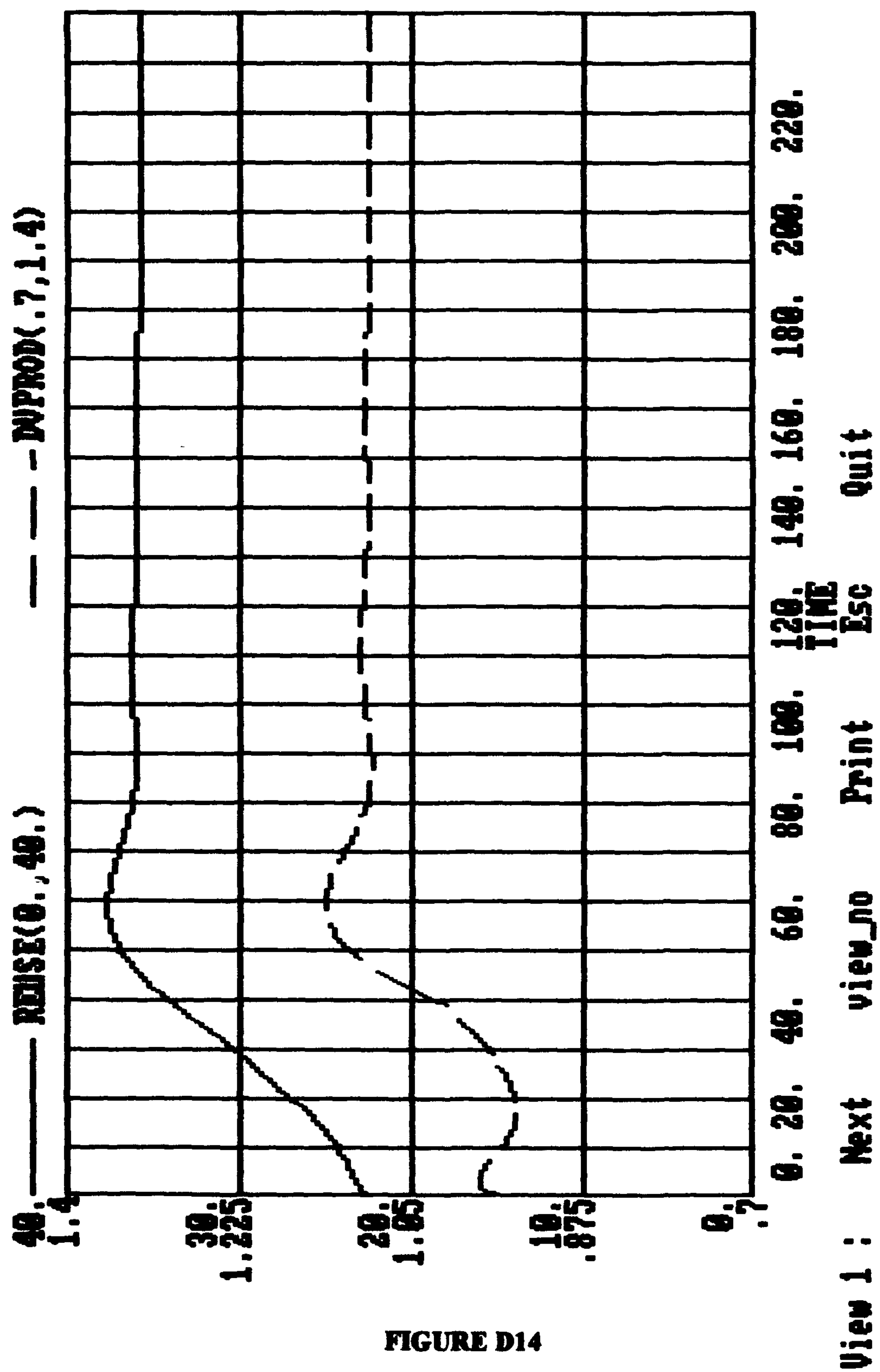

REUSE(0.,40.)
DUPROD(.7,1.4)
40. 1.4
30. 1.225
20. 1.05
10. .875
0. .7
0. 20. 40. 60. 80. 100. 120. 140. 160. 180. 200. 220.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE D14

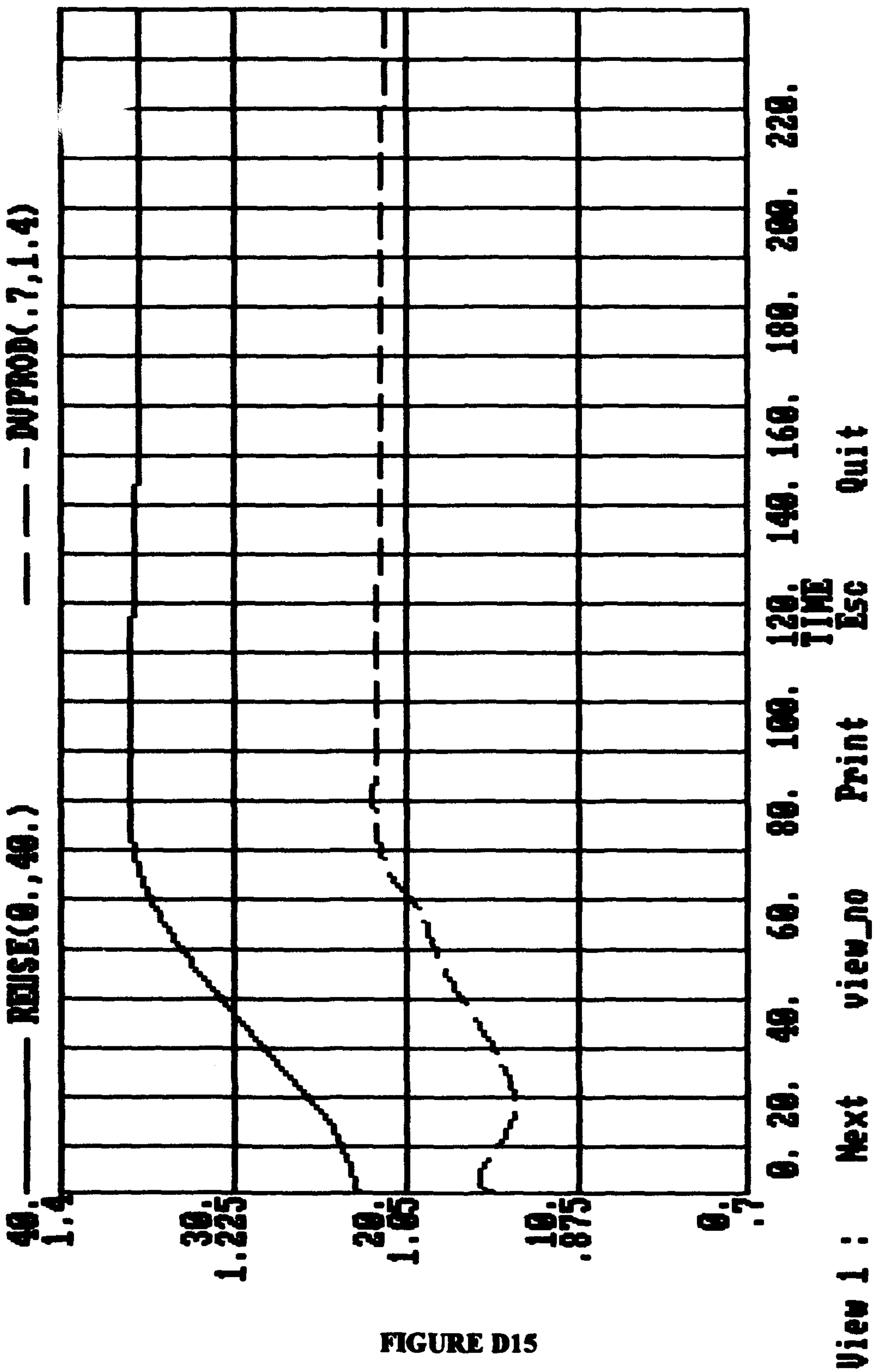
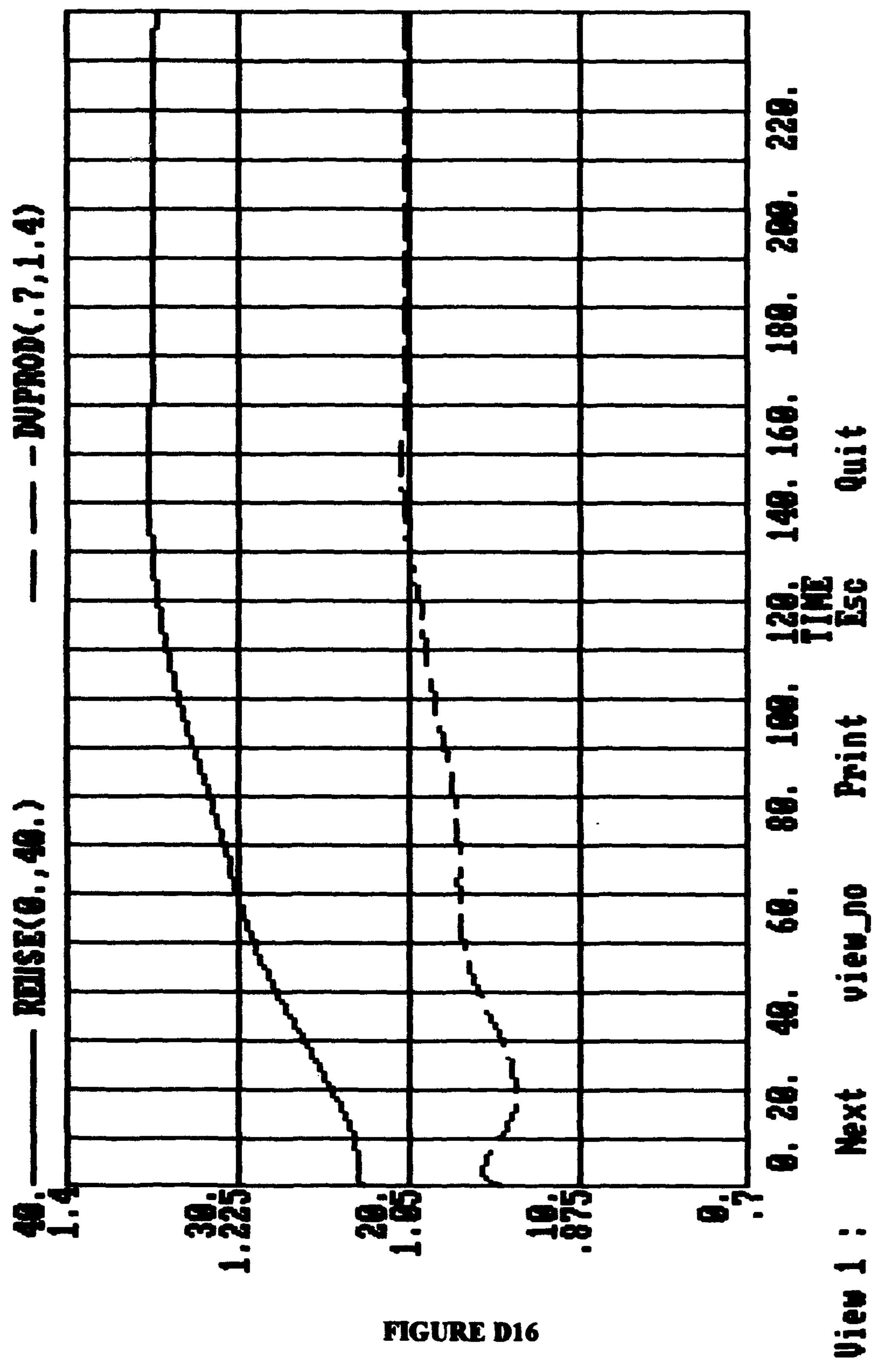

REUSE(0.,40.)
DUPROD(.7,1.4)
40.
1.4
30.
1.225
20.
1.05
10.
.875
0.
.7
0.
20.
40.
60.
80.
100.
120.
140.
160.
180.
200.
220.
TIME
View 1 :
Next
view_no
Print
Esc
Quit

FIGURE D15



REUSE(0.,40.)
INVPROD(.7,1.4)
40.
1.4
30.
1.225
20.
1.05
10.
.875
0.
.7
0.
20.
40.
60.
80.
100.
120.
140.
160.
180.
200.
220.
TIME
View 1 :
Next
view_no
Print
Esc
Quit

FIGURE D16

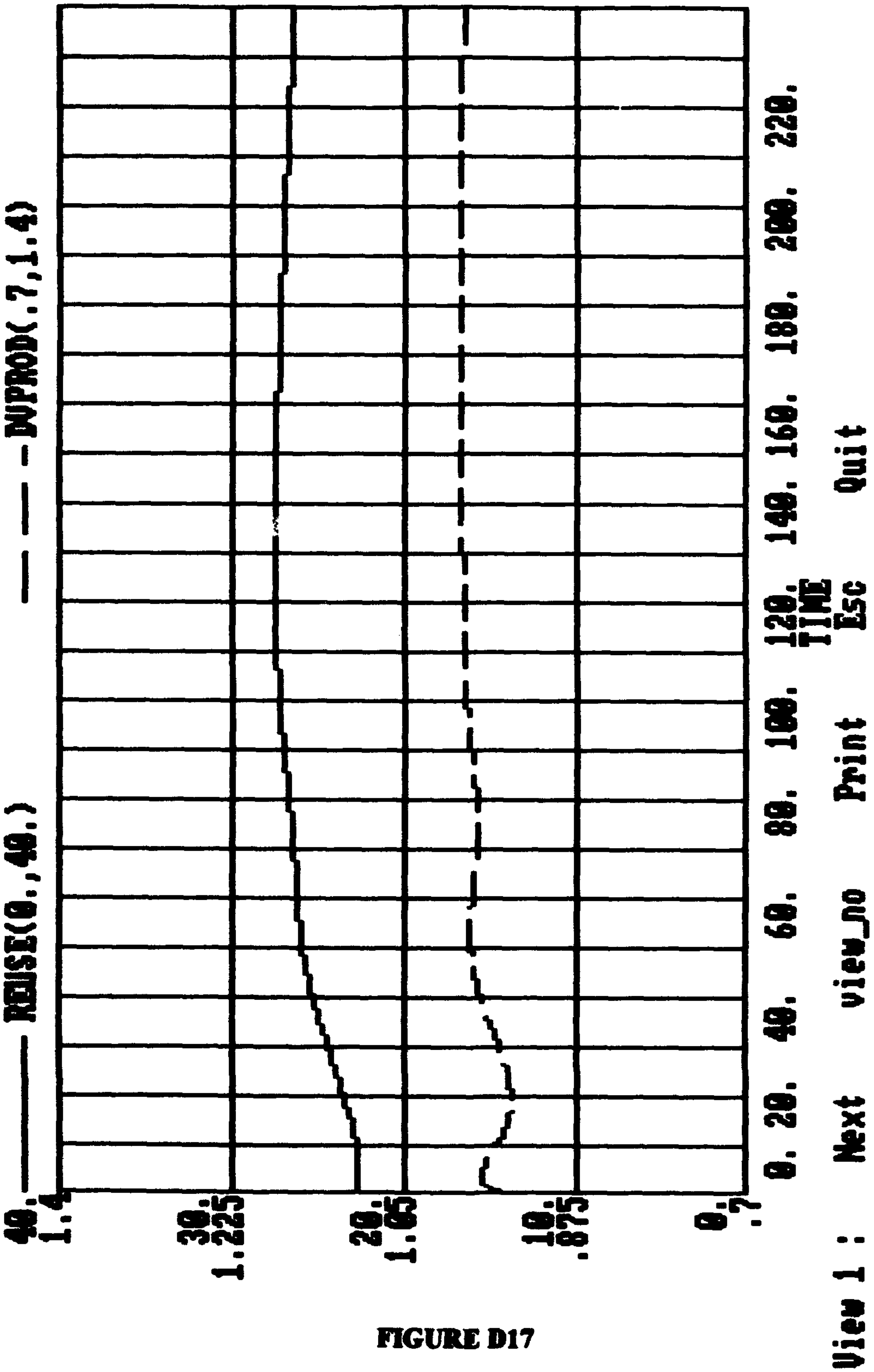

REUSE(0.,40.)
DUPROD(.7,1.4)
40.
1.4
30.
1.225
20.
1.05
10.
.875
0.
.7
0.
20.
40.
60.
80.
100.
120.
140.
160.
180.
200.
220.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE D17

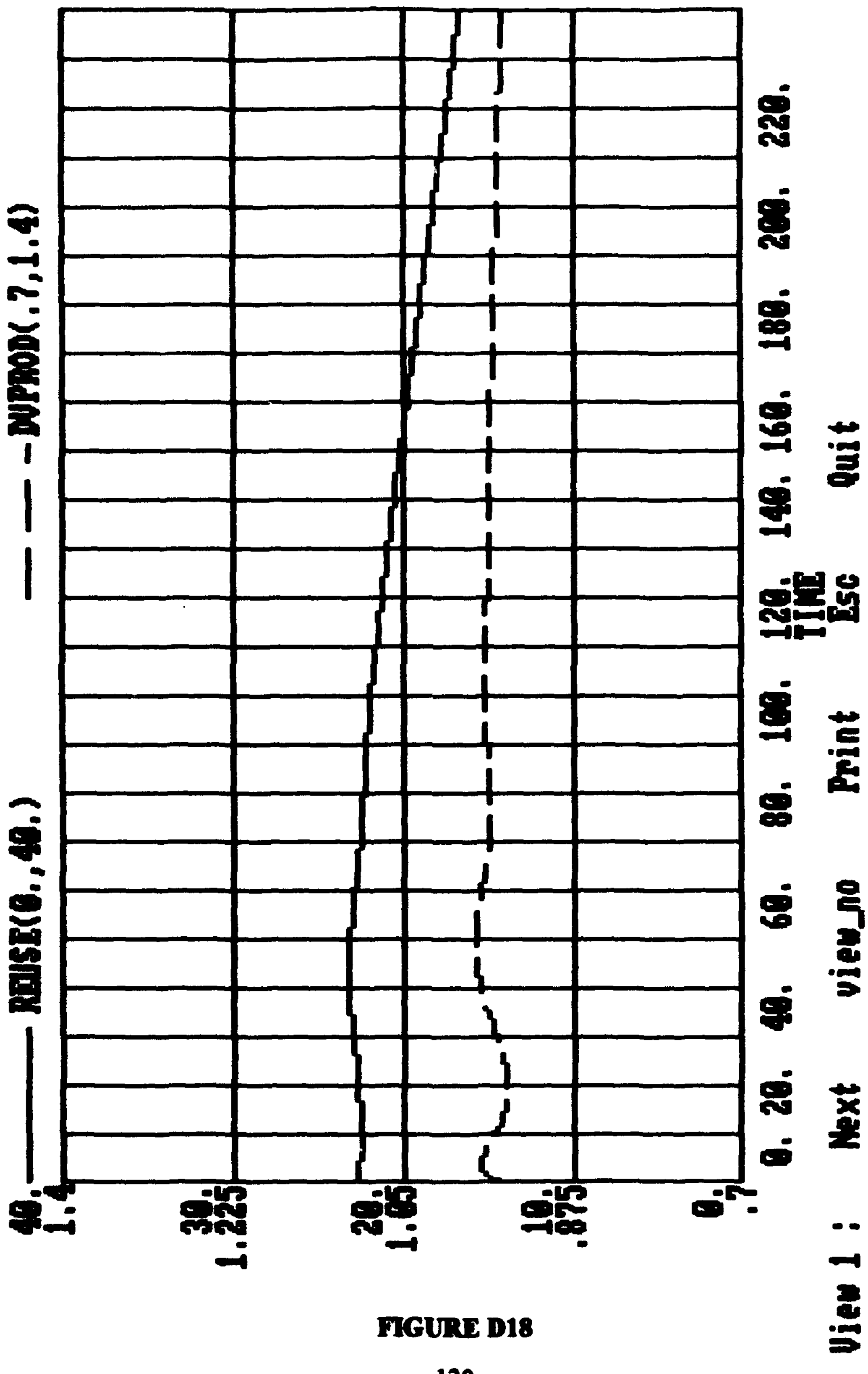

REUSE(0.,40.)
DUPROD(.7,1.4)
40.
1.4
30.
1.225
20.
1.05
10.
.875
0.
.7
0. 20. 40. 60. 80. 100. 120. 140. 160. 180. 200. 220.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE D18

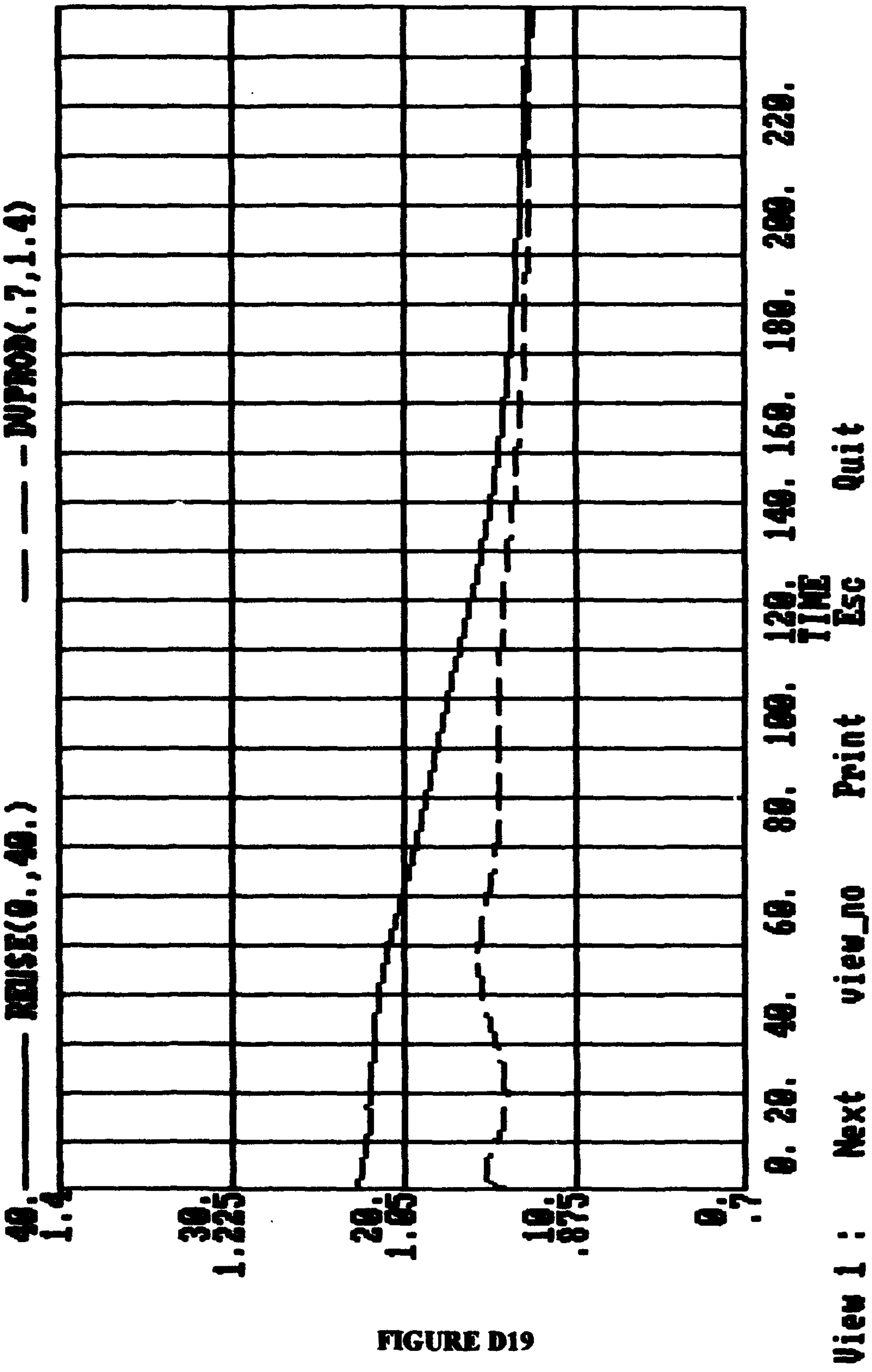

REUSE(0.,40.)
DUPROD(.7,1.4)
40.
1.4
30.
1.225
20.
1.05
10.
.875
0.
.7
0.
20.
40.
60.
80.
100.
120.
140.
160.
180.
200.
220.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE D19

