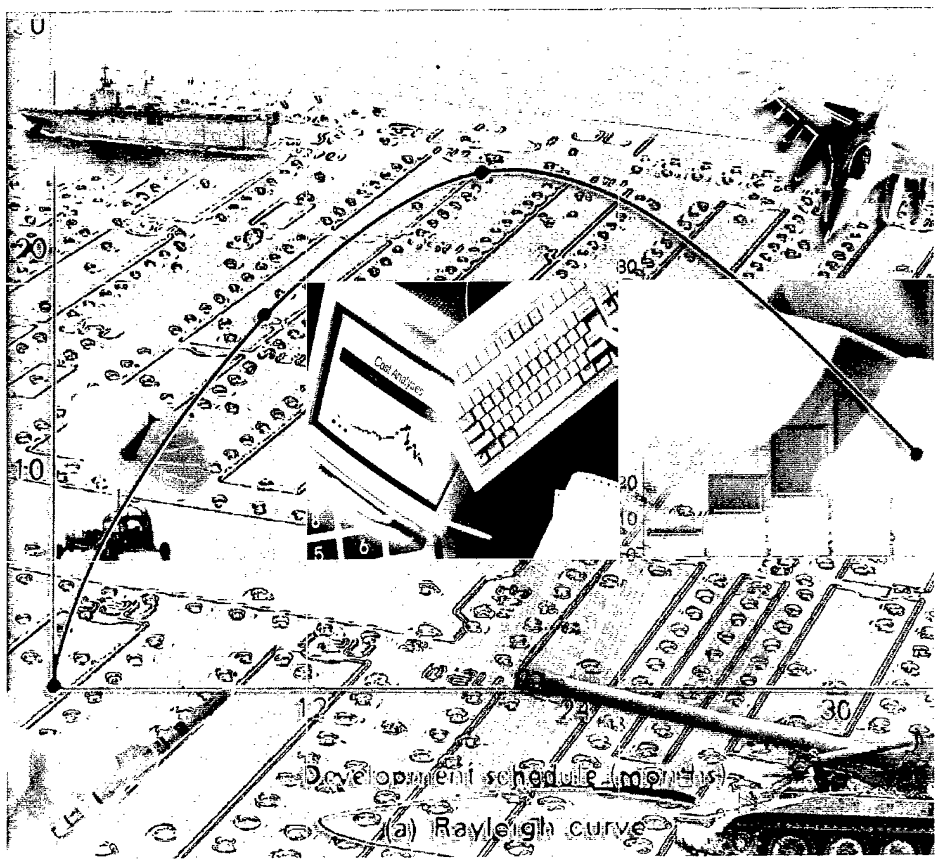
Logistics Management Institute

The Rayleigh Analyzer®

Volume I—Theory and Applications

AT902C1



John Dukovich
Scott Houser
David A. Lee

19991213 074

LMI®

Logistics Management Institute

The Rayleigh Analyzer®

Volume I—Theory and Applications

AT902C1

October 1999

John Dukovich
Scott Houser
David A. Lee

LOGISTICS MANAGEMENT INSTITUTE
2000 CORPORATE RIDGE
MCLEAN, VIRGINIA 22102-7805

The Rayleigh Analyzer®
Volume I—Theory and Applications
AT902C1/OCTOBER 1999

Executive Summary

Industrial engineers and cost analysts working on production programs have historically used mathematical models—such as cost progress (learning) curves—to predict factory production costs. Learning curves work quite well for *production* programs, but a very different approach is needed to estimate *development* programs.

When engineers analyze the cumulative expenditure profiles of many development programs, they commonly observe a "classic S-shape." The S-shape curve is the result of many project expenditures accelerating to a maximum rate, then tapering gradually until the project finally terminates. As far back as the 1960s, engineers have studied the Rayleigh statistical distribution as the model for development programs. The Rayleigh distribution models costs and schedules of major defense acquisition programs as well as all sizes of software projects.

The Department of Defense (DoD) instituted acquisition reform in 1994. However, industry literature shows that the Rayleigh model was associated with development programs started before 1990. A fundamental question is: Does the Rayleigh model estimate development program expenditures and schedules even after acquisition reform?

To find out how applicable the Rayleigh model is today, Logistics Management Institute (LMI) interviewed representatives of ten current or recently completed development programs, most of which were influenced by acquisition reform. The research team analyzed each of the development contracts using the *Rayleigh Analyzer®* and presented the results to the program representatives for evaluation and comment. Program representatives also were asked to identify major events and issues that affected the expenditure's for their development contracts, and the comment on the impact of acquisition reform.

The interviews demonstrated that *the Rayleigh model remains a valid tool for assessing performance of DoD development programs.* The consensus of the program representatives was that although acquisition reform is significantly beneficial, it did not fundamentally change the traditional S-shaped Rayleigh-like shape of development contract cost histories. The lack of change may be because the Rayleigh model primarily reflects how engineers form teams and solve problems, not how the acquisition process performs. The main factor to consider appears to be the flow of the engineering design and development process. In the programs we studied, the majority of deviations from the Rayleigh model resulted from changes in customer requirements or changes prompted by disruptive external events. Examples of external events are major hardware purchased in a short period of time, accounting adjustments, and government-funded development components (e.g., missile testing on government ranges).

In addition, some modern DoD software development programs have implemented new development processes, inspired by commerical best practices, that may not follow a Rayleigh distribution. Projects that have rolling requirements, recurring version releases, and continuously manned architecture, development, and testing teams may be better modeled as level-of-effort maintenance projects instead of traditional development projects.