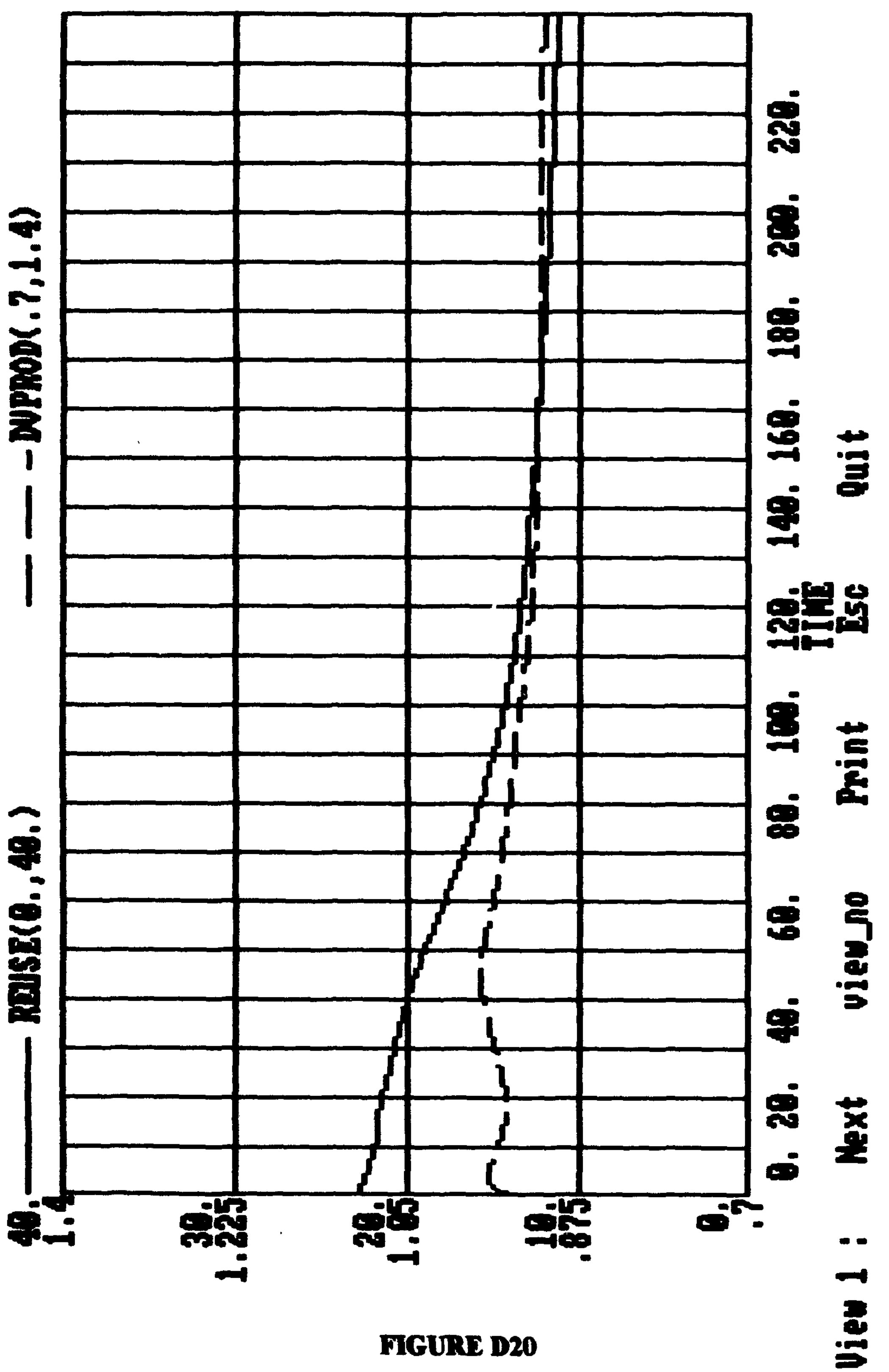

REUSE(0.,40.)
DUPROD(.7,1.4)
40.
1.4
30.
1.225
20.
1.05
10.
.875
0.
.7
0.
20.
40.
60.
80.
100.
120.
140.
160.
180.
200.
220.
TIME
View 1 :
Next
view_no
Print
Esc
Quit

FIGURE D20

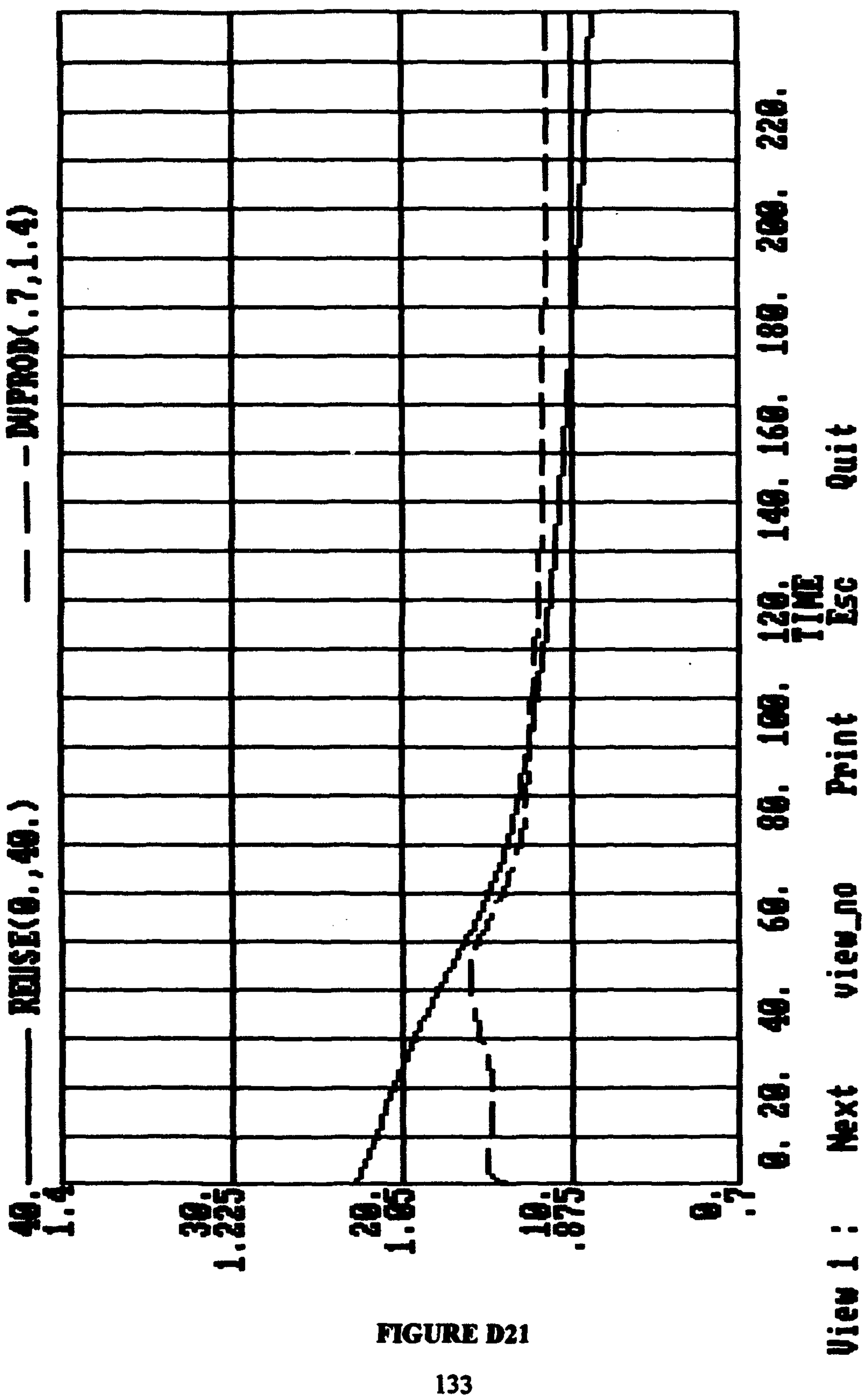

REUSE(0.,40.)
DVPROD(.7,1.4)
40. 1.4
30. 1.225
20. 1.05
10. .875
0. .7
0. 20. 40. 60. 80. 100. 120. 140. 160. 180. 200. 220.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE D21

TABLE D6

Model = REUSE5; Run = PC1.RSL; Change = NMEXTR = 1.0

Years	DVPROD	REUSE	NMEXTR	RPSTRY
0.00	1.03	22.60	1.00	2,000.00
1.00	1.26	35.71	1.00	3,132.00
2.00	1.26	38.28	1.00	3,178.67
3.00	1.30	39.22	1.00	3,120.11
4.00	1.32	38.50	1.00	3,070.36
5.00	1.28	36.65	1.00	3,040.73
6.00	1.24	35.70	1.00	3,035.71
7.00	1.25	36.07	1.00	3,045.19
8.00	1.27	36.61	1.00	3,050.67
9.00	1.27	36.58	1.00	3,048.60
10.00	1.26	36.31	1.00	3,045.80
11.00	1.26	36.22	1.00	3,045.83
12.00	1.26	36.26	1.00	3,046.53
13.00	1.26	36.24	1.00	3,045.99
14.00	1.26	36.14	1.00	3,044.99
15.00	1.26	36.07	1.00	3,044.58
16.00	1.26	36.05	1.00	3,044.59
17.00	1.26	36.02	1.00	3,044.43
18.00	1.26	35.99	1.00	3,044.07
19.00	1.26	35.95	1.00	3,043.77
20.00	1.26	35.92	1.00	3,043.61

TABLE D7

Model = REUSE5; Run = PC1_25.RSL; Change = NMEXTR = 1.25

Years	DVPROD	REUSE	NMEXTR	RPSTRY
0.00	0.95	22.60	1.25	2,000.00
1.00	1.22	35.25	1.25	3,124.11
2.00	1.27	39.11	1.25	3,197.56
3.00	1.30	40.03	1.25	3,134.40
4.00	1.30	38.64	1.25	3,072.15
5.00	1.25	36.63	1.25	3,040.46
6.00	1.22	35.80	1.25	3,036.43
7.00	1.23	36.20	1.25	3,045.37
8.00	1.24	36.65	1.25	3,049.97
9.00	1.24	36.60	1.25	3,048.09
10.00	1.24	36.37	1.25	3,045.94
11.00	1.23	36.30	1.25	3,046.13
12.00	1.24	36.33	1.25	3,046.60
13.00	1.24	36.30	1.25	3,045.97
14.00	1.24	36.21	1.25	3,045.08
15.00	1.23	36.14	1.25	3,044.74
16.00	1.23	36.12	1.25	3,044.72
17.00	1.23	36.09	1.25	3,044.53
18.00	1.23	36.05	1.25	3,044.19
19.00	1.23	36.02	1.25	3,043.92
20.00	1.23	35.99	1.25	3,043.77

TABLE D8

Model = REUSE5; Run = PC1_50.RSL; Change = NMEXTR = 1.50

Years	DVPROD	REUSE	NMEXTR	RPSTRY
0.00	0.95	22.60	1.50	2,000.00
1.00	1.14	33.45	1.50	3,008.01
2.00	1.24	38.78	1.50	3,178.98
3.00	1.29	40.29	1.50	3,139.78
4.00	1.28	38.80	1.50	3,068.63
5.00	1.23	36.51	1.50	3,022.89
6.00	1.19	35.45	1.50	3,007.06
7.00	1.20	35.71	1.50	3,009.08
8.00	1.22	36.12	1.50	3,011.92
9.00	1.22	36.08	1.50	3,009.64
10.00	1.21	35.85	1.50	3,006.23
11.00	1.21	35.77	1.50	3,004.98
12.00	1.21	35.78	1.50	3,004.75
13.00	1.21	35.74	1.50	3,003.88
14.00	1.21	35.65	1.50	3,002.61
15.00	1.21	35.58	1.50	3,001.72
16.00	1.21	35.55	1.50	3,001.23
17.00	1.21	35.52	1.50	3,000.72
18.00	1.20	35.48	1.50	3,000.07
19.00	1.20	35.44	1.50	2,999.46
20.00	1.20	35.40	1.50	2,998.93

TABLE D9

Model = REUSE5; Run = PC1_75.RSL; Change = NMEXTR = 1.75

Years	DVPROD	REUSE	NMEXTR	RPSTRY
0.00	0.99	22.60	1.75	2,000.00
1.00	1.05	28.85	1.75	2,559.12
2.00	1.14	35.32	1.75	3,019.06
3.00	1.24	39.26	1.75	3,126.84
4.00	1.27	39.38	1.75	3,090.06
5.00	1.23	37.30	1.75	3,042.91
6.00	1.18	35.81	1.75	3,022.16
7.00	1.18	35.89	1.75	3,025.25
8.00	1.20	36.40	1.75	3,031.03
9.00	1.20	36.46	1.75	3,030.38
10.00	1.20	36.23	1.75	3,027.41
11.00	1.19	36.11	1.75	3,026.50
12.00	1.19	36.13	1.75	3,026.84
13.00	1.20	36.12	1.75	3,026.42
14.00	1.19	36.04	1.75	3,025.34
15.00	1.19	35.96	1.75	3,024.62
16.00	1.19	35.93	1.75	3,024.36
17.00	1.19	35.91	1.75	3,024.10
18.00	1.19	35.87	1.75	3,023.66
19.00	1.19	35.83	1.75	3,023.22
20.00	1.19	35.80	1.75	3,022.90

TABLE D10

Model = REUSE5; Run = PC2.RSL; Change = NMEXTR = 2.0

Years	DVPROD	REUSE	NMEXTR	RPSTRY
0.00	0.98	22.60	2.00	2,000.00
1.00	1.02	27.96	2.00	2,479.16
2.00	1.09	34.26	2.00	2,952.78
3.00	1.22	38.96	2.00	3,118.07
4.00	1.26	39.60	2.00	3,096.42
5.00	1.21	37.60	2.00	3,049.40
6.00	1.16	35.98	2.00	3,026.05
7.00	1.16	35.97	2.00	3,027.64
8.00	1.18	36.48	2.00	3,033.45
9.00	1.18	36.56	2.00	3,033.21
10.00	1.18	36.33	2.00	3,030.50
11.00	1.17	36.21	2.00	3,029.58
12.00	1.18	36.23	2.00	3,029.91
13.00	1.18	36.23	2.00	3,029.50
14.00	1.17	36.15	2.00	3,028.43
15.00	1.17	36.07	2.00	3,027.69
16.00	1.17	36.04	2.00	3,027.43
17.00	1.17	36.01	2.00	3,027.19
18.00	1.17	35.98	2.00	3,026.76
19.00	1.17	35.94	2.00	3,026.33
20.00	1.17	35.91	2.00	3,026.02

TABLE D11

Model = REUSE5; Run = PC2_25.RSL; Change = NMEXTR = 2.25

Years	DVPROD	REUSE	NMEXTR	RPSTRY
0.00	0.97	22.60	2.25	2,000.00
1.00	0.99	27.13	2.25	2,407.89
2.00	1.03	33.09	2.25	2,873.53
3.00	1.19	38.48	2.25	3,103.19
4.00	1.24	39.77	2.25	3,101.97
5.00	1.20	37.94	2.25	3,057.62
6.00	1.15	36.20	2.25	3,032.37
7.00	1.14	36.08	2.25	3,032.33
8.00	1.16	36.58	2.25	3,037.93
9.00	1.16	36.69	2.25	3,037.86
10.00	1.16	36.46	2.25	3,035.29
11.00	1.15	36.34	2.25	3,034.48
12.00	1.16	36.36	2.25	3,034.85
13.00	1.16	36.36	2.25	3,034.49
14.00	1.16	36.28	2.25	3,033.41
15.00	1.15	36.20	2.25	3,032.64
16.00	1.15	36.17	2.25	3,032.38
17.00	1.15	36.15	2.25	3,032.15
18.00	1.15	36.11	2.25	3,031.74
19.00	1.15	36.07	2.25	3,031.32
20.00	1.15	36.04	2.25	3,031.02

TABLE D12

Model = REUSE5; Run = PC2_50.RSL; Change = NMEXTR = 2.50

Years	DVPROD	REUSE	NMEXTR	RPSTRY
0.00	0.97	22.60	2.50	2,000.00
1.00	0.97	26.35	2.50	2,342.07
2.00	0.98	31.84	2.50	2,773.31
3.00	1.14	37.69	2.50	3,075.66
4.00	1.22	39.83	2.50	3,106.10
5.00	1.18	38.31	2.50	3,064.72
6.00	1.13	36.45	2.50	3,039.28
7.00	1.12	36.21	2.50	3,037.68
8.00	1.14	36.69	2.50	3,043.09
9.00	1.15	36.82	2.50	3,043.23
10.00	1.14	36.60	2.50	3,040.60
11.00	1.14	36.46	2.50	3,039.80
12.00	1.14	36.49	2.50	3,040.26
13.00	1.14	36.50	2.50	3,040.00
14.00	1.14	36.42	2.50	3,038.94
15.00	1.14	36.34	2.50	3,038.11
16.00	1.14	36.31	2.50	3,037.82
17.00	1.14	36.29	2.50	3,037.61
18.00	1.14	36.25	2.50	3,037.23
19.00	1.13	36.22	2.50	3,036.81
20.00	1.13	36.19	2.50	3,036.52

TABLE D13

Model = REUSE5; Run = PC2_75.RSL; Change = NMEXTR = 2.75

Years	DVPROD	REUSE	NMEXTR	RPSTRY
0.00	0.96	22.60	2.75	2,000.00
1.00	0.96	25.60	2.75	2,277.79
2.00	0.96	30.27	2.75	2,636.99
3.00	1.08	35.75	2.75	2,976.56
4.00	1.18	39.00	2.75	3,082.12
5.00	1.17	38.46	2.75	3,061.57
6.00	1.12	36.67	2.75	3,032.01
7.00	1.10	36.10	2.75	3,023.79
8.00	1.12	36.44	2.75	3,027.04
9.00	1.12	36.62	2.75	3,027.70
10.00	1.12	36.44	2.75	3,025.33
11.00	1.12	36.29	2.75	3,024.25
12.00	1.12	36.29	2.75	3,024.76
13.00	1.12	36.31	2.75	3,024.76
14.00	1.12	36.25	2.75	3,023.81
15.00	1.12	36.17	2.75	3,022.88
16.00	1.11	36.13	2.75	3,022.46
17.00	1.11	36.11	2.75	3,022.22
18.00	1.11	36.08	2.75	3,021.86
19.00	1.11	36.04	2.75	3,021.45
20.00	1.11	36.01	2.75	3,021.13

TABLE D14

Model = REUSE5; Run = PC3.RSL; Change = NMEXTR = 3.0

Years	DVPROD	REUSE	NMEXTR	RPSTRY
0.00	0.96	22.60	3.00	2,000.00
1.00	0.95	24.81	3.00	2,206.55
2.00	0.95	28.45	3.00	2,484.78
3.00	1.00	32.74	3.00	2,776.79
4.00	1.10	36.64	3.00	2,983.46
5.00	1.14	37.78	3.00	3,030.28
6.00	1.11	36.91	3.00	3,021.79
7.00	1.09	36.17	3.00	3,011.23
8.00	1.10	36.20	3.00	3,009.87
9.00	1.10	36.34	3.00	3,009.61
10.00	1.10	36.23	3.00	3,007.36
11.00	1.10	36.07	3.00	3,005.78
12.00	1.10	36.03	3.00	3,005.87
13.00	1.10	36.05	3.00	3,006.15
14.00	1.10	36.02	3.00	3,005.66
15.00	1.10	35.95	3.00	3,004.80
16.00	1.09	35.90	3.00	3,004.22
17.00	1.09	35.87	3.00	3,003.88
18.00	1.09	35.85	3.00	3,003.53
19.00	1.09	35.82	3.00	3,003.13
20.00	1.09	35.79	3.00	3,002.79

TABLE D15

Model = REUSE5; Run = PC3_25.RSL; Change = NMEXTR = 3.25

Years	DVPROD	REUSE	NMEXTR	RPSTRY
0.00	0.96	22.60	3.25	2,000.00
1.00	0.95	24.07	3.25	2,139.38
2.00	0.95	26.72	3.25	2,338.57
3.00	0.98	29.71	3.25	2,551.38
4.00	1.02	32.70	3.25	2,740.86
5.00	1.05	34.88	3.25	2,895.57
6.00	1.08	36.00	3.25	2,967.75
7.00	1.08	36.22	3.25	2,988.07
8.00	1.08	36.16	3.25	2,990.89
9.00	1.08	36.08	3.25	2,988.95
10.00	1.08	35.95	3.25	2,986.16
11.00	1.08	35.81	3.25	2,984.07
12.00	1.08	35.72	3.25	2,983.30
13.00	1.08	35.70	3.25	2,983.31
14.00	1.08	35.68	3.25	2,983.15
15.00	1.08	35.65	3.25	2,982.65
16.00	1.08	35.61	3.25	2,982.10
17.00	1.07	35.57	3.25	2,981.67
18.00	1.07	35.55	3.25	2,981.29
19.00	1.07	35.52	3.25	2,980.93
20.00	1.07	35.49	3.25	2,980.60

TABLE D16

Model = REUSE5; Run = PC3_50.RSL; Change = NMEXTR = 3.50

Years	DVPROD	REUSE	NMEXTR	RPSTRY
0.00	0.95	22.60	3.50	2,000.00
1.00	0.95	23.44	3.50	2,081.94
2.00	0.94	25.21	3.50	2,209.10
3.00	0.97	27.20	3.50	2,347.46
4.00	0.99	28.94	3.50	2,468.56
5.00	1.00	30.13	3.50	2,567.90
6.00	1.00	31.07	3.50	2,653.49
7.00	1.01	32.12	3.50	2,731.63
8.00	1.02	33.23	3.50	2,801.12
9.00	1.03	34.13	3.50	2,859.21
10.00	1.04	34.77	3.50	2,907.60
11.00	1.05	35.19	3.50	2,936.95
12.00	1.06	35.32	3.50	2,946.39
13.00	1.06	35.26	3.50	2,947.64
14.00	1.05	35.15	3.50	2,946.64
15.00	1.05	35.08	3.50	2,945.73
16.00	1.05	35.05	3.50	2,945.27
17.00	1.05	35.02	3.50	2,945.00
18.00	1.05	35.00	3.50	2,944.79
19.00	1.05	34.98	3.50	2,944.61
20.00	1.05	34.96	3.50	2,944.43