As a result of many of the practical findings that ensued from our interviews with program managers, LMI has added some features to the *Rayleigh Analyzer*. The features enable analysts to model many of the perturbations that occur in modern development programs.

Contents

FIGURES

TABLES

Chapter 1
Overview

Background

For many decades, industrial engineers have used mathematical models to predict factory production costs. These models were based on the phenomenon of decreasing cost of production—as the quantity of items produced in a given plant increased, the unit cost of production generally decreased. These "cost progress" or "learning curve" models have been useful for estimating total manufacturing costs for an entire production plan, as well as for estimating lot-by-lot costs of a sequence of lots.[1]

Learning curves work quite well for *production* programs, but a very different approach is needed to estimate *development* programs. As far back as the 1960s, the "Norden-Rayleigh" (or simply "Rayleigh") statistical model has been studied as the development program model of choice.[2] The Rayleigh model estimates costs and schedules of major defense acquisition programs as well as those of all sizes of software projects.

To apply the theory to practice, Gallagher and Lee have implemented the Rayleigh distribution by using the multiple model adaptive estimation (MMAE) method to identify Rayleigh parameters.[3] To make the model useable by a wide range of analysts, the Logistics Management Institute (LMI) then developed a spreadsheet model, building on the work of Gallagher and Lee.[4]

However, after implementing the model in a useable format, a question arose: *Is the Rayleigh distribution valid for estimating development programs that managed under the U.S. Department of Defense's (DoD's) acquisition reform?*

To answer that question, LMI interviewed a cross section of program managers from acquisition reform-influenced programs. We checked the Rayleigh analysis for accuracy and its ability to capture and respond to the programmatic effects of acquisition reform and other programmatic perturbations.

---

[1] Lee, D., *The Cost Analyst's Companion*, December 1997.

[2] Norden, P. V., "Useful Tools for Project Management," *Operations Research in Research and Development*, B. V. Dean, Editor, John Wiley and Sons, 1963.

[3] Gallagher, M. A., and Lee, D. A., "Final-Cost Estimates for Research & Development Programs Conditioned on Realized Costs," *Military Operations Research*, Volume 2, Number 2, 1996, pp. 51–65.

[4] Lee, D.A., and Dukovich, J.A., *Using the Rayleigh Analyzer®*, LMI Report AT701C1, March 1998.

The result was somewhat surprising, yet encouraging. LMI found that the Rayleigh model remains valid for modern development programs. LMI also discovered the Rayleigh analysis can be innovatively modified to capture programmatic effects that the current model has not been able to do. This report describes our findings and discusses the application of these findings.

ORGANIZATION OF THIS REPORT

Volume I of this report, *Rayleigh Analyzer®—Theory and Applications*, has five chapters. This initial chapter introduces the reader to the purpose behind the *Rayleigh Analyzer* and our additional research of it. Chapter 2, Rayleigh Theory, provides the mathematics of the Rayleigh cumulative distribution and discusses its applicability to development program estimates. Chapter 3, Considerations for Practical Applications, discusses the results of our research of ten ongoing or recently completed development programs. We contacted each program office and offered them the chance to evaluate the *Rayleigh Analyzer*. Program office representatives also told us about the cost histories of one or more of their development contracts, and their views on acquisition reform. Chapter 4, Spline Theory, explains the use of spline smoothing in the *Rayleigh Analyzer,* devoting particular attention differentiating noisy cost report data. The fifth and final chapter, Summary and Recommendations, summarizes the major concepts in this volume, and the major findings of the program office research, and recommends direction of future analysis.

Volume II, the *Rayleigh Analyzer®—Users Manual*, is a guide for those using the *Rayleigh Analyzer* software tool. That volume has three chapters. Chapter 1 is an overview. Chapter 2 discusses procedures for installing the software. Chapter 3 has detailed instructions for operating the software.

Chapter 2
Rayleigh Theory

When describing the cost history of the typical engineering development's project, many speak of its cumulative expenditures as having a "classic S-shape." The term is used because the expenditures of many projects accelerate to a certain maximum rate, then taper gradually until the contract ends. Historically, the S-shape describes the cost history of items as different as software[5] and tanks.[6] The similarity of the cost histories for such a wide range of commodities under development implies an underlying commonality in all major development programs.

Peter Norden identifies this commonality as the relative staffing level of development programs over time.[7] Norden postulates that each development project starts with a core group of people that begins to outline the problems to be solved. As they solve the initial set of problems, they encounter more problems to solve. The team expands when the new problems can be efficiently assigned to new members. New problems are discussed and assigned to new people until all the major problems of the project have been identified. At this point, solving a problem does not create any new ones, and, therefore, the number of obstacles between the development team and success diminishes. The level of effort also diminishes until development is finished.

The Rayleigh statistical distribution does a good job of modeling the buildup, peak, and taper of the project's manpower over time. The general equation for the model is

$$R(t) = d(1 - e^{-\alpha t^2}) \qquad \text{[Eq. 2-1]}$$

where: t = time, d = the scale factor of the distribution, α = the shape parameter $R(t)$ = the total effort expended on the project. Many cost analysts have noted a correlation between the manpower buildup described by Norden and the earned value of DoD development contracts. (Earned value is a measurement of how much work was accomplished on a project compared to the work planned for that project.) In this case the Norden-Rayleigh model[8] becomes

---

[5] Putnam, L.H. and W. Myers, *Measures for Excellence—Reliable Software on Time, within Budget*, Prentice-Hall, 1992.

[6] Gallagher and Lee.

[7] Norden.

[8] For convenience and because it is the more common term, we will refer to the "Norden-Rayleigh" statistical model simply as the "Rayleigh" model from this point forward.

$$v(t) = d(1 - e^{-\alpha t^2}) \qquad \text{[Eq. 2-2]}$$

where: $v(t)$ = the earned value of the contract at a given time t.

This relationship implies that total effort expended is directly related to the actual cost of work performed (ACWP). In the context of earned value, this is equivalent to the time histories of development programs' ACWP, measured in constant dollars, being proportional to cumulative Rayleigh distributions.

The scale factor d is, therefore, measured in dollars. The scale factor is related to the estimated cost at completion of a development project by the relationship

$$D = v(t_f) = 0.97d \qquad \text{[Eq. 2-3]}$$

In equation 2-3, t_f is the estimated time at completion of the project. The estimated time at completion is related to the shape parameter α by the relationship

$$\alpha = \frac{3.5}{t_f^2} \qquad \text{[Eq. 2-4]}$$

where 3.5 is an approximation for ln(0.03). The shape parameter α also determines the time of peak expenditure rate, t_p:

$$\alpha = \frac{0.5}{t_p^2} \qquad \text{[Eq. 2-5]}$$

The implied relationship between manpower and the total effort expended relies on several assumptions, the most notable being these:

- The problems to be solved in any development project are finite, yet not known at the outset.
- The number of people employed in solving problems is proportional to the number of problems ready to be solved. [9]

The second assumption implies that problems must be identified, investigated, and defined before additional personnel can solve them. This is why it is usually infeasible to infeasible to compress the timeframe of a development contract simply by adding more people. Additional personnel are underused until existing personnel can present them with a problem to solve.

The implied relationship between the ACWP and total effort requires additional assumptions.

[9] Norden.

- The majority of costs charged must be directly related to the time spent by the development team when solving problems. In other words, hardware purchases, overhead adjustments, and other costs that are independent of the development team's billable hours must be small in comparison to the total cost of the contract.

- The contract must fund a single, integrated development project in its entirety.[10]

The first assumption means that the main cost of a development project must be that of engineers solving problems. Administrative costs, material costs, and other items not related to developers working do not fit under a Rayleigh model. Costs directly related to human-oriented problem solving, such as electrical and equipment costs, do fit in the model. The second assumption is that each development project must conceptually unify in both engineering and accounting. If the project is arbitrarily divided into two or more contracts, then the Rayleigh curve will not track the accumulated costs of each of the contracts.

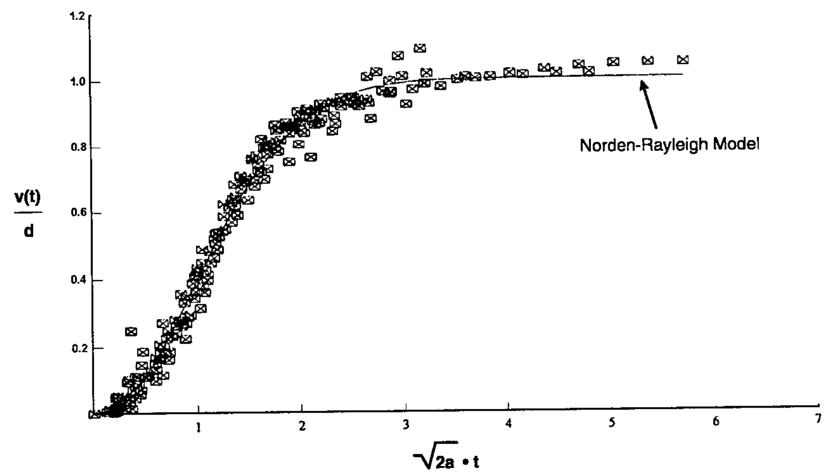
Conversely, if one contract is used for essentially independent items, such as an airframe and an electronics system, then a single Rayleigh model will fail to track the accumulated cost of the contract. This does not mean, however, that an airframe contract that includes avionics will not follow a Rayleigh distribution. In fact, the integration of the avionics into the airframe ensures that the overall contract expenditure history will follow a Rayleigh distribution.

[10] Putnam.

Chapter 3
Considerations for Practical Applications

As we discussed earlier, the Rayleigh model accurately models the expenditures and schedules of development programs. For example, Lee, Hogue, and Hoffman[11] showed that the Rayleigh model did a very nice job of collapsing data from many defense programs onto a single curve see Figure 3-1.

Figure 3-1. Profiles of Development Program Cost vs. the Rayleigh Model



However, the programs as the basis for Figure 3-1 all began before 1990. A fundamental question arises: Is the Rayleigh model still a good estimator of development program expenditures and schedules after *acquisition reform* (a major initiative mandated for DoD in 1994[12])? The goals of acquisition reform are to acquisition more effective, efficient, and productive. Acquisition reform includes reducing overhead, streamlining requirements, reengineering processes, and reducing administrative requirements. Acquisition reform strives to increase the use of commercial practices, such as using performance and commercial specifications. In addition, the use of private enterprise to do more of the functions traditionally done by government is encouraged.

To find out how applicable Rayleigh model is for modeling current programs, LMI interviewed representatives of ten current or recently completed development programs, most of which were influenced by acquisition reform. The re-

[11] Lee, D.A., M.R. Hogue, and D.C. Hoffman, "Time Histories of Expenditures for Defense Acquisition Programs in the Development Phase—Norden-Rayleigh and Other Models," presented at the 1993 Annual Meeting of the International Society of Parametric Analysis.