TABLE D17

Model = REUSE5; Run = PC3_75.RSL; Change = NMEXTR = 3.75

Years	DVPROD	REUSE	NMEXTR	RPSTRY
0.00	0.95	22.60	3.75	2,000.00
1.00	0.95	22.91	3.75	2,034.26
2.00	0.94	23.91	3.75	2,098.20
3.00	0.97	25.02	3.75	2,167.92
4.00	0.98	25.87	3.75	2,223.12
5.00	0.98	26.27	3.75	2,263.41
6.00	0.98	26.52	3.75	2,296.86
7.00	0.98	26.87	3.75	2,326.57
8.00	0.99	27.23	3.75	2,349.97
9.00	0.99	27.44	3.75	2,365.17
10.00	0.99	27.51	3.75	2,373.56
11.00	0.99	27.53	3.75	2,377.25
12.00	0.99	27.52	3.75	2,377.14
13.00	0.99	27.47	3.75	2,373.54
14.00	0.99	27.37	3.75	2,367.04
15.00	0.99	27.25	3.75	2,358.38
16.00	0.99	27.11	3.75	2,348.05
17.00	0.99	26.96	3.75	2,336.35
18.00	0.99	26.79	3.75	2,323.51
19.00	0.99	26.61	3.75	2,309.80
20.00	0.99	26.42	3.75	2,295.41

TABLE D18

Model = REUSE5; Run = PC4.RSL; Change = NMEXTR = 4.0

Years	DVPROD	REUSE	NMEXTR	RPSTRY
0.00	0.95	22.60	4.00	2,000.00
1.00	0.95	22.48	4.00	1,994.61
2.00	0.95	22.83	4.00	2,004.90
3.00	0.96	23.17	4.00	2,013.67
4.00	0.98	23.23	4.00	2,008.69
5.00	0.97	22.91	4.00	1,993.16
6.00	0.96	22.55	4.00	1,974.99
7.00	0.96	22.31	4.00	1,955.50
8.00	0.97	22.07	4.00	1,931.91
9.00	0.97	21.72	4.00	1,903.34
10.00	0.96	21.30	4.00	1,871.33
11.00	0.96	20.87	4.00	1,837.23
12.00	0.96	20.43	4.00	1,801.38
13.00	0.96	19.98	4.00	1,763.99
14.00	0.96	19.51	4.00	1,725.50
15.00	0.96	19.03	4.00	1,686.37
16.00	0.96	18.55	4.00	1,646.90
17.00	0.95	18.08	4.00	1,607.25
18.00	0.95	17.60	4.00	1,567.59
19.00	0.95	17.13	4.00	1,528.08
20.00	0.95	16.66	4.00	1,488.85

TABLE D19

Model = REUSE5; Run = PC4_25.RSL; Change = NMEXTR = 4.25

Years	DVPROD	REUSE	NMEXTR	RPSTRY
0.00	0.95	22.60	4.25	2,000.00
1.00	0.95	22.11	4.25	1,961.41
2.00	0.95	21.94	4.25	1,927.04
3.00	0.97	21.64	4.25	1,884.96
4.00	0.97	21.04	4.25	1,828.25
5.00	0.96	20.11	4.25	1,763.47
6.00	0.95	19.22	4.25	1,698.64
7.00	0.95	18.46	4.25	1,634.65
8.00	0.95	17.71	4.25	1,569.45
9.00	0.95	16.91	4.25	1,503.02
10.00	0.95	16.11	4.25	1,436.68
11.00	0.94	15.35	4.25	1,373.41
12.00	0.94	14.73	4.25	1,321.65
13.00	0.93	14.25	4.25	1,281.49
14.00	0.93	13.89	4.25	1,250.52
15.00	0.93	13.60	4.25	1,226.13
16.00	0.92	13.36	4.25	1,206.10
17.00	0.92	13.16	4.25	1,188.99
18.00	0.92	12.99	4.25	1,173.97
19.00	0.92	12.83	4.25	1,160.59
20.00	0.92	12.69	4.25	1,148.59

TABLE D20

Model = REUSE5; Run = PC4_50.RSL; Change = NMEXTR = 4.50

Years	DVPROD	REUSE	NMEXTR	RPSTRY
0.00	0.94	22.60	4.50	2,000.00
1.00	0.95	21.80	4.50	1,933.40
2.00	0.95	21.19	4.50	1,861.64
3.00	0.97	20.37	4.50	1,777.57
4.00	0.97	19.24	4.50	1,679.11
5.00	0.96	17.85	4.50	1,574.71
6.00	0.95	16.53	4.50	1,471.84
7.00	0.94	15.43	4.50	1,378.06
8.00	0.94	14.56	4.50	1,303.82
9.00	0.93	13.88	4.50	1,247.32
10.00	0.92	13.37	4.50	1,204.76
11.00	0.92	12.99	4.50	1,172.30
12.00	0.92	12.69	4.50	1,146.43
13.00	0.92	12.43	4.50	1,124.59
14.00	0.92	12.21	4.50	1,105.30
15.00	0.91	12.00	4.50	1,087.74
16.00	0.91	11.82	4.50	1,071.58
17.00	0.91	11.65	4.50	1,056.64
18.00	0.91	11.49	4.50	1,042.84
19.00	0.91	11.34	4.50	1,030.10
20.00	0.91	11.20	4.50	1,018.30

TABLE D21

Model = REUSE5; Run = PC5.RSL; Change = NMEXTR = 5.0

Years	DVPROD	REUSE	NMEXTR	RPSTRY
0.00	0.94	22.60	5.00	2,000.00
1.00	0.96	21.32	5.00	1,888.96
2.00	0.96	20.05	5.00	1,762.12
3.00	0.98	18.47	5.00	1,616.35
4.00	0.98	16.61	5.00	1,458.87
5.00	0.95	14.86	5.00	1,324.73
6.00	0.93	13.64	5.00	1,227.38
7.00	0.92	12.83	5.00	1,157.24
8.00	0.92	12.22	5.00	1,103.41
9.00	0.91	11.71	5.00	1,059.36
10.00	0.91	11.27	5.00	1,022.59
11.00	0.91	10.91	5.00	991.27
12.00	0.91	10.60	5.00	963.62
13.00	0.90	10.31	5.00	938.61
14.00	0.90	10.05	5.00	915.85
15.00	0.90	9.81	5.00	895.03
16.00	0.90	9.57	5.00	875.90
17.00	0.90	9.33	5.00	858.60
18.00	0.90	9.12	5.00	843.00
19.00	0.90	8.93	5.00	828.37
20.00	0.90	8.75	5.00	814.12

APPENDIX E: AVERAGE EMPLOYMENT DOCUMENTATION

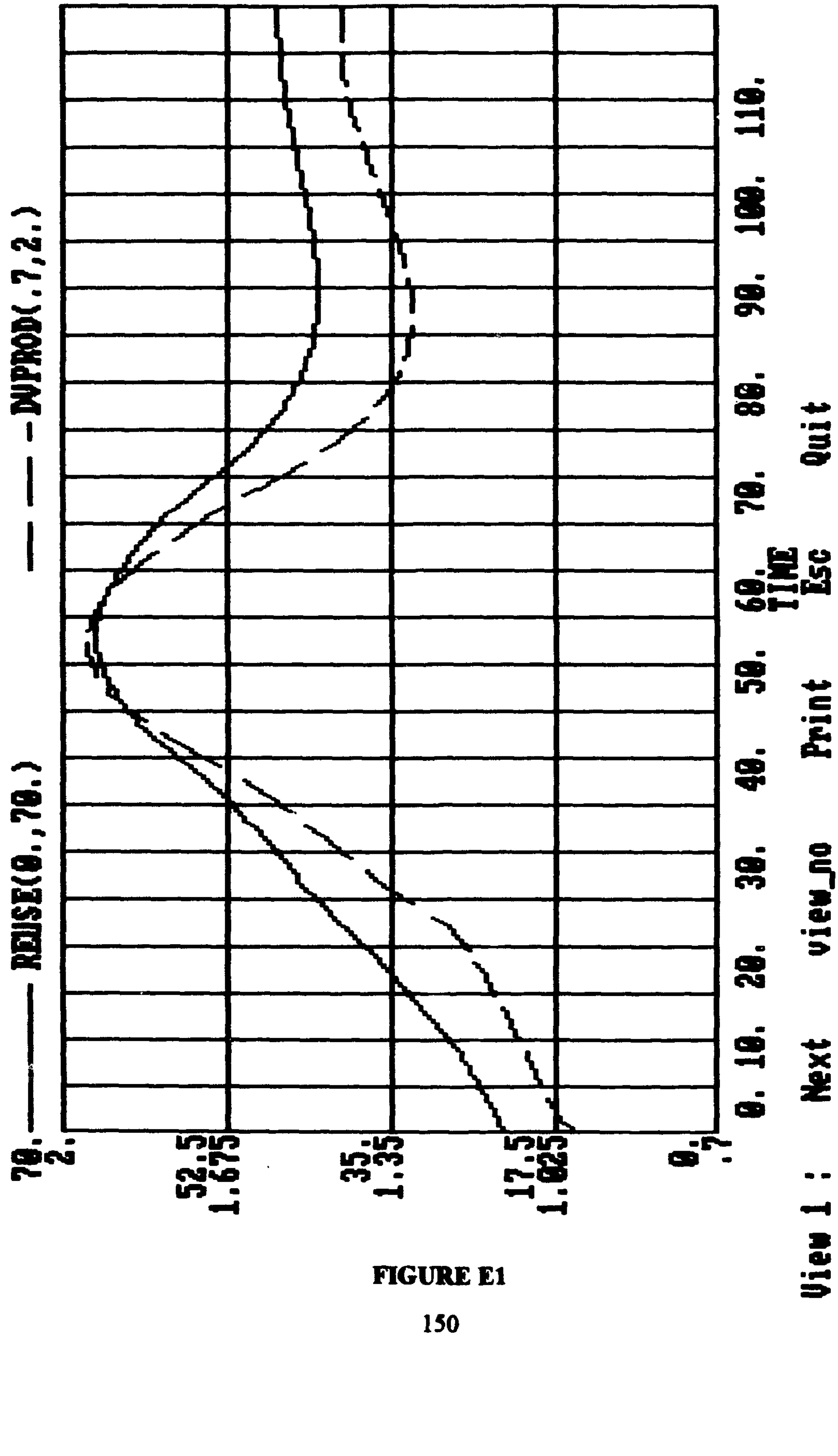


FIGURE E1

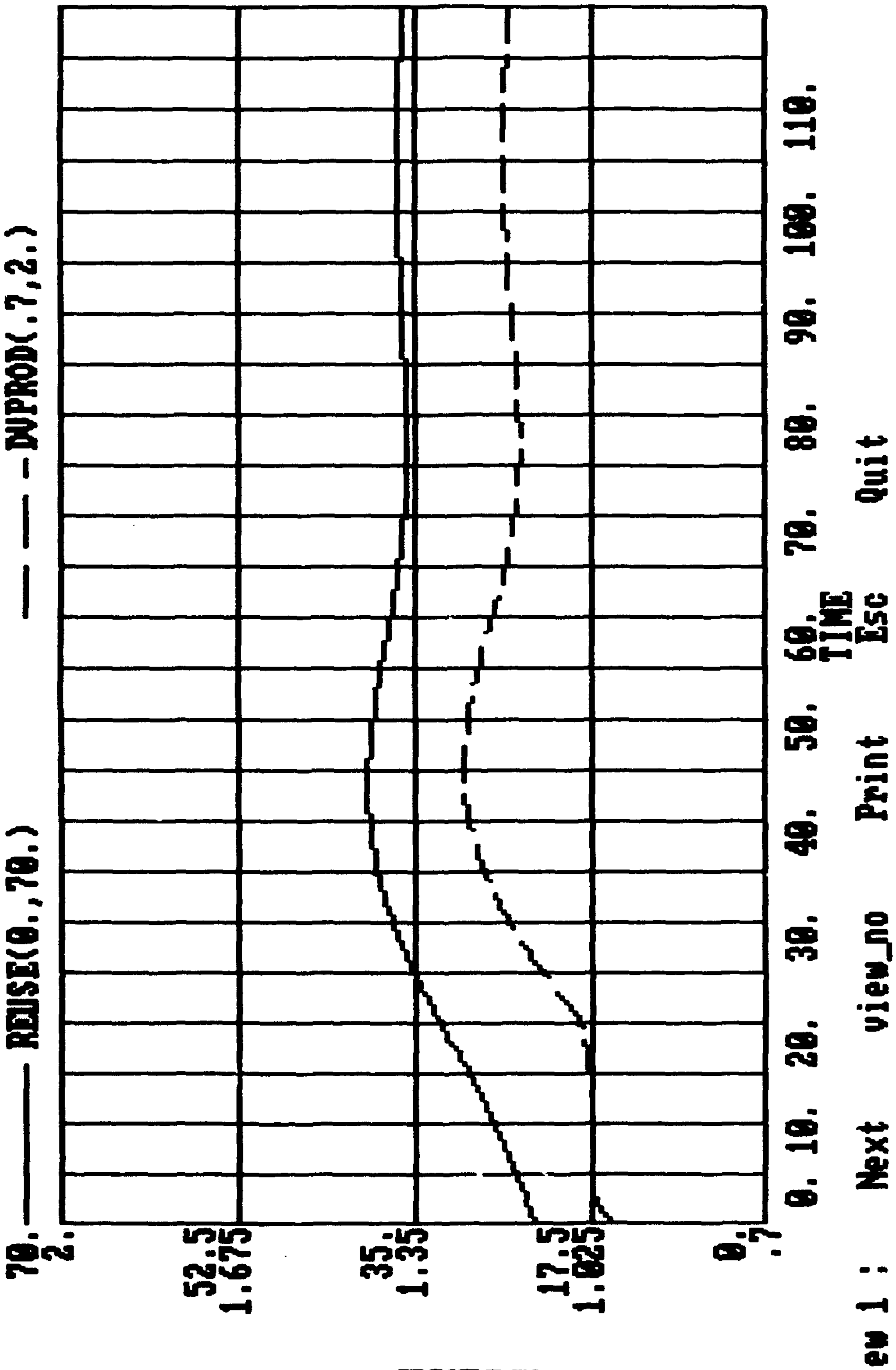

REUSE(0.,70.)
DUPROD(.7,2.)
70.
2.
52.5
1.675
35.
1.35
17.5
1.025
0.
.7
0. 10. 20. 30. 40. 50. 60. 70. 80. 90. 100. 110.
TIME
View 1 :
Next
view_no
Print
Esc
Quit

FIGURE E2

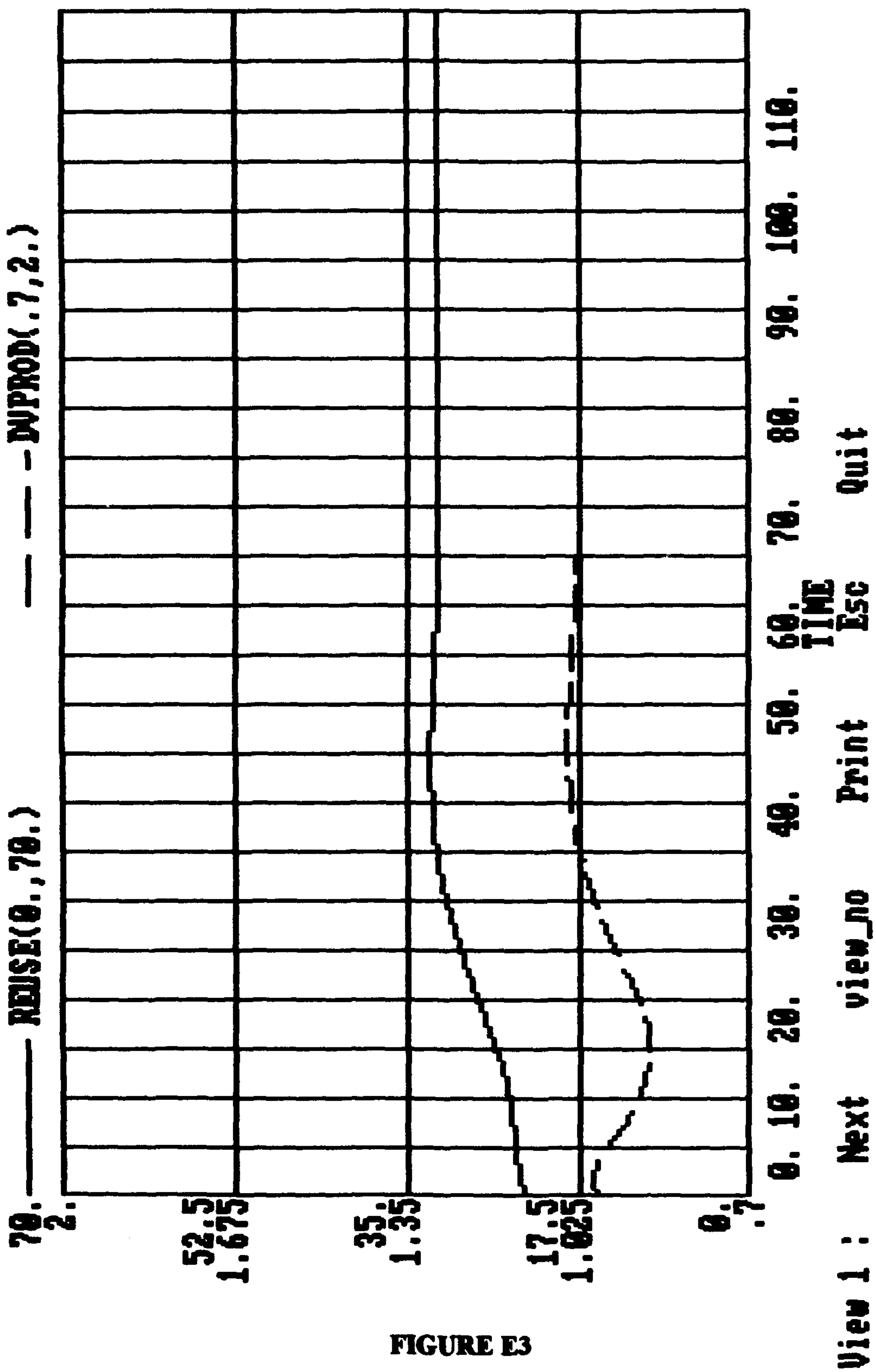

REUSE(0.,70.)
DUPROD(.7,2.)
70. 2.
52.5 1.675
35. 1.35
17.5 1.025
0. .7
0. 10. 20. 30. 40. 50. 60. 70. 80. 90. 100. 110.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE E3

TABLE E1

Model = REUSE5; Run = TURNLOW.RSL; Change = AVEMPT = 84

Years	DVPROD	REUSE	AVGC	RPSTRY
0.00	0.98	22.60	2.00	2,000.00
1.00	1.11	30.43	2.83	2,535.03
2.00	1.30	42.13	3.62	3,100.54
3.00	1.60	53.14	4.89	3,288.92
4.00	1.93	65.27	6.43	3,311.82
5.00	1.87	64.07	7.79	3,237.48
6.00	1.51	51.57	8.61	3,110.59
7.00	1.32	43.58	8.81	3,045.31
8.00	1.34	43.56	8.82	3,050.78
9.00	1.43	45.98	8.90	3,072.31
10.00	1.45	47.21	9.07	3,080.42

TABLE E2

Model = REUSE5; Run = TURNNOM.RSL; Change = AVEMPT = 42

Years	DVPROD	REUSE	AVGUSE	RPSTRY
0.00	0.98	22.60	2.00	2,000.00
1.00	1.02	28.08	2.76	2,479.13
2.00	1.10	34.62	3.38	2,954.76
3.00	1.23	38.98	4.25	3,118.52
4.00	1.26	39.46	5.07	3,093.69
5.00	1.21	37.46	5.64	3,046.61
6.00	1.16	35.96	6.02	3,024.58
7.00	1.16	36.01	6.32	3,027.06
8.00	1.18	36.55	6.59	3,033.33
9.00	1.19	36.65	6.82	3,033.43
10.00	1.18	36.45	7.00	3,031.05

TABLE E3

Model = REUSE5; Run = TURNHIGH.RSL; Change = AVEMPT = 21

Years	DVPROD	REUSE	AVGUSE	RPSTRY
0.00	0.98	22.60	2.00	2,000.00
1.00	0.90	25.02	2.65	2,391.52
2.00	0.95	29.24	3.17	2,780.00
3.00	1.03	32.19	3.81	2,968.30
4.00	1.04	32.55	4.39	2,990.93
5.00	1.03	32.03	4.84	2,972.34
6.00	1.02	31.76	5.18	2,962.06
7.00	1.02	31.72	5.46	2,961.98
8.00	1.02	31.78	5.70	2,965.12
9.00	1.02	31.83	5.89	2,967.07
10.00	1.02	31.83	6.06	2,967.19