[12] DoD Secretary William Perry memorandum "Specifications and Standards—A New Way of Doing Business," June 29, 1994.

search team analyzed each of the development contracts using the *Rayleigh Analyzer* and presented the results to the program representatives for their evaluation and comment. Program representatives also were asked to identify any major events and issues that affected the expenditure history of their development contracts, and to comment in general on the impact of acquisition reform.

The interviews demonstrated, with a few caveats, that *the Rayleigh model remains a valid tool for assessing the performance assessment of DoD development contracts*. Most of the caveats that would preclude using the Rayleigh model are not related to acquisition reform or other modern development considerations and, therefore, apply equally to modern and pre-acquisition reform programs. In general, acquisition reform does not affect the applicability of the Rayleigh model because it does not fundamentally change the *engineering* processes upon which the model is based. Rather, the reform focuses on reducing *administrative* and *bureaucratic* aspects of programs. Should a paradigm shift occur in the engineering processes, then acquisition processes that make the change possible will certainly affect the validity of the model. However, acquisition reform in its present state has either no effect on engineering development, or enhances the development by reducing the possibility of acquisition issues interfering with engineering. Therefore, if anything, the overall effect of acquisition reform may be to increase, rather than decrease, the accuracy of the model.

Our research identified several other issues that may affect the applicability or effectiveness of the Rayleigh model. Modifying the Rayleigh model, for example by combining two or more Rayleigh curves into a single composite curve, can address many programmatic or contractual issues. Other issues may be addressed by accounting for special circumstances, such as accounting adjustments, one-time hardware purchases, and outside funding such as that used for government-funded testing, that affect how the history expenditure is recorded.

The Rayleigh model does not apply to some situations. Most of the situations are contractual arrangements that break with the assumption that each contract is funding one and only one development program, and that the program effort is independent of those under other contracts. Modern software development programs are another exception. Under certain circumstances, the Rayleigh model does not apply to pure software development (see the subsection titled "The Software Exception" on page 3-5). Many software cost estimation tools currently available are better for projecting costs of developing software than the Rayleigh model. This exception does not apply to the development of one-of-a-kind software or integrated hardware and software; rather, it applies to the development programs that release periodic versions and upgrades, much in the way that a commercial software vendor might.

DESCRIPTION OF PROGRAMS STUDIED

Working with the Office of the Secretary of Defense (OSD), we strove to consider programs that covered a cross section of services and systems. The ten selected programs cover each branch of service and provide a range of commodities and program sizes (in terms of cost). The development programs we selected were either ongoing or recently finished, and the bulk of their development contracts were completed in the acquisition reform era (defined loosely as 1994 or later). Table 3-1 lists the programs that were covered by our interviews. In the table, we use generic names for the programs and contracts because the cost information is proprietary.

The ten programs in our sample provided us with data for fourteen contracts, because some programs used more than one contract. Of the fourteen contracts shown in Table 3-1, only ten (and possibly only nine) fit the assumptions that enable using a Rayleigh estimate at their completion. The second contract of the first missile program is undergoing a cost-as-an-independent-variable (CAIV) restructuring that has essentially halted development for the past year. The contract continues to charge for work performed, but the work is a trade study, not system development. The land-based mobile weapons system contract is for initial research and development (R&D) only, not full-scale development. The Rayleigh model has only been proven effective for contracts for full-scale development (FSD) or engineering and manufacturing development (E&MD). However, this land-based mobile weapons system has adopted a new development strategy that more tightly couples its initial development and FSD, raising the possibility of tracking the entire multi-contract effort by using a single Rayleigh model. Another contract is for the simultaneous design, development, and manufacture of a very small number of Navy vessels. Although this process is unique to complex Navy vessels, it does not necessarily conflict with the assumptions of a Rayleigh analysis. However, the contract ends arbitrarily about three years before the first-unit is developed. Although modeling the entire development with a Rayleigh model may be valid, the model cannot be used to estimate the completion of the contract. Finally, the software development contract should not be analyzed with a Rayleigh model because its continued releases of new versions violate the assumption of working on finite problems.

When possible, the model's estimated final cost and completion time were compared with the actual final cost and completion time. Active programs have no actual final cost or completion time, so the program manager's estimate was used as a basis for comparison. However, program managers' estimates care based on many qualitative, as well as quantitative factors, so conclusions made from these data must be carefully regarded. For example, a difference of greater than 30 percent between the two estimates may indicate that the Rayleigh model is invalid for the program, the program office estimate is wrong, or some unknown acquisition issue is affecting the analysis.

Table 3-1. Percent Differences between Rayleigh and Program Manager Estimates

Program	Contract	Percent difference / final ACWP ($millions)	Percent difference / completion time (years)
1	Airframe I	–3.9* /–1.9	–11.3 /–10.5
2	Airframe II	4.6 / 10.7	1.0 / 8.4
	Aircraft Engine I	0.2 / 10.2	–9.8 / 4.3
3	Missile I (Contract A)**	1.4 / 3.2	–5.7 / 11.7
	Missile I (Contract A) *Two-Rayleigh Model*	2.0 (point estimate only)	-7.3 (point estimate only)
	Missile I (Contract B)	N/A	N/A
4	Missile II***	–33.5 /–25.3	5 / 23.8
	Missile II–Constrained to 5.0 Years	3.8 / 8.1	N/A
5	Naval Electronics I**	–1.4 / 5.2	–5.4 / 13.4
	Naval Electronics I – *Two-Rayleigh Model*	4.8 (point estimate only)	6.3 (point estimate only)
6	Land Vehicle Electronics I	–0.7 / 1.7	–8.1 / 1.4
	Land Vehicle Electronics II	1.4 / 3.8	11.2 / 34.6
7	Land-based Mobile Weapons System	N/A (Within 20% of combined total for all R & D contracts)	N/A
8	Naval Vessel I	N/A	N/A
9	Missile Electronics I	0.3 /9.2	–1.0 / 10.5
10	Software I	–13.4 / 9.8	–6.8 / 14.7