APPENDIX F: RETIREMENT AGE DOCUMENTATION

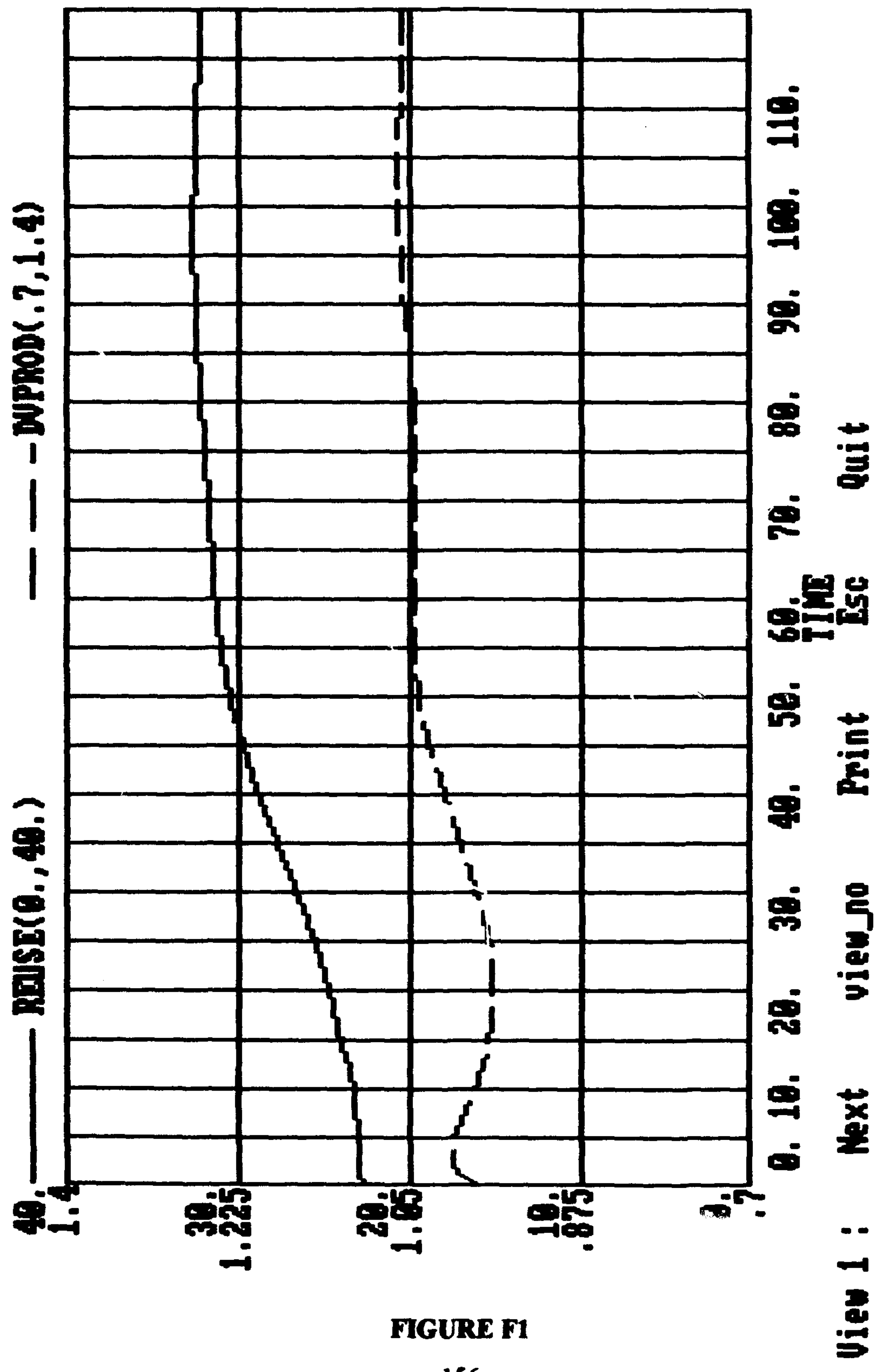


FIGURE F1

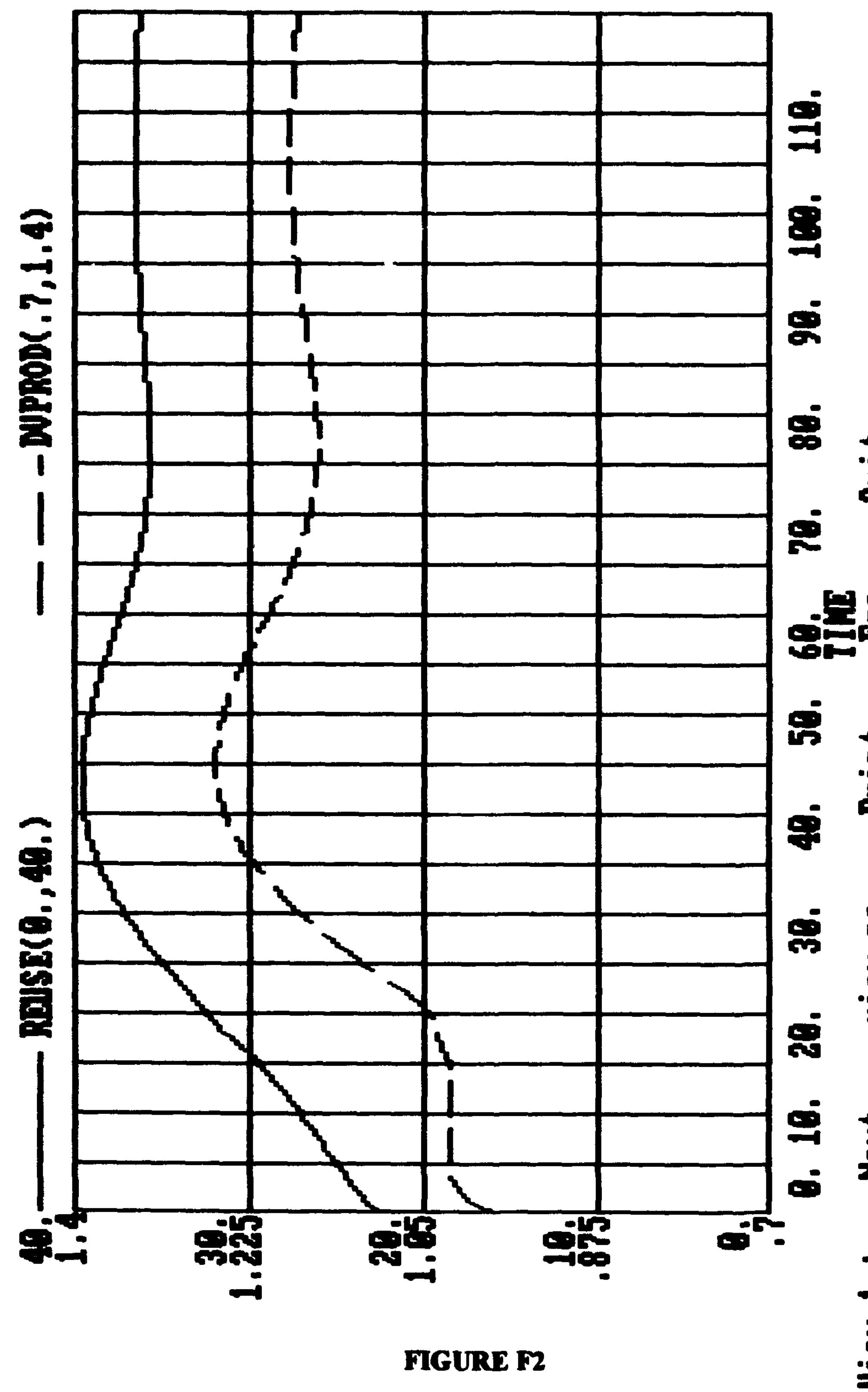

REUSE(0.,40.)
DUPROD(.7,1.4)
40. 1.4
30. 1.225
20. 1.05
10. .875
0. .7
0. 10. 20. 30. 40. 50. 60. 70. 80. 90. 100. 110.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE F2

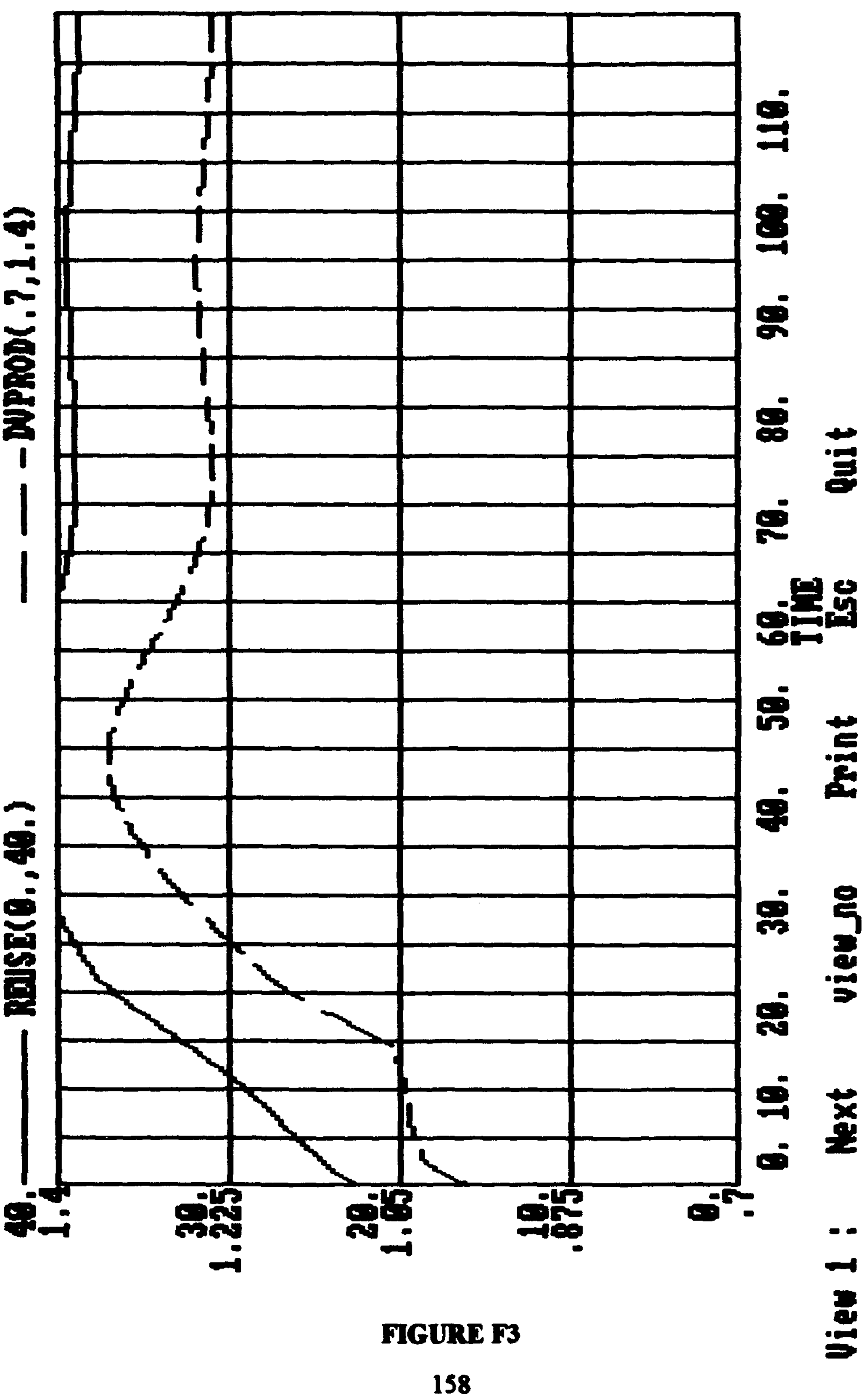

REUSE(0.,40.)
DUPROD(.7,1.4)
40. 1.4
30. 1.225
20. 1.05
10. .875
0. .7
0. 10. 20. 30. 40. 50. 60. 70. 80. 90. 100. 110.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE F3

TABLE F1

Model = REUSE5; Run = RETIRE30.RSL; Change = NMRCLF = 30

Years	DVPROD	REUSE	AVGUSE	RPSTRY
0.00	0.98	22.60	2.00	2,000.00
1.00	0.98	23.56	2.69	2,079.20
2.00	0.97	25.43	3.09	2,239.25
3.00	1.00	28.03	3.39	2,428.59
4.00	1.04	30.31	3.62	2,580.42
5.00	1.05	31.43	3.81	2,667.51
6.00	1.04	31.93	3.95	2,723.14
7.00	1.05	32.45	4.05	2,765.70
8.00	1.06	32.73	4.15	2,780.09
9.00	1.06	32.58	4.23	2,771.27
10.00	1.06	32.27	4.27	2,759.35

TABLE F2

Model = REUSE5; Run = RETIRE60.RSL; Change = NMRCLF = 60

Years	DVPROD	REUSE	AVGUSE	RPSTRY
0.00	0.98	22.60	2.00	2,000.00
1.00	1.02	28.08	2.76	2,479.13
2.00	1.10	34.62	3.38	2,954.76
3.00	1.23	38.98	4.25	3,118.52
4.00	1.26	39.46	5.07	3,093.69
5.00	1.21	37.46	5.64	3,046.61
6.00	1.16	35.96	6.02	3,024.58
7.00	1.16	36.01	6.32	3,027.06
8.00	1.18	36.55	6.59	3,033.33
9.00	1.19	36.65	6.82	3,033.43
10.00	1.18	36.45	7.00	3,031.05

TABLE F3

Model = REUSE5; Run = RETIR120.RSL; Change = NMRCLF = 120

Years	DVPROD	REUSE	AVGUSE	RPSTRY
0.00	0.98	22.60	2.00	2,000.00
1.00	1.05	30.63	2.79	2,710.51
2.00	1.21	38.78	3.70	3,219.08
3.00	1.32	42.16	4.86	3,345.43
4.00	1.34	42.29	5.91	3,353.42
5.00	1.28	40.17	6.77	3,329.16
6.00	1.24	39.06	7.50	3,311.59
7.00	1.25	39.29	8.17	3,306.82
8.00	1.26	39.56	8.80	3,298.14
9.00	1.25	39.25	9.40	3,272.26
10.00	1.24	38.77	9.99	3,229.91

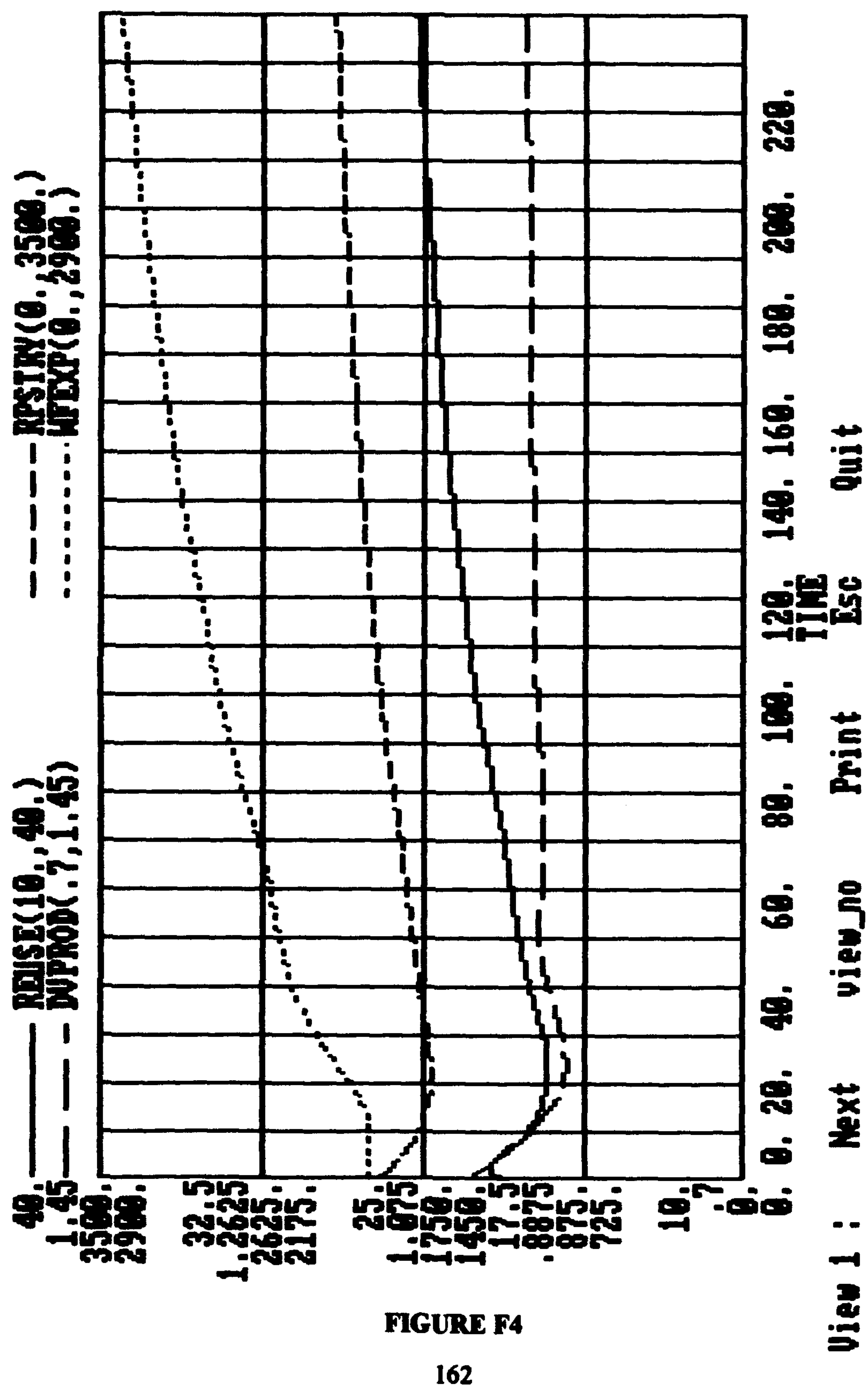

REUSE(10.,40.)
DVPROD(.7,1.45)
RPSTRY(0.,3500.)
WPEXP(0.,2900.)
TIME
View 1 : Next view_no Print Esc Quit

FIGURE F4

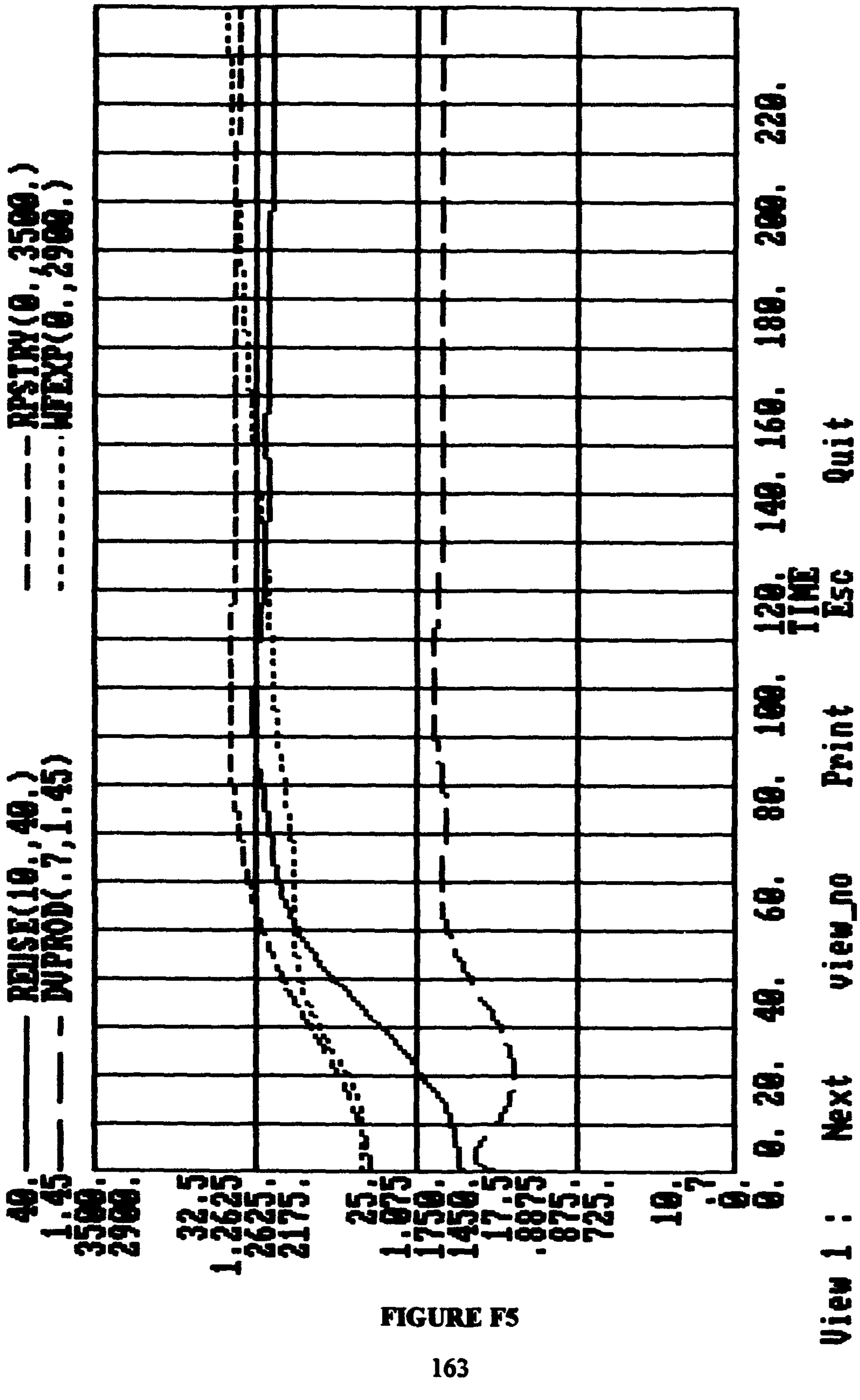

REUSE(10.,40.)
DUPROD(.7,1.45)
RPSTRY(0.,3500.)
WFEXP(0.,2900.)
0. 20. 40. 60. 80. 100. 120. 140. 160. 180. 200. 220.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE F5