* The first number is a comparison between the program office's estimate and the Rayleigh Analyzer's estimate at 50% confidence. The second number uses the 90% confidence estimate. The two numbers give a range.

** These contracts confirmed a major expansion in scope that would lead to a two-Rayleigh model. The best estimate of the two-Rayleigh model are given in the entry immediately following.

*** This contract had too few data points to form the basis for an accurate Rayleigh estimate. It was further investigated to learn more about the effect of acquisition reform on recent contracts.

The fact that the Rayleigh model did not closely fit the data for some of our sample development contracts is partially because of the samples used. One reason we talked with the program representatives was to discover weaknesses in the model that may not have been known to exist. Therefore, because of time and resources for interviewing were limited, the research team emphasized *unusual*, potentially non-Rayleigh expenditure patterns. In the four cases listed above, the common thread was not odd expenditure patterns; rather, it was in the way the contracts were structured. One contract funds a project that has suspended development activities for almost a year, while the others fund only part of a single development effort. This observation leads to an important conclusion: of this research: *Merely knowing that a contract funds a development project is not enough for correctly applying the model. The user must also know whether the contract funds all or part of the project, and whether any major events have occurred that have halted or impeded the normal progress of the project.*

APPLICABILITY OF THE RAYLEIGH MODEL

Table 3-1 readily identifies one general type of project for which the Rayleigh model cannot be applied: that in which the contract does not correspond to a single, complete development project. In terms of commodities, the model may be ineffective for modeling modern software development projects. This is because mature software projects—although called development projects as a matter of form—may become more of a maintenance project in practice as developers create new versions and patches periodically. Our research did not identify other major situations in which the Rayleigh model cannot be applied, but our work found several factors that may adversely affect its results. The remainder of this section discusses those factors, how to recognize them, and what to do to minimize or remove their impact on a cost analysis.

The Software Exception

Our research revealed a change in software development that may restrict the Rayleigh model's effectiveness for analyzing modern software processes. The Rayleigh model has been historically successful in modeling software development,[13] so the possibility that the model may no longer apply to certain types of software development projects is a new and potentially controversial idea. However, this idea is supported by both our interview with a program manager from a DoD software development project and modern software metrics tools.

Our lone software project is a program that supplies intelligence fusion to Army theatre commanders. The developers produce a new version approximately every 12 to 18 months. Planning time for each new version typically lasts six months. As a result, no single product is being developed from concept to tested prototype as we would expect in a traditional development project. Because the development is focused on enhancing the current software and correcting defects, the program manager we spoke to claims that there is no ramp-up period that we would otherwise expect. In additional, the design, development, and testing teams are never disbanded; they simply work on different versions at varying

[13] Putnam (1992).

stages of their development. For example, the designers may be planning to build version five while developers are building version four.

In this case, the software project resembles more of a maintenance or production effort than traditional development, and could not, therefore, be modeled accurately with a Rayleigh model. The contract structure of the program also is a major issue. Because the development team produces new versions continually, the program manager can set relatively arbitrary starting and stopping points for the contracts that fund the development. According to the program manager we interviewed, the ten-year length of the contract was set for cosmetic purposes: "No one likes to see a software contract go on forever." When the contract ends, we were told, a new one will take its place. The program manager called this process of continuous, assembly line production of new software versions "spiral development," and pointed out that it is a prevailing model for commercial developers such as Microsoft.

According to our findings, modern software projects apparently pose two problems for an analyst using a Rayleigh model:

- The project may result in a level-of-effort expenditure profile rather than the gradual ramp-up and close-out pattern typical of DoD development projects; and
- With this spending profile established, start and end points become arbitrary, so program managers may be more likely to parcel funding to the project in several contracts, rather than just one.

These issues only apply to multiple-version, mature software, not combined hardware software products or single-version software.

Our research shows that the Rayleigh model still applies to most software-intensive projects, particularly those that involve hardware development. This is good because almost all modern electronics development projects require dedicated software. Relatively few DoD software projects fit the spiral development profile detailed above, but options are available for obtaining estimates for the few that do. Several software metrics tools predict time or cost at completion based on the basis of a number of parameters available from the development team.

The models are not as simple to use as the *Rayleigh Analyzer*, but may estimate more accurately. Enlisting the help of a program office point of contact should be the first step in determining if a software project requires an alternative cost estimation tool. A program representative should be able to say whether the contract has a functional or arbitrary end point, whether the software is on a single- or continuous-release cycle, and whether it is independent of hardware development.