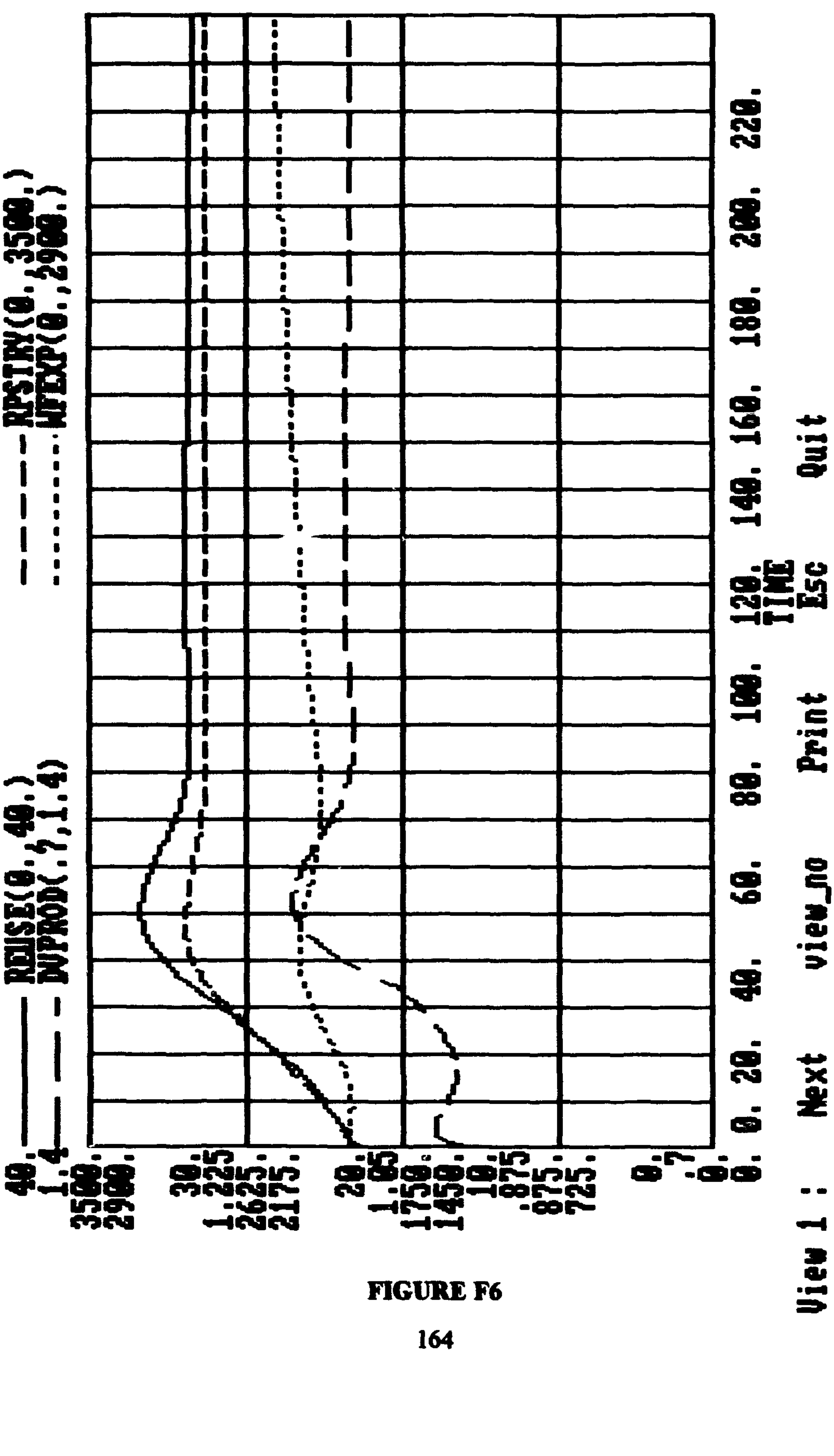
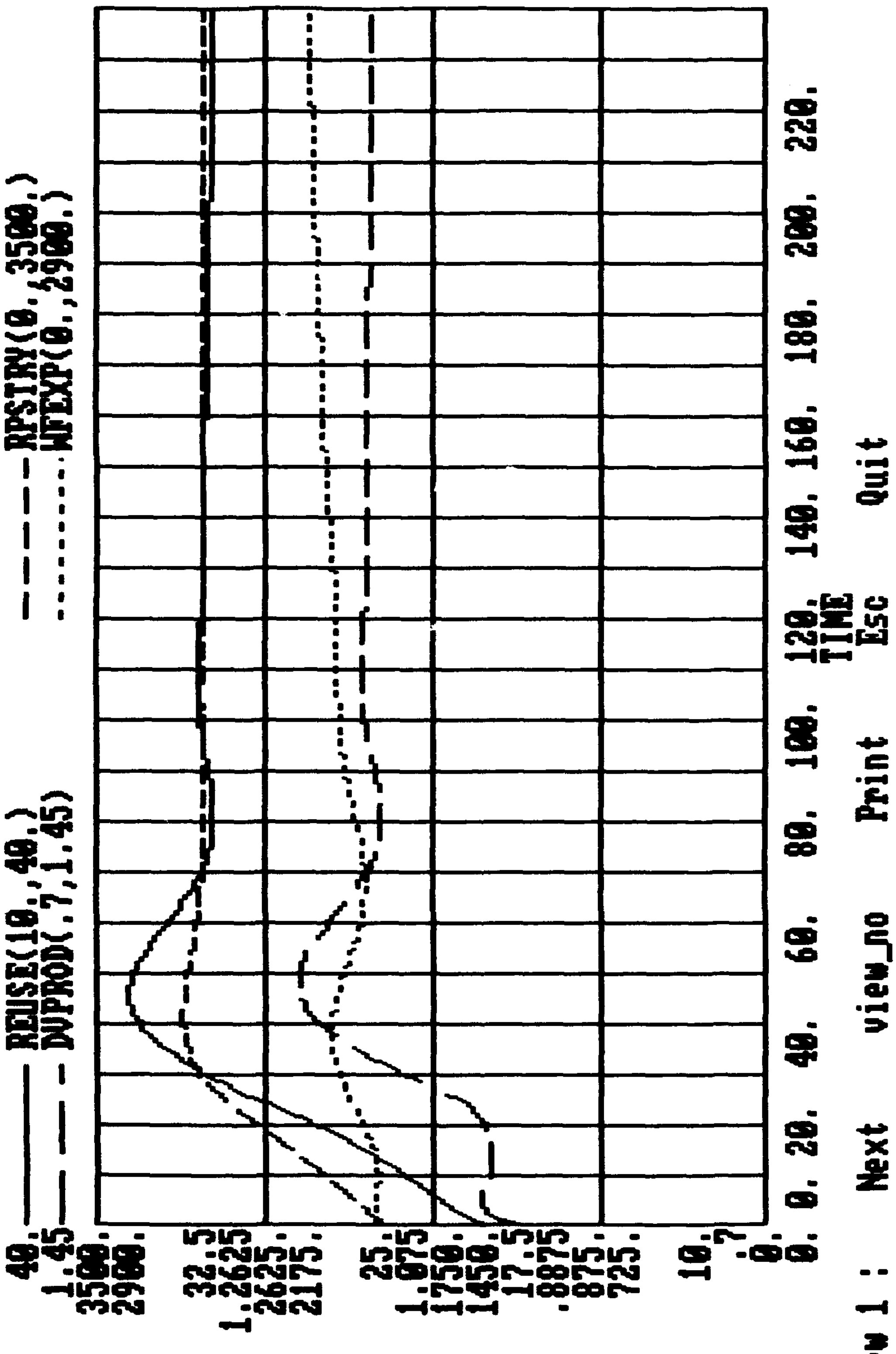

REUSE(0.,40.)
DVPROD(.7,1.4)
RPSTRY(0.,3500.)
WPEXP(0.,2900.)
TIME
View 1 : Next view_no Print Esc Quit

FIGURE F6



REUSE(10.,40.)
DVPROD(.7,1.45)
RPSTRY(0.,3500.)
WFEXP(0.,2900.)
TIME
View 1 : Next view_no Print Esc Quit

FIGURE F7

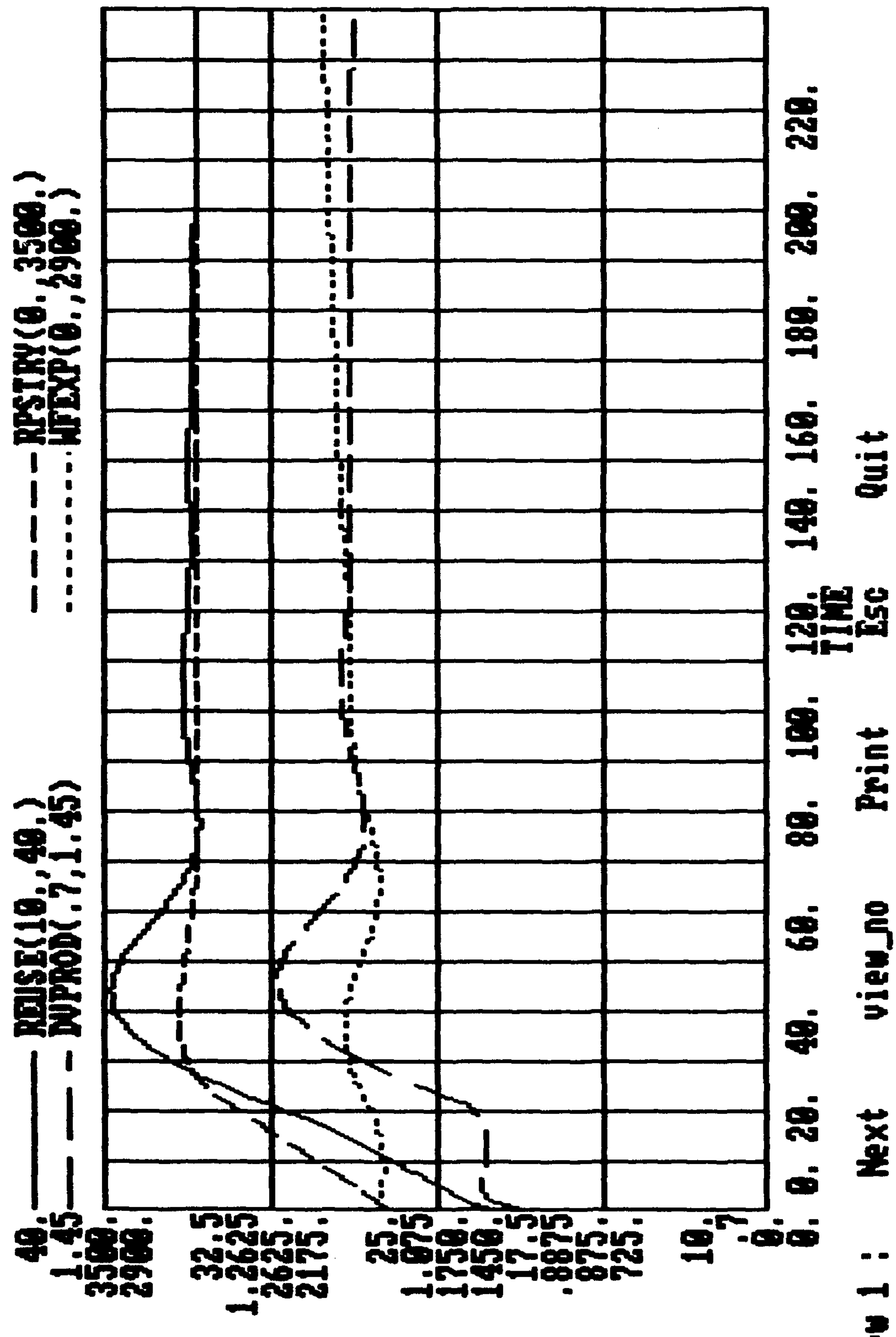

REUSE(10.,40.)
DUPROD(.7,1.45)
RPSTRY(0.,3500.)
WFEXP(0.,2900.)
TIME
View 1 : Next view_no Print Esc Quit

FIGURE F8

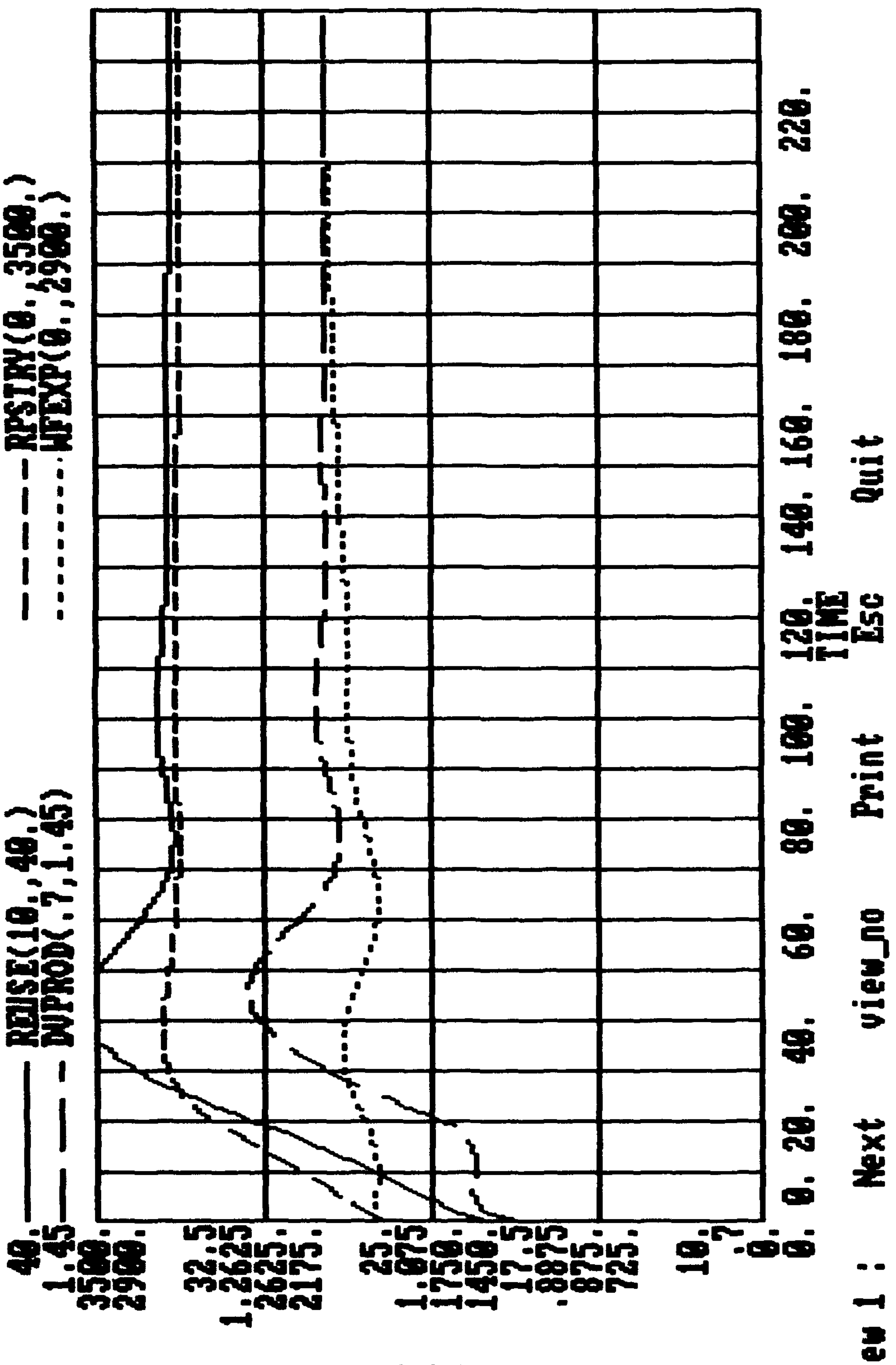
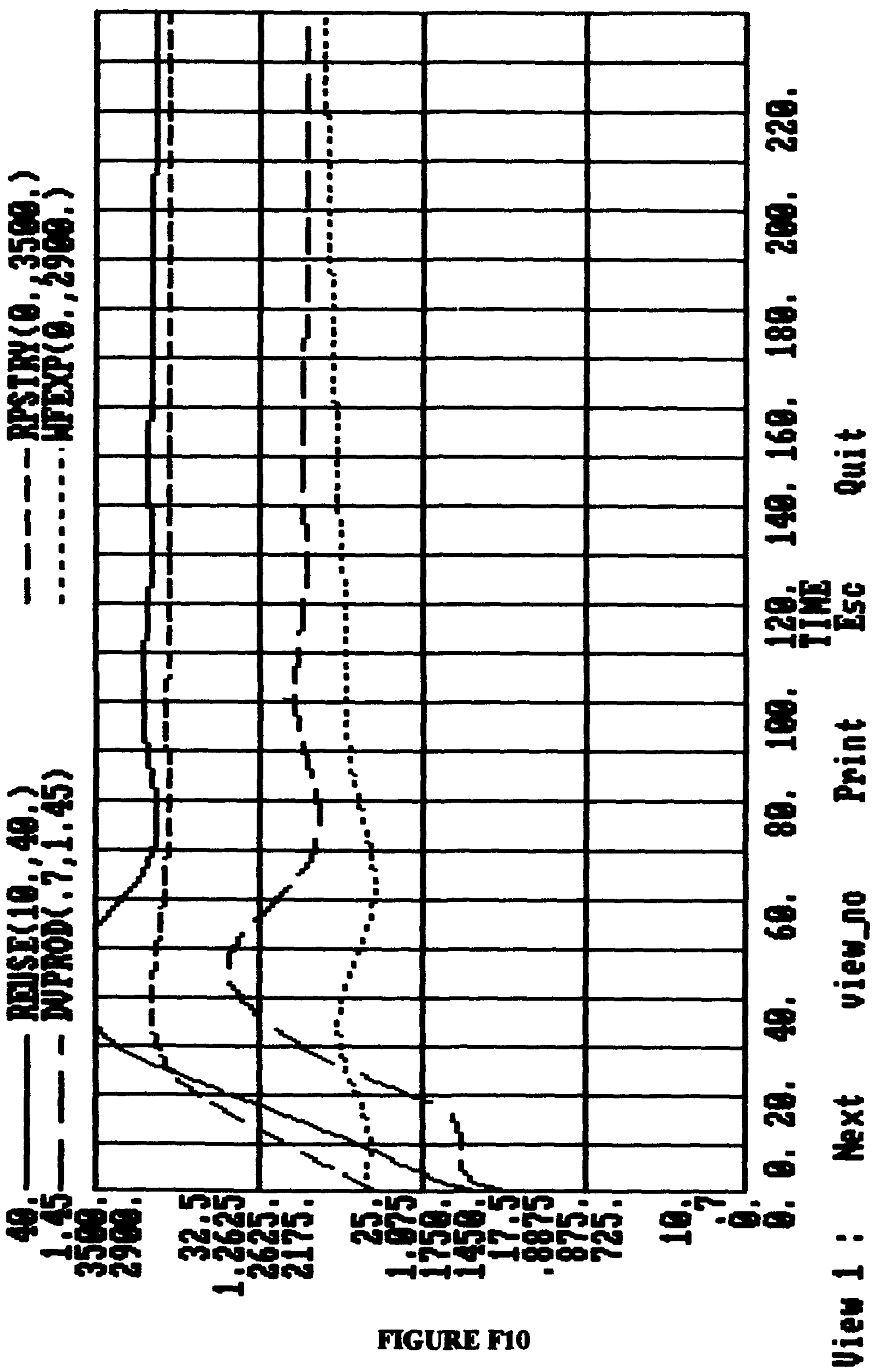

REUSE(10.,40.)
DVPROD(.7,1.45)
RPSTRY(0.,3500.)
WFEXP(0.,2900.)
0. 20. 40. 60. 80. 100. 120. 140. 160. 180. 200. 220.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE F9



REUSE(10.,40.)
DVPROD(.7,1.45)
RPSTRY(0.,3500.)
WFEXP(0.,2900.)
TIME
View 1 : Next view_no Print Esc Quit

FIGURE F10

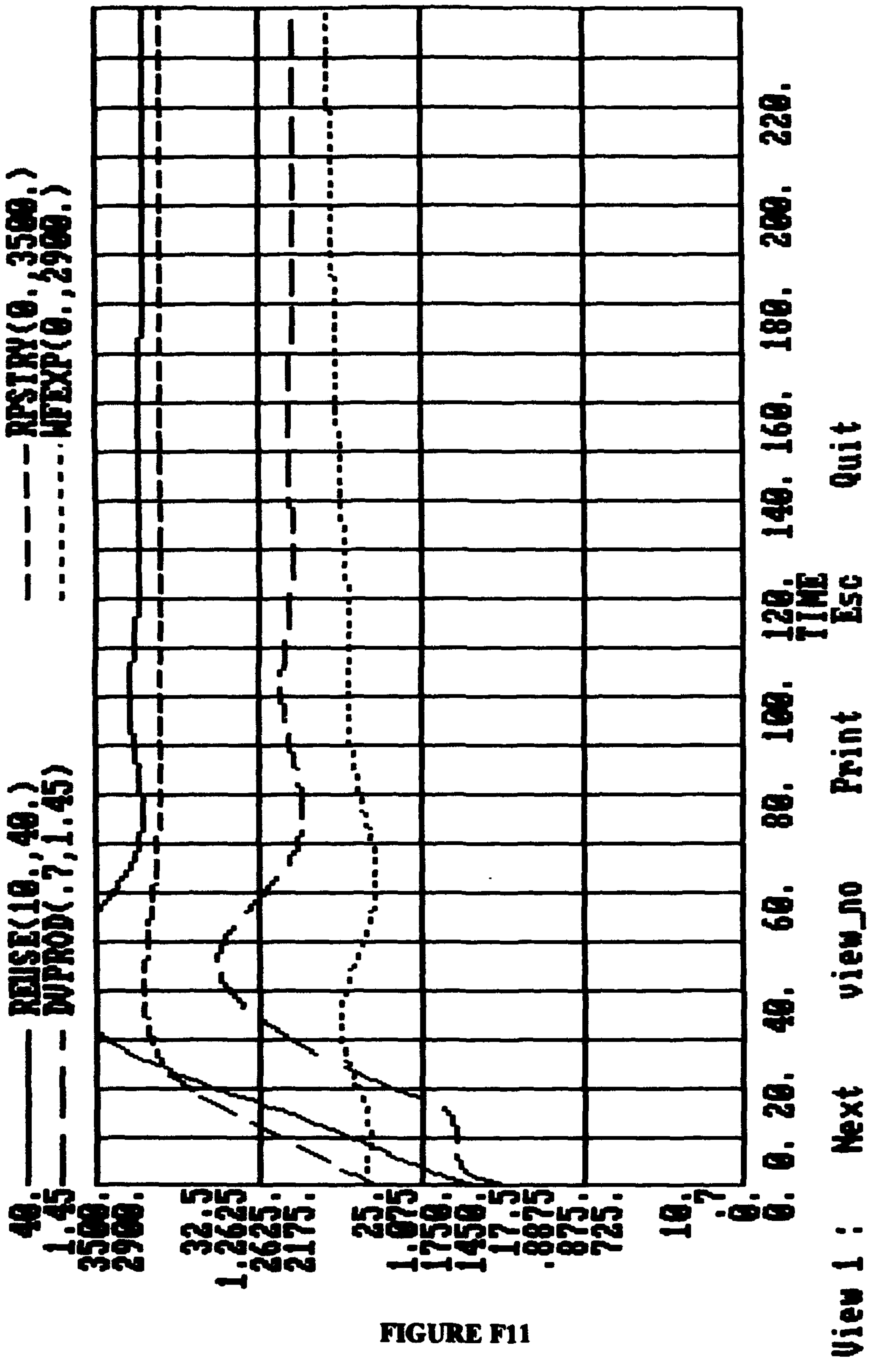

REUSE(10.,40.)
DUPROD(.7,1.45)
RPSTRY(0.,3500.)
WFEXP(0.,2900.)
TIME
View 1 :
Next
view_no
Print
Esc
Quit

FIGURE F11

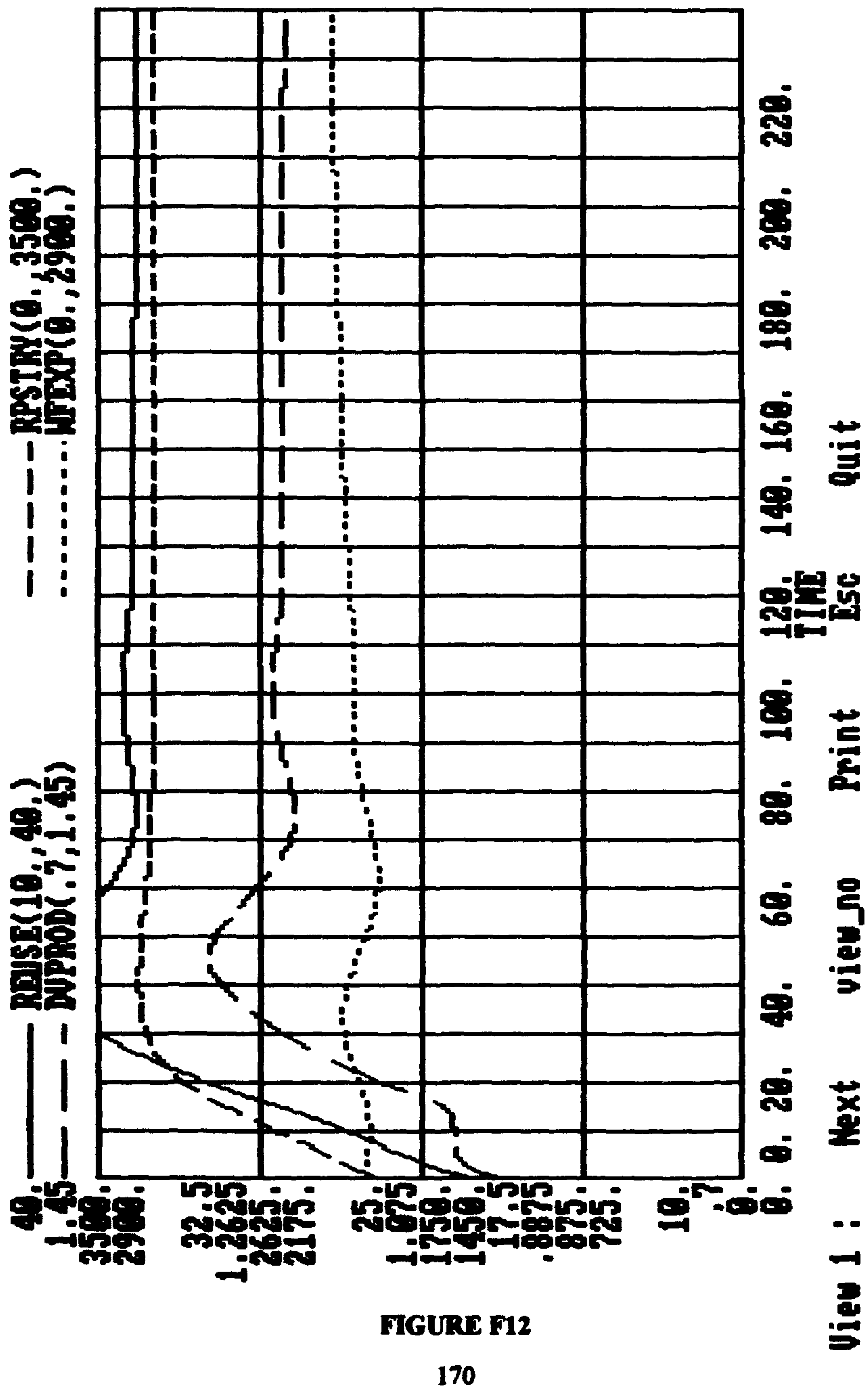

REUSE(10.,40.)
DVPROD(.7,1.45)
RPSTRY(0.,3500.)
WFEXP(0.,2900.)
0. 20. 40. 60. 80. 100. 120. 140. 160. 180. 200. 220.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE F12

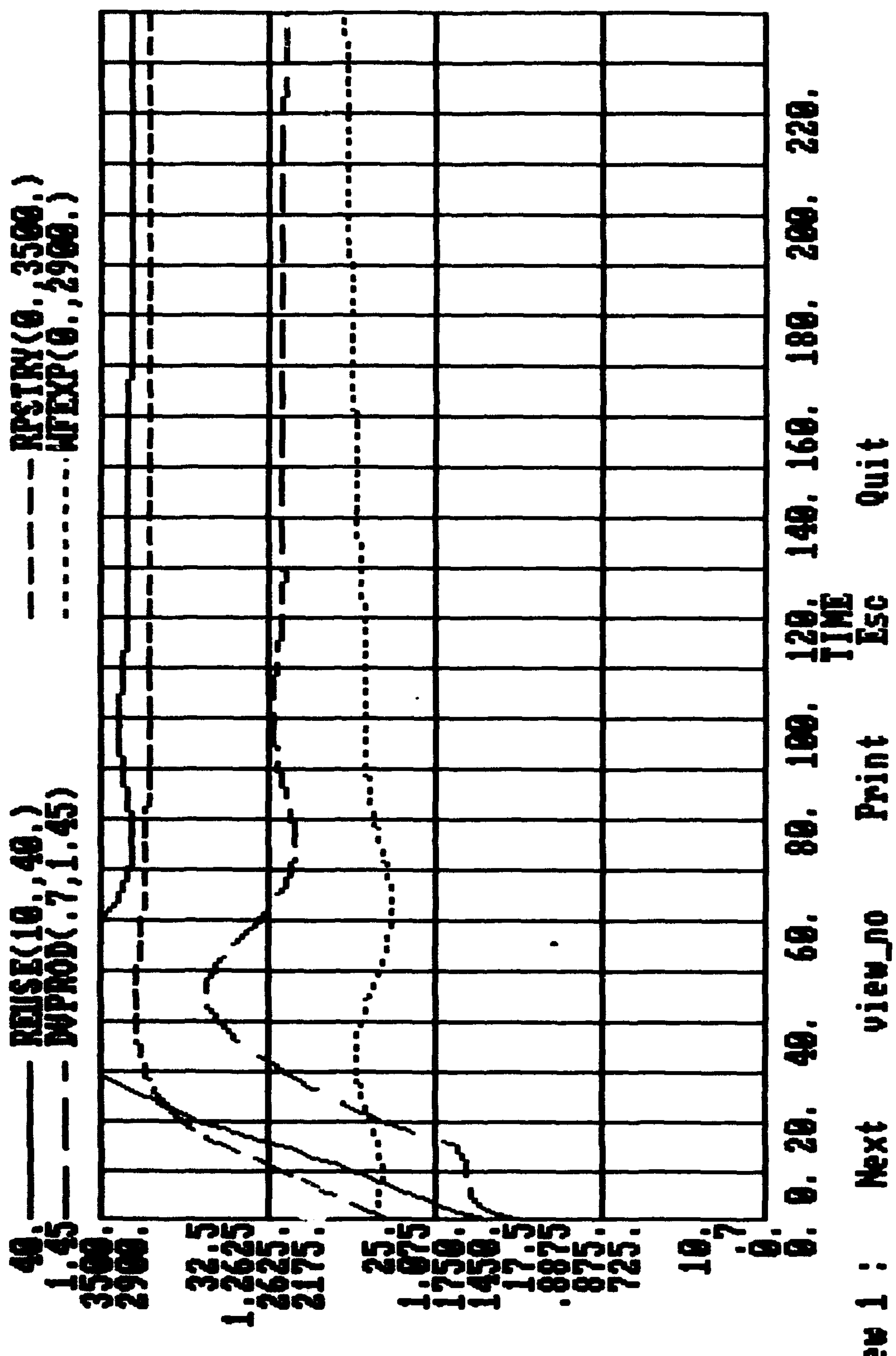

REUSE(10.,40.)
DVPROD(.7,1.45)
RPSTRY(0.,3500.)
WTEXP(0.,2900.)
TIME
View 1 : Next view_no Print Esc Quit

FIGURE F13

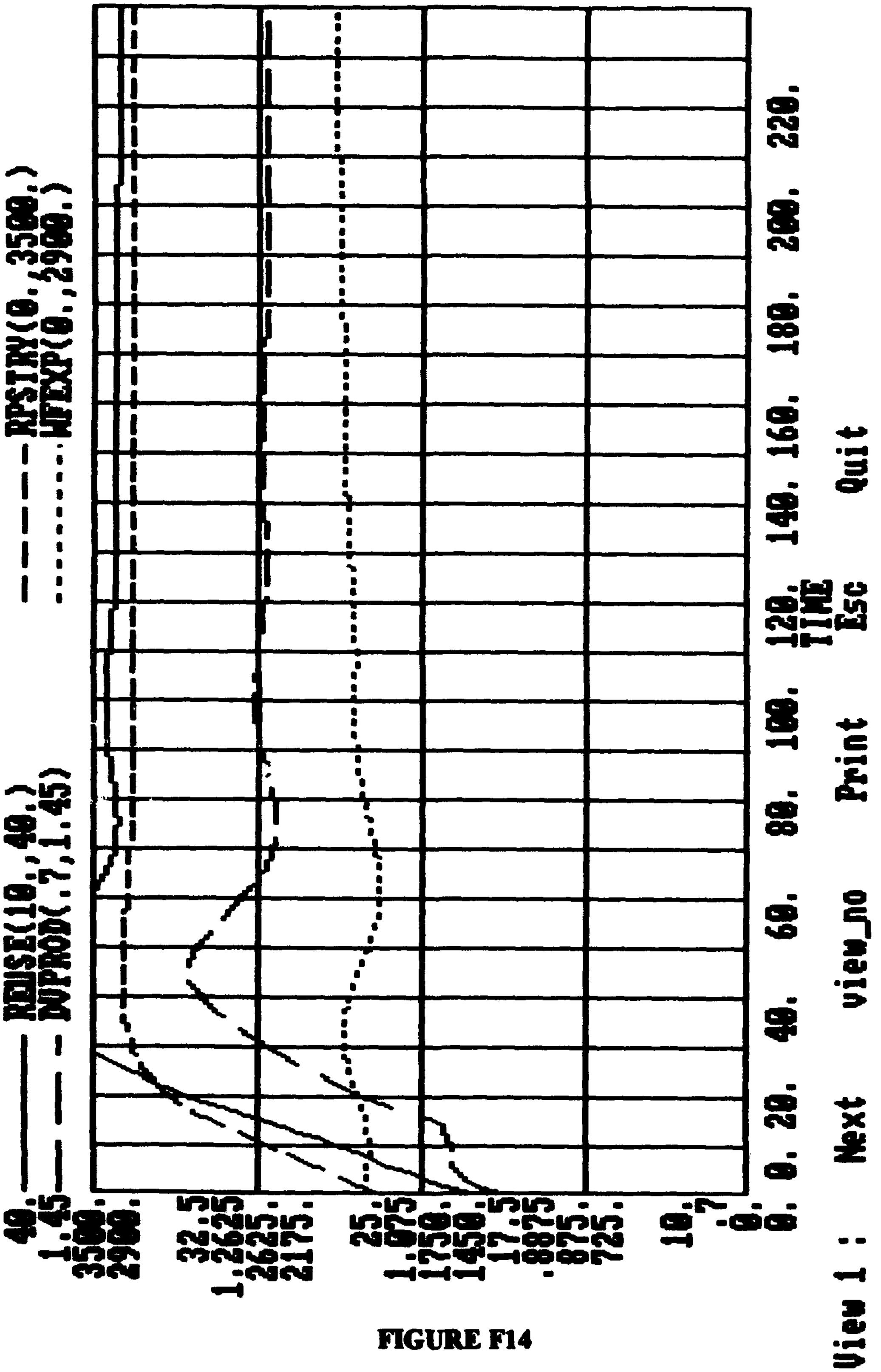

REUSE(10.,40.)
DVPROD(.7,1.45)
RPSTRY(0.,3500.)
WFEXP(0.,2900.)
TIME
View 1 : Next view_no Print Esc Quit

FIGURE F14

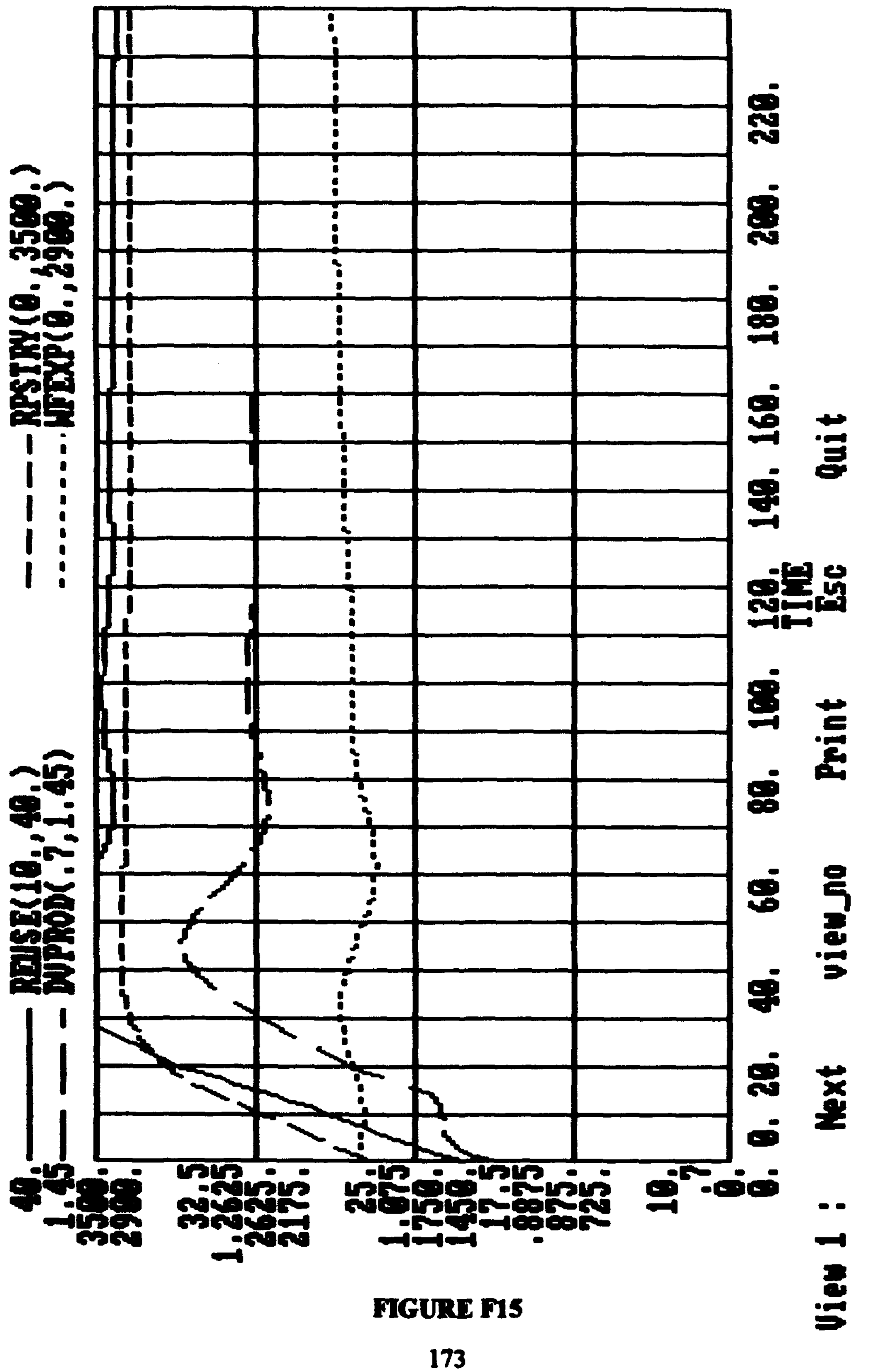

REUSE(10.,40.)
DUPROD(.7,1.45)
RPSTRY(0.,3500.)
WFEXP(0.,2900.)
TIME
View 1 :
Next
view_no
Print
Esc
Quit

FIGURE F15

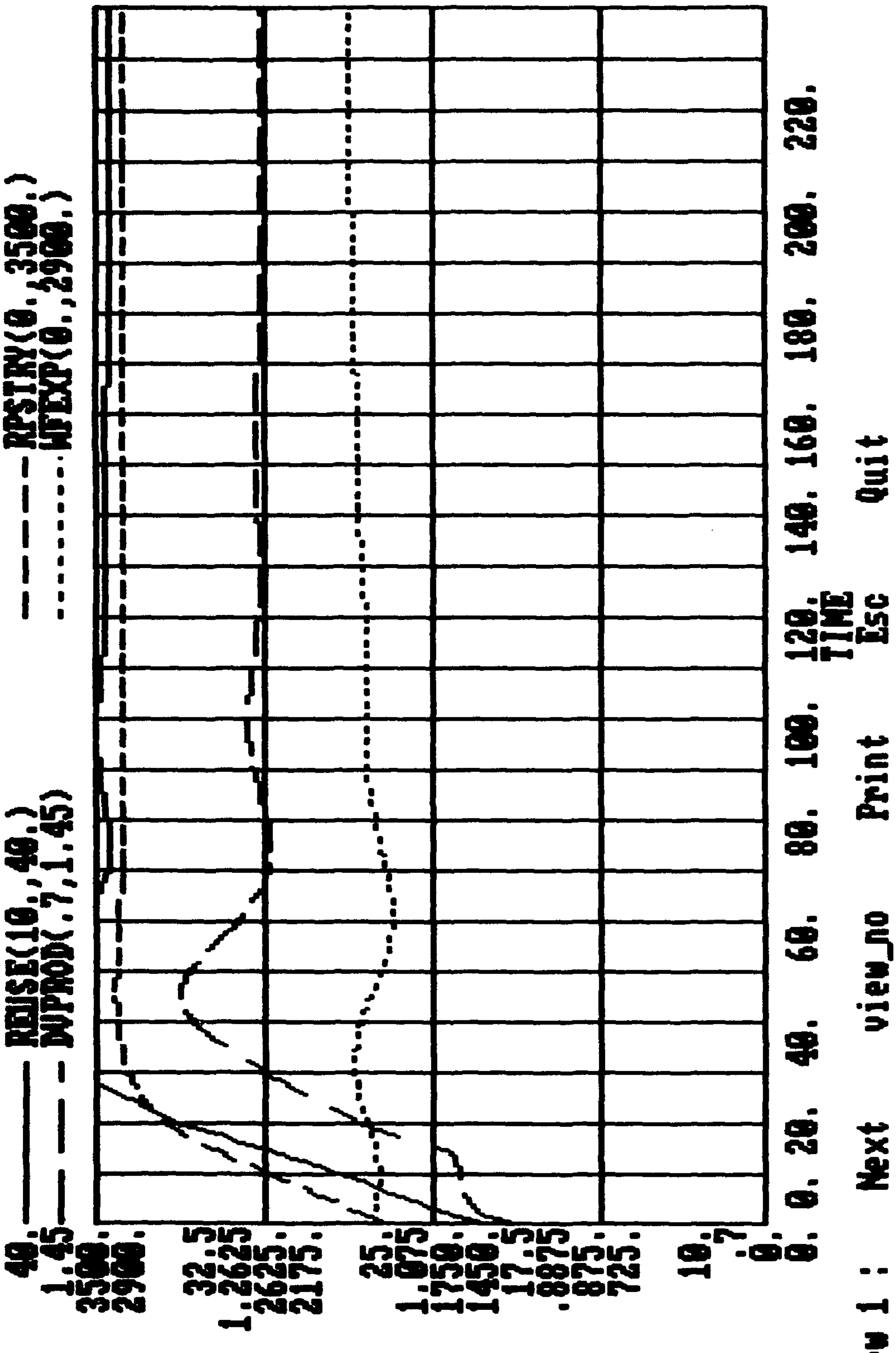
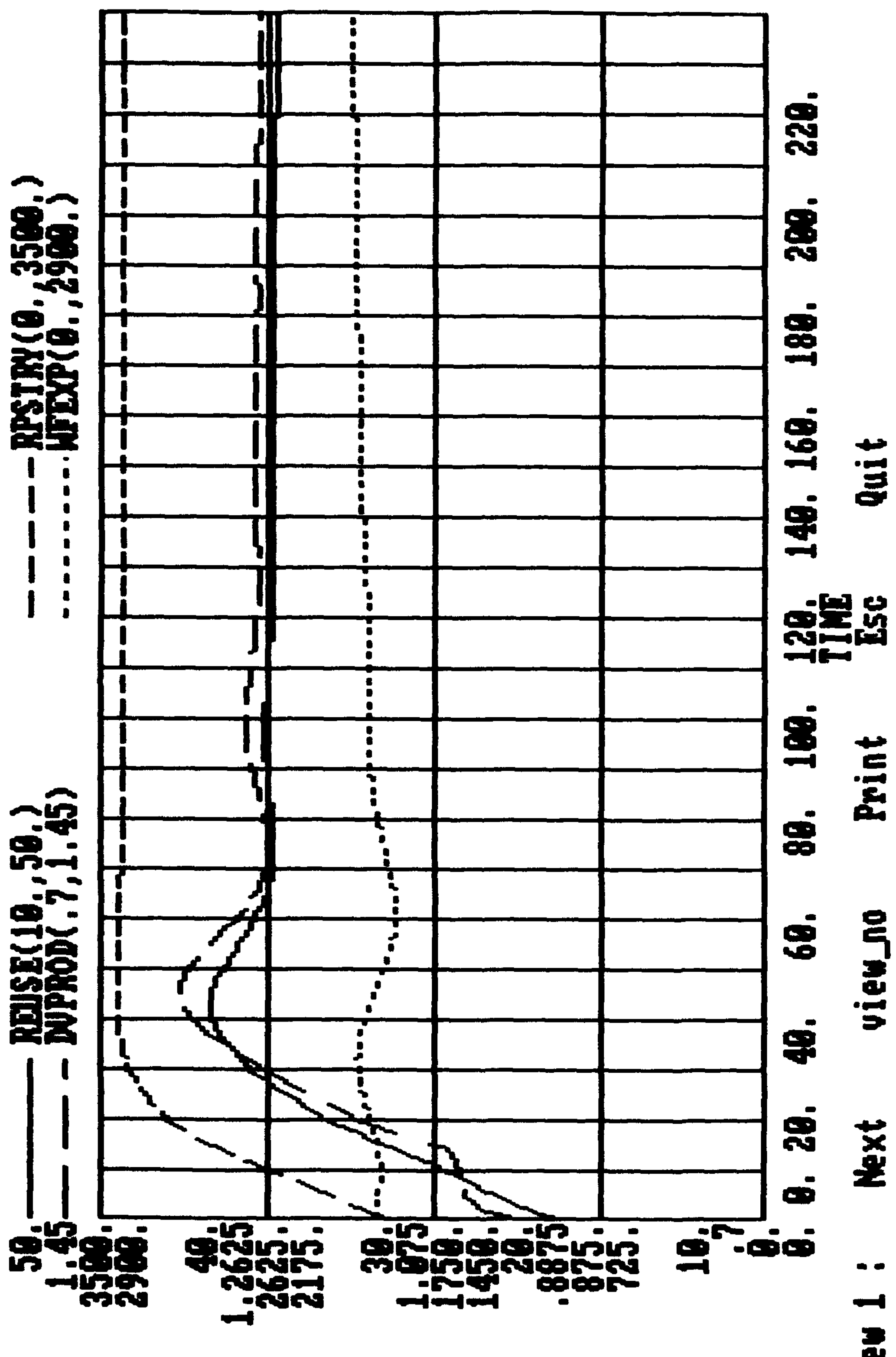

REUSE(10.,40.)
DUPROD(.7,1.45)
RPSTRY(0.,3500.)
WFEXP(0.,2900.)
TIME
View 1 : Next view_no Print Esc Quit

FIGURE F16



REUSE(10.,50.)
DUPROD(.7,1.45)
RPSTRY(0.,3500.)
WTEXP(0.,2900.)
50.
1.45
3500.
2900.
40.
1.2625
2625.
2175.
30.
1.075
1750.
1450.
20.
.8875
875.
725.
10.
.7
0.
0.
0. 20. 40. 60. 80. 100. 120. 140. 160. 180. 200. 220.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE F17