Multiple Contracts

The previous discussion showed that multiple contracts applied to a single development effort violates the assumption of a Rayleigh model. However, many caveats to this rule must be discussed. First of all, development projects often have an investigative research or initial design phase funded by one contract, followed by FSD or E&MD funded by another. If the work performed under the initial contract is relatively extensive, is applying a Rayleigh analysis to the second contract still valid? What about modeling both contracts as a single project by combining the cost reports from each? Historically, the answer to the first question is *yes*: even though the initial work may have been substantial, the FSD team must still fully form and collect its resources to build on that work. That means that the FSD/E&MD contract must go through its own ramp-up period, and will, therefore, take on a Rayleigh expenditure profile. The answer to the second question is historically *no*. The initial contract may not have a Rayleigh expenditure profile at all; often the contract involves linear expenditures. In addition, because of time lapses between the two contracts, the two contracts will behave independently, even if the first contract is sufficiently substantial to show a Rayleigh expenditure profile.

Some evidence exists that modeling a project structured in this way is possible with a Rayleigh model. For example, the land-vehicle program we investigated has a relatively extensive preliminary design contract followed by an E&MD contract. The program office is implementing a contract structure called "single objective development." This is a relatively new idea that requires overlapping the end of the preliminary design phase with the start of the E&MD phase to minimize the dip in effort that normally takes place between the two. If successful, the development team will minimize the scaling back and subsequent rebuild between the peak activity times of both contracts, making the two contracts appear almost seamless. Further study of similarly structured programs may show that they can be modeled using one or more Rayleigh curves.

In addition to programs that use more than one contract for a development project, there are those that are forced to use two or more separate contracts because of outside intervention, such as a congressional cancellation and reinstatement. In the case of outside intervention, modeling the second contract with a Rayleigh model is usually valid because the intervention most often forces the development team to re-establish itself. This is particularly true for this situation as opposed to a planned halt to development because the contractor's management has no reason to believe that the team will be working on the project in the future. Therefore, unless removing the first contract and establishing a second one takes a relatively insignificant amount of time, the second contract will have a Rayleigh-like expenditure pattern that is independent of the first contract.

Volatility of Requirements

Volatility of requirements were a recurring issue in our discussions with program office personnel. The software development program, the most extreme case, reported three complete turnovers of its requirements. The first airframe contract incorporated a new variant that added 20 percent to its final estimated cost. The first missile program also incorporated a new variant into its contract that added 80 to 100 percent to its value, and the Naval electronics program had an expansion in scope that increased the value of its contract by about 100 percent.

The effects of expanding requirements vary. To assess the effects, we fit the cost data of each contract using several formulations of cubic splines (discussed in the next chapter), and computed the derivatives of the spline curves to show the expenditure rate over time for each project. While the derivative of a Rayleigh curve is a simple curve with one inflection point, the spline-based expenditure rates often showed two distinct peaks corresponding to the time in which a new variant was added, a major set of new requirements was introduced, or some other major event occurred. The representatives of programs exhibiting these extra peaks confirmed them, leading to our conclusion that the extra peaks are commonplace. Our comparison with the expenditure profiles of older contracts also shows the extra peaks to be a historical phenomenon, rather than an exclusively recent one.

As a result of our findings, we developed a *two-Rayleigh composite model* that fits the composite of two separate curves to a given project's cost data. We also obtained detailed monthly data for the first missile contract and the Naval electronics contract in order to model them more closely. We discovered that the two-Rayleigh model could approximate the expenditure rate curves of these projects reasonably well, and could fit the cost curves well, even though the curve-fitting algorithm used in the two-Rayleigh model is inferior to the MMAE algorithm. An MMAE-based single-Rayleigh result shows slightly less accurate results, despite the better algorithm. We can conclude from the results that the MMAE implementation of the Rayleigh model can accurately estimate of final cost and completion time even if the requirements of a program expand extensively. Also we can conclude, however, that an MMAE implementation of the two-Rayleigh model will perform even better in these circumstances.

Government-Funded Testing of Missiles

Representatives from both missile programs said that government-funded testing comprised a significant part of their projects. Because such testing is not funded with contract money, it may appear in these programs may appear to cost little during the test period, which may last many months. As a result, the Rayleigh model may underpredict the amount of time required to complete the contract. However, we do not expect that this testing will cause an error with estimating cost because the development group will still taper their activities before beginning testing, almost as if they were completing the project. In addition, if the time interval between each cost report is sufficiently long, the error introduced by government-funded testing probably will not be high compared to the error introduced by the sparseness of the data.

Our review of the historical data showed that the development of missiles may have long periods of little activity at the end of the contracts. This may be caused by the government expending most of the funds for testing. However, the accuracy of the final cost and completion time for the projects seem about as expected. The number of cost reports that show this behavior is still relatively small compared to the number of cost reports in the project as a whole. In Addition, because the Rayleigh model predicts an expenditure history that converges asymptotically to some value, any government-funded activity near the end of these contracts seems to affect the model's predictions very little.

Hardware-Intensive Development

The development of many defense-related commodities involves building several prototypes for testing and redesign. Often the prototypes require a one-time purchase of hardware at some point during the development contract. These purchases may lead to a large departure from the expected spending patterns of the project for one or more recording periods. Using a Rayleigh analysis on a project that has recently made a large purchase of hardware can hinder the accuracy of the model. Therefore, the analyst should be sensitive to the possibility of such purchases when examining the cost data of a development project. If the user suspects that such a purchase is skewing the analysis of a given project, he or she should determine from the program office how much was spent in one-time hardware purchases, and when those purchases occurred.