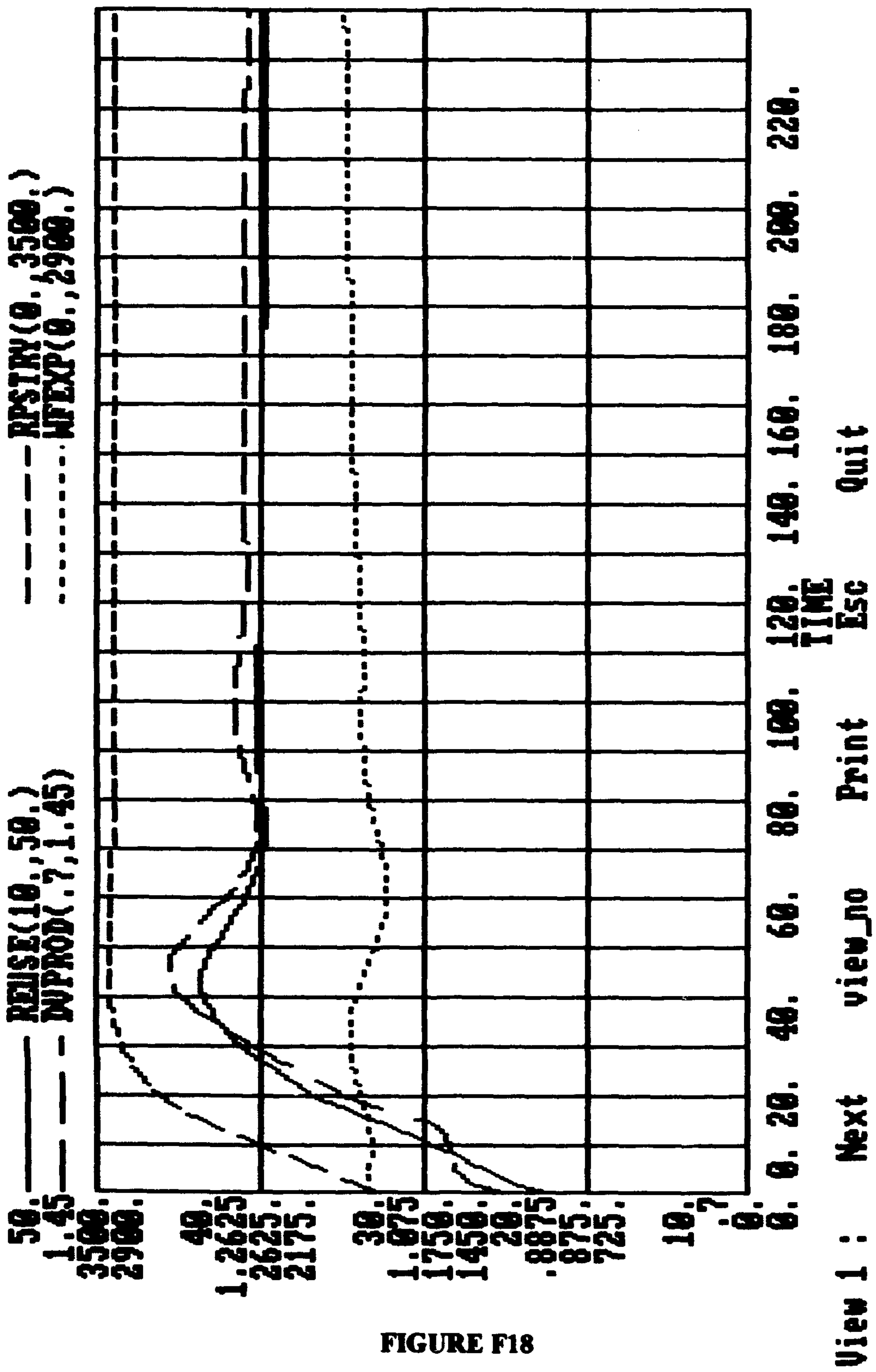

REUSE(10.,50.)
DVPROD(.7,1.45)
WFEXP(0.,2900.)
TIME
View 1 : Next view_no Print Esc Quit

FIGURE F18

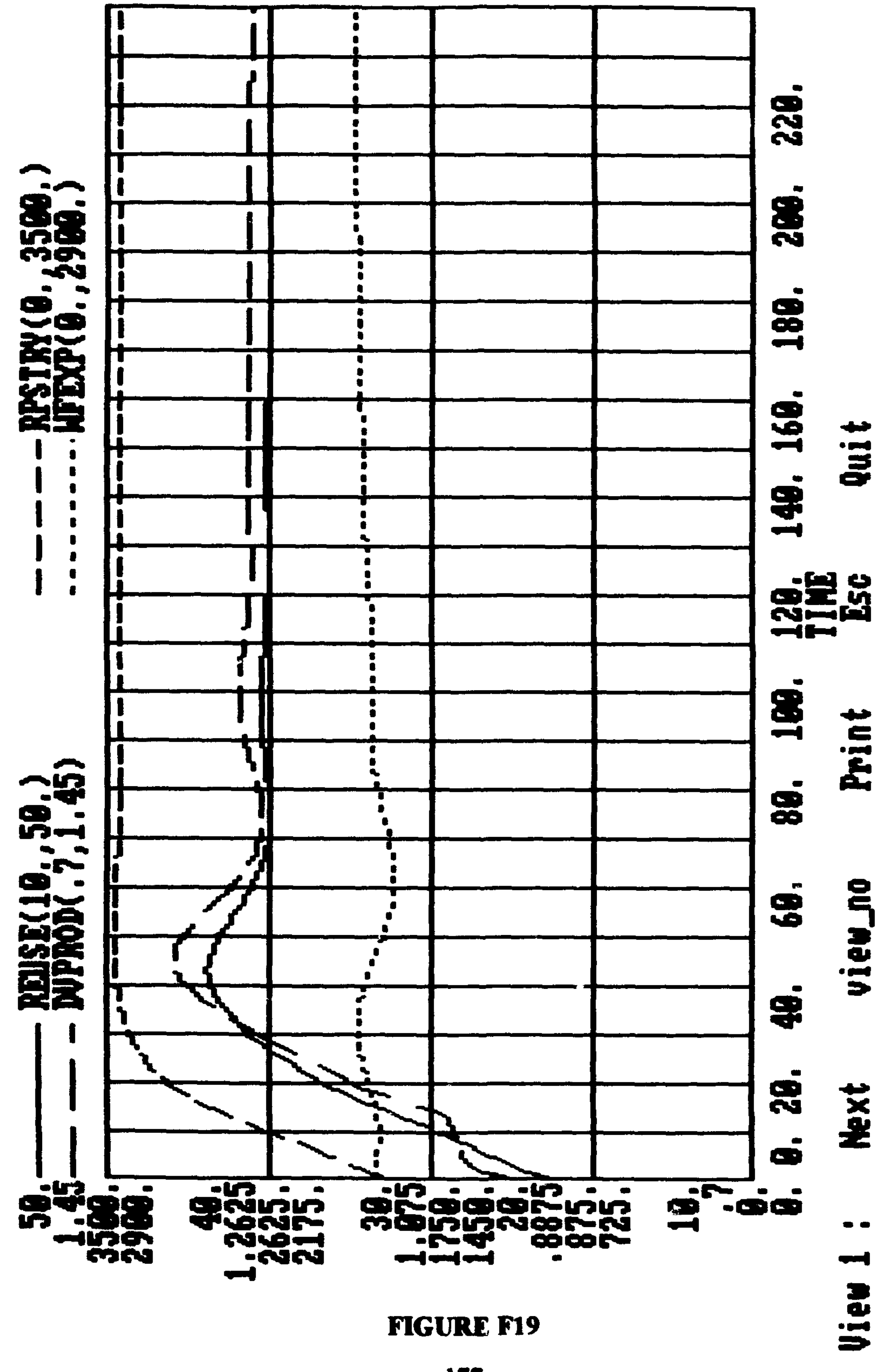

REUSE(10.,50.)
DVPROD(.7,1.45)
RPSTMY(0.,3500.)
WFEXP(0.,2900.)
TIME
View 1 : Next view_no Print Esc Quit

FIGURE F19

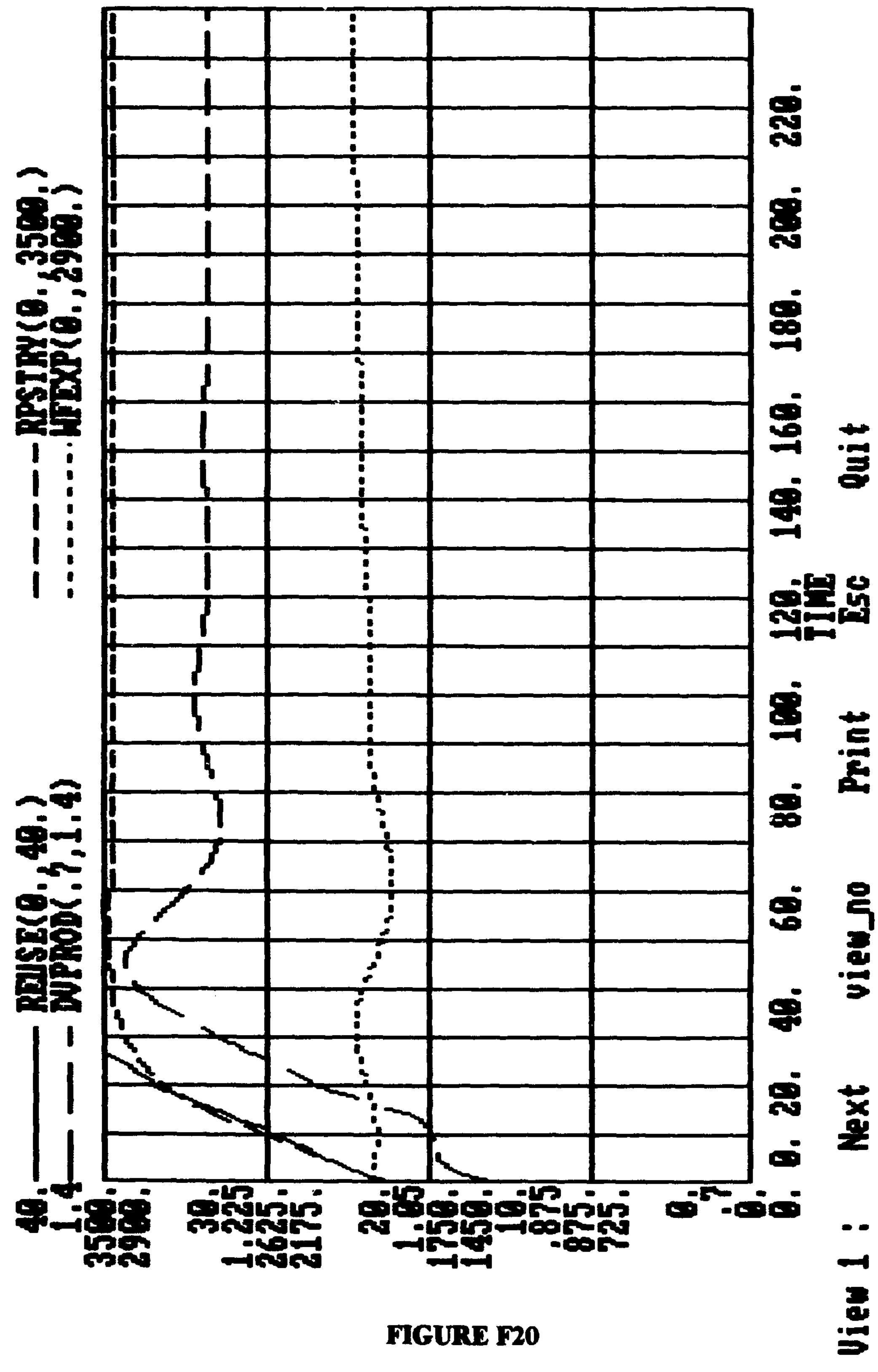

REUSE(0.,40.)
DUPROD(.7,1.4)
RPSTRY(0.,3500.)
WFEXP(0.,2900.)
0. 20. 40. 60. 80. 100. 120. 140. 160. 180. 200. 220.
TIME
View 1 : Next view_no Print Esc Quit

FIGURE F20

TABLE F4

Model = REUSE6; Run = R_AGE20.RSL; Change = NMRCLF = 20

Years	DVPROD	REUSE	RPSTRY	WFEXP
0.00	0.98	22.60	2,000.00	1,700.00
1.00	0.94	19.81	1,750.43	1,701.86
2.00	0.91	19.16	1,713.34	1,841.41
3.00	0.92	19.76	1,753.75	2,007.50
4.00	0.94	20.48	1,802.08	2,095.55
5.00	0.94	20.91	1,841.23	2,145.02
6.00	0.93	21.32	1,883.87	2,209.25
7.00	0.94	21.87	1,931.83	2,287.19
8.00	0.94	22.41	1,974.87	2,355.65
9.00	0.94	22.81	2,008.78	2,409.65
10.00	0.95	23.14	2,038.17	2,458.87
11.00	0.95	23.46	2,066.52	2,508.49
12.00	0.95	23.76	2,093.13	2,555.67
13.00	0.95	24.04	2,116.58	2,597.60
14.00	0.95	24.27	2,137.22	2,635.29
15.00	0.95	24.48	2,156.14	2,670.66
16.00	0.95	24.68	2,173.69	2,703.96
17.00	0.95	24.87	2,189.58	2,734.59
18.00	0.95	25.03	2,203.78	2,762.48
19.00	0.95	25.17	2,216.63	2,788.19
20.00	0.95	25.30	2,228.40	2,812.08

TABLE F5

Model = REUSE6; Run = R_AGE30.RSL; Change = NMRCLF = 30

Years	DVPROD	REUSE	RPSTRY	WFEXP
0.00	0.98	22.60	2,000.00	1,700.00
1.00	0.97	23.47	2,079.21	1,697.39
2.00	0.96	25.43	2,238.04	1,818.00
3.00	1.00	28.03	2,427.50	1,956.55
4.00	1.04	30.34	2,581.18	2,003.48
5.00	1.04	31.47	2,669.62	2,005.22
6.00	1.04	31.97	2,726.00	2,025.51
7.00	1.05	32.46	2,768.70	2,064.59
8.00	1.06	32.69	2,779.38	2,094.51
9.00	1.06	32.50	2,768.92	2,108.74
10.00	1.05	32.20	2,756.42	2,124.10
11.00	1.05	32.06	2,750.31	2,149.23
12.00	1.05	32.05	2,749.64	2,176.88
13.00	1.05	32.05	2,749.58	2,199.94
14.00	1.05	32.01	2,747.38	2,219.07
15.00	1.05	31.95	2,743.81	2,237.40
16.00	1.05	31.89	2,740.32	2,255.66
17.00	1.04	31.83	2,737.27	2,272.85
18.00	1.04	31.78	2,734.44	2,288.58
19.00	1.04	31.74	2,731.72	2,303.18
20.00	1.04	31.69	2,729.16	2,316.87

TABLE F6

Model = REUSE6; Run = R_AGE40.RSL; Change = NMRCLF = 40

Years	DVPROD	REUSE	RPSTRY	WFEXP
0.00	0.98	22.60	2,000.00	1,700.00
1.00	1.00	25.61	2,269.39	1,694.96
2.00	1.00	29.55	2,574.64	1,804.07
3.00	1.08	34.21	2,876.65	1,920.76
4.00	1.17	36.79	2,971.49	1,925.65
5.00	1.16	36.09	2,931.62	1,863.79
6.00	1.12	34.41	2,876.37	1,832.10
7.00	1.11	33.70	2,859.46	1,849.27
8.00	1.11	33.74	2,863.21	1,880.17
9.00	1.11	33.93	2,868.50	1,903.97
10.00	1.12	34.00	2,870.18	1,921.20
11.00	1.11	33.98	2,869.85	1,938.08
12.00	1.11	33.93	2,869.18	1,955.73
13.00	1.11	33.88	2,868.47	1,971.94
14.00	1.11	33.83	2,867.68	1,985.97
15.00	1.11	33.77	2,866.99	1,998.78
16.00	1.11	33.72	2,866.51	2,010.97
17.00	1.11	33.69	2,866.15	2,022.27
18.00	1.11	33.65	2,865.78	2,032.41
19.00	1.11	33.62	2,865.41	2,041.56
20.00	1.11	33.59	2,865.07	2,049.97

TABLE F7

Model = REUSE6; Run = R_AGE50.RSL; Change = NMRCLF = 50

Years	DVPROD	REUSE	RPSTRY	WFEXP
0.00	0.98	22.60	2,000.00	1,700.00
1.00	1.01	27.01	2,392.77	1,693.45
2.00	1.03	32.36	2,806.40	1,794.80
3.00	1.17	37.34	3,043.80	1,889.05
4.00	1.23	38.6[illegible]	3,046.92	1,851.54
5.00	1.19	36.[illegible]	[illegible],990.32	1,766.74
6.00	1.14	35.20	2,959.13	1,748.77
7.00	1.13	34.90	2,357.22	1,802.74
8.00	1.15	35.35	2,964.73	1,851.30
9.00	1.16	35.57	2,966.82	1,864.58
10.00	1.15	35.42	2,964.31	1,869.63
11.00	1.15	35.27	2,962.60	1,886.51
12.00	1.15	35.25	2,962.56	1,907.78
13.00	1.15	35.25	2,962.24	1,922.83
14.00	1.15	35.18	2,961.13	1,932.99
15.00	1.15	35.10	2,960.13	1,943.65
16.00	1.15	35.06	2,959.67	1,955.43
17.00	1.15	35.03	2,959.35	1,966.02
18.00	1.15	35.00	2,958.87	1,974.72
19.00	1.15	34.96	2,958.34	1,982.53
20.00	1.15	34.92	2,957.91	1,990.12

TABLE F8

Model = REUSE6; Run = R_AGE60.RSL; Change = NMRCLF = 60

Years	DVPROD	REUSE	RPSTRY	WFEXP
0.00	0.98	22.60	2,000.00	1,700.00
1.00	1.02	27.96	2,479.16	1,692.43
2.00	1.09	34.26	2,952.78	1,788.13
3.00	1.22	38.96	3,118.07	1,861.16
4.00	1.26	39.60	3,096.42	1,803.24
5.00	1.21	37.60	3,049.40	1,716.58
6.00	1.16	35.98	3,026.05	1,721.21
7.00	1.16	35.97	3,027.64	1,786.75
8.00	1.18	36.48	3,033.45	1,826.76
9.00	1.18	36.56	3,033.21	1,830.95
10.00	1.18	36.33	3,030.50	1,836.02
11.00	1.17	36.21	3,029.58	1,855.46
12.00	1.18	36.23	3,029.91	1,875.68
13.00	1.18	36.23	3,029.50	1,887.68
14.00	1.17	36.15	3,028.43	1,896.10
15.00	1.17	36.07	3,027.69	1,906.30
16.00	1.17	36.04	3,027.43	1,917.41
17.00	1.17	36.01	3,027.19	1,926.76
18.00	1.17	35.98	3,026.76	1,934.30
19.00	1.17	35.94	3,026.33	1,941.34
20.00	1.17	35.91	3,026.02	1,948.30

TABLE F9

Model = REUSE6; Run = R_AGE70.RSL; Change = NMRCLF = 70

Years	DVPROD	REUSE	RPSTRY	WFEXP
0.00	0.98	22.60	2,000.00	1,700.00
1.00	1.03	28.60	2,542.99	1,691.70
2.00	1.12	35.47	3,037.09	1,782.61
3.00	1.25	40.01	3,165.82	1,840.15
4.00	1.28	40.38	3,139.40	1,770.25
5.00	1.23	38.19	3,099.03	1,687.62
6.00	1.18	36.70	3,080.60	1,705.47
7.00	1.18	36.83	3,082.84	1,772.74
8.00	1.20	37.32	3,087.00	1,805.85
9.00	1.20	37.31	3,085.84	1,806.22
10.00	1.20	37.06	3,083.32	1,812.30
11.00	1.19	36.97	3,082.85	1,832.56
12.00	1.20	37.01	3,083.27	1,851.26
13.00	1.20	37.00	3,082.85	1,861.29
14.00	1.20	36.91	3,081.90	1,868.74
15.00	1.19	36.84	3,081.34	1,878.48
16.00	1.19	36.82	3,081.18	1,888.84
17.00	1.19	36.79	3,080.97	1,897.21
18.00	1.19	36.76	3,080.60	1,903.91
19.00	1.19	36.72	3,080.25	1,910.33
20.00	1.19	36.70	3,080.02	1,916.73

TABLE F10

Model = REUSE6; Run = R_AGE80.RSL; Change = NMRCLF = 80

Years	DVPROD	REUSE	RPSTRY	WFEXP
0.00	0.98	22.60	2,000.00	1,700.00
1.00	1.03	29.10	2,592.05	1,691.16
2.00	1.15	36.34	3,094.06	1,777.84
3.00	1.26	40.58	3,209.85	1,824.54
4.00	1.30	41.01	3,190.54	1,749.00
5.00	1.24	38.85	3,146.23	1,669.99
6.00	1.20	37.36	3,126.28	1,692.78
7.00	1.20	37.50	3,126.98	1,759.39
8.00	1.22	37.96	3,129.88	1,789.12
9.00	1.22	37.92	3,128.45	1,787.97
10.00	1.21	37.66	3,126.15	1,794.33
11.00	1.21	37.58	3,125.83	1,814.45
12.00	1.21	37.64	3,126.23	1,832.03
13.00	1.21	37.62	3,125.82	1,840.90
14.00	1.21	37.53	3,124.97	1,847.65
15.00	1.21	37.47	3,124.51	1,856.88
16.00	1.21	37.44	3,124.40	1,866.59
17.00	1.21	37.42	3,124.23	1,874.24
18.00	1.21	37.39	3,123.91	1,880.32
19.00	1.21	37.35	3,123.62	1,886.23
20.00	1.21	37.33	3,123.43	1,892.14

TABLE F11

Model = REUSE6; Run = R_AGE90.RSL; Change = NMRCLF = 90

Years	DVPROD	REUSE	RPSTRY	WFEXP
0.00	0.98	22.60	2,000.00	1,700.00
1.00	1.04	29.49	2,630.92	1,690.73
2.00	1.16	36.99	3,135.60	1,773.79
3.00	1.28	41.02	3,250.57	1,813.00
4.00	1.31	41.41	3,240.97	1,734.16
5.00	1.26	39.37	3,202.01	1,658.85
6.00	1.22	38.03	3,174.40	1,683.92
7.00	1.22	38.12	3,166.43	1,747.32
8.00	1.24	38.48	3,165.59	1,774.36
9.00	1.23	38.40	3,163.51	1,773.51
10.00	1.23	38.15	3,161.61	1,780.58
11.00	1.23	38.09	3,161.47	1,800.06
12.00	1.23	38.15	3,161.83	1,816.41
13.00	1.23	38.14	3,161.43	1,824.52
14.00	1.23	38.05	3,160.67	1,830.90
15.00	1.23	37.99	3,160.28	1,839.68
16.00	1.23	37.96	3,160.18	1,848.77
17.00	1.23	37.95	3,160.03	1,855.85
18.00	1.23	37.91	3,159.76	1,861.47
19.00	1.22	37.88	3,159.51	1,866.97
20.00	1.22	37.85	3,159.36	1,872.46