Program representatives have said that aircraft engine development often follows this pattern because engines are relatively complex systems with a large number of vendor-supplied parts. The engine contract listed in the table at the beginning of this chapter shows higher than expected expenditure rates for the first five cost reports. The program office attributes these higher costs to the purchase of parts for the prototype engines. When modeling the engine development project with a two-Rayleigh composite model, the region of higher-than-expected spending rates produces a Rayleigh curve with a magnitude of about 15 percent that of the total estimated cost of the contract. In this case, the effect of the up-front purchases are minimal: the single Rayleigh and two-Rayleigh models differ by slightly less than 5 percent with each other, while the two program office estimates for this contract also differ by slightly less than 5 percent.

It is expected that a one-time purchase would hurt the accuracy of the model most if it occur during the time of peak expenditure rate in the project. Therefore, we ran an experiment to determine the effect of adding a 4 percent expenditure, spread over three quarters, on a project expected to last approximately ten years. We created a model project by computing a Rayleigh curve with a magnitude

$1 million and a peak expenditure rate occurring 3.0 years from the start of work. The team then added a $40K Rayleigh curve to this data to simulate a one-time, phased in purchase of equipment or hardware (for a total program of $1,04 million). The peak expenditure occurred 0.25 years after its start, which was 2.75 years into the project. The curve converges to its full magnitude at the 3.5 year point.

The composite data were loaded into the two-Rayleigh sheet, which computed the best-fit single Rayleigh curve for the simulated project at various points in the project's life. (The two-Rayleigh sheet can compute a single Rayleigh curve fit when the second curve's magnitude is set to zero.) The curve fit started with an initial guess of 3 years for the time of peak expenditure and $2 million for the magnitude of the curve. The results were much better when more post-expenditure cost reports were considered in the curve fit, as shown in Table 3-2.

Table 3-2. Best-Fit Single Rayleigh Curve of a Program with a One-Time Expenditure (Computed Using Two-Rayleigh Spreadsheet)

ACWP reports beyond the expenditure	Computed magnitude of curve (theoretical value $1.04 million) (thousands of dollars)	Percent error (%)
2 (1/2 year)	1,676	61.1
6 (1.5 year)	1,404	35.0
10	1,094	5.1
14	1,047	0.7
18	1,039	–0.1
22	1,038	–0.2

When the $40K expenditure was removed from the data, the optimizer in the two-Rayleigh spreadsheet matched the data with a less than 0.01 percent error, as expected. This test removes the possibility that the optimizer was not nimble enough to converge to the right solution given the initial guess selected. We can see from the data that the error becomes manageable in a reasonable amount of time, even for this contrived example: at 2.5 years beyond the expenditure, the magnitude of the fit curve and the theoretical curve are within 5 percent. However, in the time immediately following the expenditure, the curve fit is thrown off by the temporarily accelerated expenditure rates, producing large errors.

Because this test uses a curve-fitting routine that provides a single point estimate, the effects of one-time expenditures on the accuracy of the Rayleigh model are easier to show. However, the real question is whether the same effect occurs in the *Rayleigh Analyzer's* MMAE routine. MMAE is a Bayesian-state model, while the built-in optimizer in the two-Rayleigh spreadsheet is a simple, deterministic linear programming tool. Therefore, we expect that the MMAE routine will show a range of answers that are more reliable than the linear programming routine. Table 3-3 lists the percentage of difference between the theoretical final cost of the project and that predicted by the *Rayleigh Analyzer* at 50-percent and 90-percent confidence values. The time intervals listed are identical as those for the table above. Because constant dollars are converted to then-year dollars, the magnitudes listed here are slightly different from those above. However, the underlying data remain the same.

Table 3-3. MMAE/Single Rayleigh Modeling of a Program with a Simulated Hardware Purchase

ACWP reports beyond the expenditure	Computed magnitude of curve ($K) (theoretical value $994K) (50% / 90% confidence)	Percent difference between the data point and the theoretical value (%)
2 (1/2 year)	978 /1,305	–1.6 / 31.3
6 (1.5 year)	849 /942	–14.6 / –5.2
10	928 /993	–6.6 / –0.1
14	967 /989	–2.7 / –0.6
18	986 /1,002	–0.8 / 0.8
22	993 /1,013	–0.1 / 1.9

The errors are relatively small compared to those found using the simple optimizer. Because the MMAE provides a range of values rather than a single-point estimate, relating the errors shown above to the accuracy of the model is more difficult; one also must consider the range of values in making this judgement. Clearly, the model shows a much wider range for the first two data points, indicating significant uncertainty in the estimate. However, the data are overall more reliable than those obtained through a linear programming technique. The researchers computed a similar set of results from the same data set with the simulated expenditure removed. Because this data set was computed from a single Rayleigh curve, the results are expected to show a tight band around the theoretical endpoint. The results meet these expectations: a difference of no greater than 3 percent exists in the 50 to 90 percent confidence range for any of the data points above.

Accounting Adjustments

Several of the program personnel we spoke to mentioned accounting practices as a possible source of error in a Rayleigh analysis. Specifically, adjusting the general and administrative (G&A) or overhead rates on a contract can radically affect the recorded cost for the period in which the adjustments were made. One program representative went so far as to say that the adjustments can make it seem like the program office is making money, rather than spending it, during those reporting periods. The single-point discrepancies between the actual and accounted expenditure rates can introduce a large uncertainty in the final estimate, particularly if the adjustment is made during a recent cost reporting period.

Table 3-4 displays the potential severity of such a discrepancy by inserting a single-point adjustment during the peak expenditure rate of a curve that is expected to converge at approximately ten years.