TABLE F12

Model = REUSE6; Run = R_AGE100.RSL; Change = NMRCLF = 100

Years	DVPROD	REUSE	RPSTRY	WFEXP
0.00	0.98	22.60	2,000.00	1,700.00
1.00	1.04	29.81	2,662.48	1,690.39
2.00	1.18	37.50	3,167.34	1,770.38
3.00	1.29	41.38	3,285.85	1,804.17
4.00	1.32	41.77	3,284.44	1,722.92
5.00	1.27	39.71	3,251.86	1,650.24
6.00	1.23	38.42	3,228.10	1,677.73
7.00	1.23	38.55	3,221.06	1,739.97
8.00	1.25	38.90	3,219.34	1,764.75
9.00	1.25	38.81	3,216.17	1,763.07
10.00	1.24	38.56	3,213.11	1,770.21
11.00	1.24	38.51	3,212.01	1,789.38
12.00	1.24	38.56	3,211.91	1,805.03
13.00	1.24	38.54	3,211.32	1,812.55
14.00	1.24	38.46	3,210.31	1,818.52
15.00	1.24	38.40	3,209.59	1,826.86
16.00	1.24	38.38	3,209.24	1,835.47
17.00	1.24	38.36	3,208.93	1,842.11
18.00	1.24	38.33	3,208.52	1,847.35
19.00	1.24	38.30	3,208.12	1,852.48
20.00	1.24	38.27	3,207.82	1,857.60

TABLE F13

Model = REUSE6; Run = R_AGE110.RSL; Change = NMRCLF = 110

Years	DVPROD	REUSE	RPSTRY	WFEXP
0.00	0.98	22.60	2,000.00	1,700.00
1.00	1.04	30.07	2,688.61	1,690.11
2.00	1.18	37.77	3,193.32	1,767.48
3.00	1.30	41.72	3,316.60	1,797.18
4.00	1.33	42.11	3,321.91	1,714.03
5.00	1.28	40.01	3,294.59	1,643.24
6.00	1.24	38.76	3,273.99	1,672.40
7.00	1.24	38.91	3,267.90	1,733.65
8.00	1.26	39.25	3,266.16	1,756.78
9.00	1.26	39.14	3,263.00	1,754.56
10.00	1.25	38.89	3,260.11	1,761.82
11.00	1.25	38.84	3,259.11	1,780.78
12.00	1.25	38.90	3,259.01	1,795.84
13.00	1.25	38.88	3,258.44	1,802.80
14.00	1.25	38.80	3,257.50	1,808.43
15.00	1.25	38.74	3,256.84	1,816.48
16.00	1.25	38.72	3,256.52	1,824.74
17.00	1.25	38.71	3,256.24	1,830.99
18.00	1.25	38.68	3,255.86	1,835.90
19.00	1.25	38.64	3,255.50	1,840.75
20.00	1.25	38.62	3,255.24	1,845.59

TABLE F14

Model = REUSE6; Run = R_AGE120.RSL; Change = NMRCLF = 120

Years	DVPROD	REUSE	RPSTRY	WFEXP
0.00	0.98	22.60	2,000.00	1,700.00
1.00	1.05	30.29	2,710.60	1,689.88
2.00	1.19	38.00	3,215.86	1,765.01
3.00	1.30	42.02	3,343.42	1,791.50
4.00	1.34	42.41	3,354.38	1,706.79
5.00	1.28	40.28	3,331.52	1,637.41
6.00	1.24	39.05	3,313.59	1,667.77
7.00	1.25	39.22	3,308.36	1,728.19
8.00	1.27	39.56	3,306.69	1,750.01
9.00	1.26	39.43	3,303.59	1,747.37
10.00	1.26	39.18	3,300.88	1,754.71
11.00	1.26	39.14	3,299.97	1,773.46
12.00	1.26	39.20	3,299.87	1,788.01
13.00	1.26	39.18	3,299.32	1,794.51
14.00	1.26	39.10	3,298.43	1,799.85
15.00	1.26	39.04	3,297.82	1,807.65
16.00	1.26	39.02	3,297.54	1,815.61
17.00	1.26	39.01	3,297.28	1,821.55
18.00	1.26	38.98	3,296.93	1,826.17
19.00	1.25	38.94	3,296.60	1,830.78
20.00	1.25	38.92	3,296.36	1,835.38

TABLE F15

Model = REUSE6; Run = R_AGE130.RSL; Change = NMRCLF = 130

Years	DVPROD	REUSE	RPSTRY	WFEXP
0.00	0.98	22.60	2,000.00	1,700.00
1.00	1.05	30.48	2,729.36	1,689.68
2.00	1.20	38.20	3,235.46	1,762.90
3.00	1.31	42.28	3,366.91	1,786.79
4.00	1.35	42.68	3,382.70	1,700.77
5.00	1.29	40.51	3,363.71	1,632.46
6.00	1.25	39.30	3,348.10	1,663.71
7.00	.26	39.49	3,343.64	1,723.43
8.00	1.27	39.82	3,342.09	1,744.18
9.00	1.27	39.68	3,339.11	1,741.20
10.00	1.26	39.43	3,336.56	1,748.58
11.00	1.26	39.40	3,335.73	1,767.15
12.00	1.27	39.46	3,335.64	1,781.26
13.00	1.27	39.44	3,335.11	1,787.37
14.00	1.26	39.36	3,334.27	1,792.46
15.00	1.26	39.30	3,333.71	1,800.05
16.00	1.26	39.29	3,333.45	1,807.74
17.00	1.26	39.27	3,333.20	1,813.42
18.00	1.26	39.24	3,332.88	1,817.80
19.00	1.26	39.21	3,332.58	1,822.19
20.00	1.26	39.19	3,332.36	1,826.60

TABLE F16

Model = REUSE6; Run = R_AGE140.RSL; Change = NMRCLF = 140

Years	DVPROD	REUSE	RPSTRY	WFEXP
0.00	0.98	22.60	2,000.00	1,700.00
1.00	1.05	30.65	2,745.55	1,689.51
2.00	1.20	38.36	3,252.61	1,761.08
3.00	1.32	42.51	3,387.58	1,782.82
4.00	1.35	42.92	3,407.58	1,695.67
5.00	1.30	40.72	3,391.98	1,628.18
6.00	1.26	39.53	3,378.41	1,660.14
7.00	1.27	39.73	3,374.67	1,719.25
8.00	1.28	40.05	3,373.28	1,739.08
9.00	1.28	39.91	3,370.44	1,735.74
10.00	1.27	39.66	3,368.04	1,743.11
11.00	1.27	39.62	3,367.29	1,761.55
12.00	1.27	39.69	3,367.21	1,775.35
13.00	1.27	39.67	3,366.70	1,781.14
14.00	1.27	39.59	3,365.91	1,786.00
15.00	1.27	39.53	3,365.39	1,793.41
16.00	1.27	39.52	3,365.15	1,800.90
17.00	1.27	39.50	3,364.93	1,806.34
18.00	1.27	39.47	3,364.63	1,810.52
19.00	1.27	39.44	3,364.35	1,814.73
20.00	1.27	39.42	3,364.15	1,818.96

TABLE F17

Model = REUSE6; Run = R_AGE150.RSL; Change = NMRCLF = 150

Years	DVPROD	REUSE	RPSTRY	WFEXP
0.00	0.98	22.60	2,000.00	1,700.00
1.00	1.05	30.79	2,759.67	1,689.36
2.00	1.20	38.51	3,267.70	1,759.49
3.00	1.32	42.71	3,405.89	1,779.42
4.00	1.36	43.13	3,429.59	1,691.29
5.00	1.30	40.90	3,416.99	1,624.46
6.00	1.26	39.72	3,405.25	1,656.97
7.00	1.27	39.94	3,402.17	1,715.56
8.00	1.29	40.26	3,400.96	1,734.59
9.00	1.29	40.11	3,398.27	1,730.91
10.00	1.28	39.85	3,396.02	1,738.25
11.00	1.28	39.82	3,395.34	1,756.59
12.00	1.28	39.89	3,395.28	1,770.13
13.00	1.28	39.87	3,394.80	1,775.65
14.00	1.28	39.79	3,394.05	1,780.32
15.00	1.28	39.74	3,393.56	1,787.56
16.00	1.28	39.72	3,393.35	1,794.88
17.00	1.28	39.71	3,393.14	1,800.13
18.00	1.28	39.68	3,392.87	1,804.13
19.00	1.28	39.65	3,392.61	1,808.18
20.00	1.28	39.63	3,392.42	1,812.26

TABLE F18

Model = REUSE6; Run = R_AGE160.RSL; Change = NMRCLF = 160

Years	DVPROD	REUSE	RPSTRY	WFEXP
0.00	0.98	22.60	2,000.00	1,700.00
1.00	1.05	30.92	2,772.08	1,689.23
2.00	1.21	38.64	3,281.07	1,758.11
3.00	1.33	42.89	3,422.20	1,776.47
4.00	1.37	43.32	3,449.19	1,687.48
5.00	1.31	41.07	3,439.28	1,621.19
6.00	1.27	39.90	3,429.17	1,654.14
7.00	1.28	40.12	3,426.71	1,712.27
8.00	1.29	40.44	3,425.70	1,730.61
9.00	1.29	40.28	3,423.17	1,726.63
10.00	1.28	40.03	3,421.05	1,733.94
11.00	1.28	40.00	3,420.44	1,752.18
12.00	1.28	40.07	3,420.40	1,765.49
13.00	1.29	40.05	3,419.94	1,770.78
14.00	1.28	39.97	3,419.24	1,775.28
15.00	1.28	39.92	3,418.78	1,782.39
16.00	1.28	39.91	3,418.58	1,789.54
17.00	1.28	39.89	3,418.39	1,794.63
18.00	1.28	39.86	3,418.13	1,798.47
19.00	1.28	39.83	3,417.89	1,802.39
20.00	1.28	39.82	3,417.72	1,806.34

TABLE F19

Model = REUSE6; Run = R_AGE170.RSL; Change = NMRCLF = 170

Years	DVPROD	REUSE	RPSTRY	WFEXP
0.00	0.98	22.60	2,000.00	1,700.00
1.00	1.05	31.03	2,783.09	1,689.12
2.00	1.21	38.75	3,292.99	1,756.89
3.00	1.33	43.06	3,436.82	1,773.89
4.00	1.37	43.49	3,466.74	1,684.14
5.00	1.31	41.22	3,459.25	1,618.28
6.00	1.27	40.06	3,450.62	1,651.60
7.00	1.28	40.29	3,448.73	1,709.33
8.00	1.30	40.61	3,447.92	1,727.07
9.00	1.30	40.45	3,445.55	1,722.82
10.00	1.29	40.19	3,443.57	1,730.09
11.00	1.29	40.16	3,443.03	1,748.24
12.00	1.29	40.23	3,443.00	1,761.35
13.00	1.29	40.22	3,442.57	1,766.43
14.00	1.29	40.14	3,441.90	1,770.78
15.00	1.29	40.08	3,441.48	1,777.77
16.00	1.29	40.07	3,441.30	1,784.79
17.00	1.29	40.06	3,441.12	1,789.72
18.00	1.29	40.03	3,440.88	1,793.43
19.00	1.29	40.00	3,440.65	1,797.23
20.00	1.29	39.98	3,440.50	1,801.06

TABLE F20

Model = REUSE6; Run = R_AGE180.RSL; Change = NMRCLF = 180

Years	DVPROD	REUSE	RPSTRY	WFEXP
0.00	0.98	22.60	2,000.00	1,700.00
1.00	1.05	31.13	2,792.92	1,689.02
2.00	1.21	38.85	3,303.69	1,755.80
3.00	1.34	43.20	3,449.97	1,771.61
4.00	1.38	43.65	3,482.54	1,681.18
5.00	1.31	41.35	3,477.24	1,615.68
6.00	1.28	40.20	3,469.96	1,649.30
7.00	1.29	40.44	3,468.60	1,706.68
8.00	1.30	40.76	3,468.00	1,723.89
9.00	1.30	40.59	3,465.79	1,719.39
10.00	1.29	40.33	3,463.93	1,726.62
11.00	1.29	40.31	3,463.46	1,744.69
12.00	1.29	40.38	3,463.45	1,757.63
13.00	1.29	40.37	3,463.04	1,762.53
14.00	1.29	40.28	3,462.41	1,766.75
15.00	1.29	40.23	3,462.01	1,773.63
16.00	1.29	40.22	3,461.85	1,780.53
17.00	1.29	40.21	3,461.68	1,785.33
18.00	1.29	40.18	3,461.45	1,788.91
19.00	1.29	40.15	3,461.24	1,792.60
20.00	1.29	40.13	3,461.10	1,796.34

LIST OF REFERENCES

Abdel-Hamid, Tarek K., "Modeling the Dynamics of Software Reuse: an Integrated System Dynamics Perspective," paper presented at the Sixth Annual Workshop on Software Reuse (WISR '93), Owego, New York, 2-4 November 1993.

Abdel-Hamid, Tarek K. and Stuart E. Madnick, *Software Development Dynamics: An Integrated Approach*, pp. 6-7, Prentice-Hall, Inc., 1990.

Allen, K., William Krutz and Dan Olivier, "Software Reuse: Mining, Refining, and Designing," *Tri-Ada '90*, Baltimore, Maryland, pp. 222-226, 3-7 December 1990.

Bailey, John W. and Victor R. Basili, "Software Reclamation: Improving Post-Development Reusability," *Proceedings of the Eighth Annual National Conference on Ada Technology*, Atlanta, Georgia, pp. 477-499, 5-8 March 1990.

Banker, Rajiv D., Robert J. Kauffman, and Dani Zweig, "Repository Evaluation of Software Reuse," *IEEE Transactions on Software Engineering*, v. 19, n. 4, pp. 379-389, April 1993.

Bollinger, Terry B. and Shari Lawrence Pfleeger, "The Economics of Reuse: Issues and Alternatives," *Proceedings of the Eighth Annual National Conference on Ada Technology*, Atlanta, Georgia, pp. 436-447, 5-8 March 1990.

Cruickshank, R. D. and J. E. Gaffney, "The Economics of Software Reuse," *Software Productivity Consortium* SPC-92119-CMC, Herndon, Virginia, pp. 1-50, September 1991.

Drake, R. and W. Ett, "Reuse: the Two Concurrent Life Cycles Paradigm," *Tri-Ada'90*, Baltimore, Maryland, pp. 208-221, 3-7 December 1990.

Durin, Bernard, Jeannette Abadir, and Bernard Mouton, "Software Reuse--the Challenge of the 90's in Software Development," *International Symposium on Space Information Systems*, 2nd, pp. 1010-1018, September 1990.

Emery, James C. and Dani Zweig, "The Use of Ada for the Implementation of Automated Information Systems within the Department of Defense," paper from Information Systems Management (IS4182) class at Naval Postgraduate School, Monterey, California, January 1994.

Endoso, Joyce, "Audit says DOD's Software Reuse Plan Faces Many Barriers," *Government Computer News*, v. 12, no. 6, p. 46, 15 March 1993.

Endoso, Joyce, "AF Rates Software Building Skills; Service Will Use Two Methods to Evaluate Proposals for Systems Buys," *Government Computer News*, v. 12, no. 18, p.58, 30 August 1993.

Endoso, Joyce, "House Puts Teeth into DOD Systems Unification," *Government Computer News*, v. 12, no. 22, p. 8, 11 October 1993.

Favaro, J., "What Price Reusability? A Case Study," *Ada Letters*, v. XI, n. 3, pp. 115-124, Spring 1991.

Ferris, Nancy, "Defense Plans to Reuse Code in Languages Other than Ada, DISA CIM official says (Dennis Brown, director of Defense Information Systems Agency's Center for Information Management)," *Government Computer News*, v. 12, no. 1, p. 44, 4 January 1993.

Gaffney, J. E. and R. D. Cruickshank, "A General Economics Model of Software Reuse," *International Conference on Software Engineering*, Melbourne, Australia, pp. 327-337, May 1992.

Gaffney, J. E. and T. A. Durek, "Software Reuse--Key to Enhanced Productivity: Some Quantitative Models," *Information and Software Technology*, v. 31, n. 5, pp. 258-267, June 1989.

Gall, Harald and Rene Klosch, "Reuse Engineering: Software Construction from Reusable Components," *IEEE*, pp. 79-86, 1992.

GAO/IMTEC-93-16 Report (U. S. General Accounting Office, Information Management and Technology Division Report), *Software Reuse--Major Issues Need to be Resolved Before Benefits can be Achieved*, pp. 1-21, 28 January 1993.

Green, Robert, "Why Waste Perfectly Good Code? Ada is Sign of Thrifty Times," *Government Computer News*, v. 11, n. 26, p. 58, 21 December 1992.

Hooper, James W. and Rowena O. Chester, "Software Reuse: Managerial and Technical Guidelines," *Proceedings of Eighth Annual National Conference on Ada Technology*, Fort Monmouth, New Jersey, pp. 424-35, 1990.

Incorvaia, Angelo. J., Alan M. Davis and Richard E. Fairley, "Case Studies in Software Reuse," *Proceedings of the Fourteenth Annual International Computer Software and Applications Conference*, pp. 301-306, 1990.

Isoda, Sadahiro, "An Experience of Software Reuse Activities," *Proceedings of the Fifteenth Annual International Computer Software and Applications Conference*, Tokyo, Japan, pp. 8-9, 11 September 1991.

Joiner, Harry F., "Economic Analysis of Software Reuse--A Model Study," *Proceedings of Eighth Annual National Conference on Ada Technology*, Fort Monmouth, New Jersey, pp. 448-51, 1990.

Kang, K. C. and L. S. Levy, "Software Methodology in the Harsh Light of Economics," *Information and Software Technology*, v. 31, n. 5, pp. 239-250, June 1989.

Kim, Yongbeom and Edward A. Stohr, "Software Reuse: Issues and Research Directions," *Proceedings of the Hawaii International Conference on System Sciences*, v. 4, pp. 612-623, 1992.

Kitfield, James, "Is Software DOD's Achilles' Heel?," *Military Forum*, pp. 28-35, July 1989.

Margono, Johan and Thomas. E. Rhoads, "Software Reuse Economics: Cost-Benefit Analysis on a Large-Scale Ada Project," *International Conference on Software Engineering*, Melbourne, Australia, pp. 338-348, May 1992.

Matsumura, Kazuo, Akihiro Yamashiro, Toshiyuki Tanaka, and Ikumme Takahashi, "Modeling of Software Reusable Component Approach and its Case Study," *Proceedings of the Fourteenth Annual International Computer Software and Applications Conference*, pp. 307-313, 1990.

Menke, Susan M., "Defense Users, Executives Want to Know Where Ada Effort Stands (Department of Defense official Peter A. Kind speaking at the 11th national conference on Ada technology in Williamsburg, Virginia)," *Government Computer News*, v. 12, n. 7, p. 3, 29 March 1993.

Mii, N. and T. Takeshita, "Software Re-engineering and Reuse from a Japanese Point of View," *Information and Software Technology*, v. 35, n. 1, p. 45-53, January 1993.

O'Brien, Larry, "The Billion Dollar Vision. (Microsoft Chairman Bill Gates' keynote address at the Software Development '93 conference)," *AI Expert*, v. 8, n. 5, p. 5, May 1993.

Palmer, Constance and Sholom Cohen, "Engineering and Application of Reusable Software Resources," *Aerospace Software Engineering--A Collection of Concepts*, Washington, D.C., American Institute of Aeronautics and Astronautics, Inc., pp. 415-440, 1990.

Prieto-Diaz, Ruben and P. Freeman, "Classifying Software for Reusability," *IEEE Software*, vol. 4, no. 1, pp. 6-16, January 1987.

Prieto-Diaz, Ruben, "Status Report: Software Reusability," *IEEE Software*, v. 10, n. 3, p. 61, May 1993.

Reifer, Donald J., "Reuse Metrics and Measurement--A Framework," *NASA/Goddard Fifteenth Annual Software Engineering Workshop*, pp. 1-20, 28 November 1990.

Reynolds, Robert G., Jonathan I. Maletic and Elena Zannoni, "Operationalizing Software Reuse as a Problem in Inductive Learning," *Proceedings of Fifth International Conference IEA/AIE-92*, Paderborn, Germany, pp. 143-153, 9-12 June 1992.

Rombach, H. Dieter, "A Comprehensive Approach to Reuse," *Proceedings of the Second European Conference on Software Quality Assurance*, pp. 1-20, 1990.

Rymer, John R, "The Case for Reuse," *Distributed Computing Monitor*, v. 8, n. 1, p. 2, January 1993.

Schimsky, David, "Software Reuse--Some Realities," *Vitro Technical Journal*, v. 10, n. 1, pp. 47-57, Summer 1992.

Schwartz, Karen D., "General urges new approach to software development (Major General Albert J. Edmonds)," *Government Computer News*, v. 11, n. 25, p. 94, 7 December 1992.

Schwartz, Karen D., "DoD Reuse Libraries Get Linked Today. (Department of Defense project boosts software engineering efficiency)," *Government Computer News*, v. 12, n. 8, p. 1, 12 April 1993.

Schwartz, Karen D., "Where to Find Free Software," *Government Computer News*, v. 12, n. 10, p. 50, 10 May 10, 1993.

Sutcliffe, Alistair and Neil Maiden, "Software Reusability: Delivering Productivity Gains or Short Cuts," *Proceedings of the IFIP TC 13 Third International Conference on Human-Computer Interaction*, pp. 895-901, 1990.

Tirso, Jesus R., "Establishing a Software Reuse Support Structure," *IEEE Software*, pp. 1500-1504, 1991.

Williams, L. G., "Software Reuse--Definition and Overview," *Aerospace Software Engineering--A Collection of Concepts*, Washington, D.C., American Institute of Aeronautics and Astronautics, Inc., pp. 395-414, 1991.

Wolff, F., "Long-Term Controlling of Software Reuse," *Information and Software Technology*, v. 34, n. 3, pp. 178-184, March 1992.

Yourdon, Edward. Talk presented to Information Systems faculty and students at Naval Postgraduate School, Monterey, California, 3 March 1994.

INITIAL DISTRIBUTION LIST

		Number of Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 052 Naval Postgraduate School Monterey, California 93943-5002	2
3.	George S. Coker Software Reuse Program Management Office Defense Information Systems Agency Center for Information Management 500 N. Washington Street Suite 200 Falls Church, VA 22046	1
4.	Gary Shupe Head, Software Engineering Branch Software and Data Management Division Naval Computer and Telecommunications Command 4401 Massachusetts Avenue, N. W. Washington, D.C. 20394-5000	1
5.	Dr. Michael Mestrovich Office of Integration and Interoperabilility 5201 Leesburg Pike 3 Skyline Place Suite 1501 Falls Church, VA 22041-3201	1
6.	Donald J. Reifer Executive Administrator DoD Software Initiatives Defense Information Systems Agency Center for Information Management 1951 Kidwell Dr., Rm. 521 Vienna, VA 22182	1

7.	Computer Technology, Code 37 Naval Postgraduate School Monterey, California 93943-5002	1
8.	Tarek Abdel-Hamid, Code SM/Hc Department of Systems Management Naval Postgraduate School Monterey, California 93943-5002	1
9.	James T. Emery, Code SM/Hg Department of Systems Management Naval Postgraduate School Monterey, California 93943-5002	1
10.	LT Rick Arai, Code 37 Naval Postgraduate School Monterey, California 93943-5002	1
11.	LCDR Pamela J. Gallup Naval Computer and Telecommunications Area Master Station, Eastern Pacific Wahiawa, Hawaii 96786-3050	2
12.	CDR William B. Short, Code SM/Sh Department of Systems Management Naval Postgraduate School Monterey, California 93943-5002	1