Table 3-4. Comparison of the MMAE/Single Rayleigh Modeling of a Perfect Rayleigh Expenditure Profile and a Rayleigh Expenditure Profile with an Accounting Adjustment

Number of ACWP reports beyond period of peak expenditure rate	No adjustment made ($K) (theoretical value $600K) (50% conf/ 90% confidence)	Adjustment made at time of peak expenditure rate ($K) (theoretical value $600K) (50% conf/ 90% confidence)
0	599 / 608	320 / 750
4 (1 Year)	585 / 586	622 / 994
8	610 / 610	545 / 635
12	590 / 603	575 / 607
16	593 / 605	605 / 611
20	615 / 615	625 / 640

As expected, the *Rayleigh Analyzer* fits a tight range around the pure Rayleigh data (second column): the largest discrepancy at any point in the 50 to 90 percent confidence range is 2.5 percent. The accounting adjustment caused considerable uncertainty in runs containing few or no data points after the period in which the adjustment was made. When no data points are provided after the adjustment, the maximum discrepancy between the theoretical endpoint of the curve within the 50 to 90 percent confidence band approaches 50 percent. The magnitude of this band is greater than two-thirds the total size of the curve, which is further evidence of extreme uncertainty in the estimate. This uncertainty remains pronounced when a year (four quarterly data points) of additional data is provided, and is still noticeable when two years of data are provided. Beyond two years, however, the effect of the adjustment becomes relatively insignificant.

Because this contrived example is near-worst case, we can safely conclude that a Rayleigh analysis will be reasonably accurate on data that has an accounting adjustment, if at least one, and preferably two, years of quarterly data are provided after the adjustment is made.

Alternative Rayleigh-Based Models

Many of the program offices we investigated said the original requirements expanded considerably during the contracts. The first airframe contract includes a new variant that added approximately 20 percent to the total expected cost of the contract midway through the development. The first missile program also included a new variant that added about 100 percent to the expected cost. The naval electronics program also saw a 100 percent midstream increase in cost because the original requirements expanded.

Almost certainly these events have happened before to many DoD development programs, so we expect that a single Rayleigh model should still suffice for these programs. However, possibly one of three models will be more accurate:

- Two-Rayleigh model
- Three-Rayleigh model
- Modified Rayleigh model.

The *two-Rayleigh model* addresses the scenario in which a project must address a new set of requirements, in midstream, that are sufficiently substantial to warrant essentially independent development to meet the requirements and integrate them into the existing system. The two-Rayleigh model is a composite of two Rayleigh curves, fitted to the data: one starts at the start of work, while the other starts at some offset time corresponding to the start of work to meet the new requirements.

The *three-Rayleigh model*, follows the theory that each development project has three distinct stages, such as conceptual design, detailed design, prototype/test. These three stages of development should be able to be modeled as single Rayleigh curves, then combined to form a composite curve that fits the data better than a single Rayleigh curve. Because some program offices support for this theory, the research team constructed the three-Rayleigh model to test it. The three-Rayleigh model also can be used for some umbrella contracts, and contracts with two distinct restructuring or rescoping events.

A *modified Rayleigh model* addresses the scenario in which the new requirements force the team to accelerate their efforts, but allows them to build almost entirely on their existing work to meet the new requirements. In essence, the spending patterns of these projects shift from a shallower Rayleigh curve to a steeper one in response to the new requirements. The composite model, therefore, follows an initial Rayleigh curve to some intersection point with another, steeper Rayleigh curve. Then the model follows the second curve for the remainder of the contract. This method works whether the change adds or removes contract scope, and whether it compresses or expands the schedule.

Two-Rayleigh Composite Model

When a project funded by a contact changes significantly in scope, which that drives a major change in the projected total cost (following an initial peak expenditure rate), the resulting spending history will be distinctly bimodal. By major, we mean at least a 50 percent increase in accumulated cost at completion. Experiments have shown that contracts with smaller increases in projected cost still follow a single Rayleigh model rather well. In fact, our experiments show that even expansions resulting in an increase of 100 percent to the projected total cost can be modeled reasonably well by a single Rayleigh curve, if MMAE or a similar parameter identification algorithm is used.

Transitioning from a single Rayleigh model to a two-Rayleigh model is like transitioning from a lower order curve to a higher order curve for a given data set. The higher order means greater degrees of freedom, which translates to more flexibility to fit the data. However, the additional degrees of freedom also can introduce extraneous features in the curve fit.

The same principle applies to a two-Rayleigh model: the two distinct curves give the model greater flexibility to fit a given set of cost data, which can translate to a better estimate than that provided by a single Rayleigh model. However, this additional flexibility also can create considerable errors in the estimate, particularly if the project is distinctly bimodal, but hasn't reached the second peak yet. The cause of this error is the same as that which could cause severe inaccuracy in a single Rayleigh analysis of an insufficient data set: the model needs to identify when the peak expenditure rate occurs in order to properly determine the parameters of the curve. Our observations lead to the following precautions:

- Always perform a single Rayleigh analysis in tandem with a two-Rayleigh analysis. A major discrepancy between the two may indicate a false fit in the two-Rayleigh model.
- Always verify that each of the individual curves in the two-Rayleigh model make physical sense.

For example, if a program receives a new set of requirements 2.5 years after the start of work that correspond to a total cost increase of 100 percent, then both curves in the model should have roughly the same magnitude. The first curve should also start about 2.5 years into the contract. Figure 3-2 shows exactly such a case: it is a graph of the best two-Rayleigh curve fit of the first missile program listed in Table 3-1.

Figure 3-2. Two-Rayleigh Extrapolation of a Missile Program

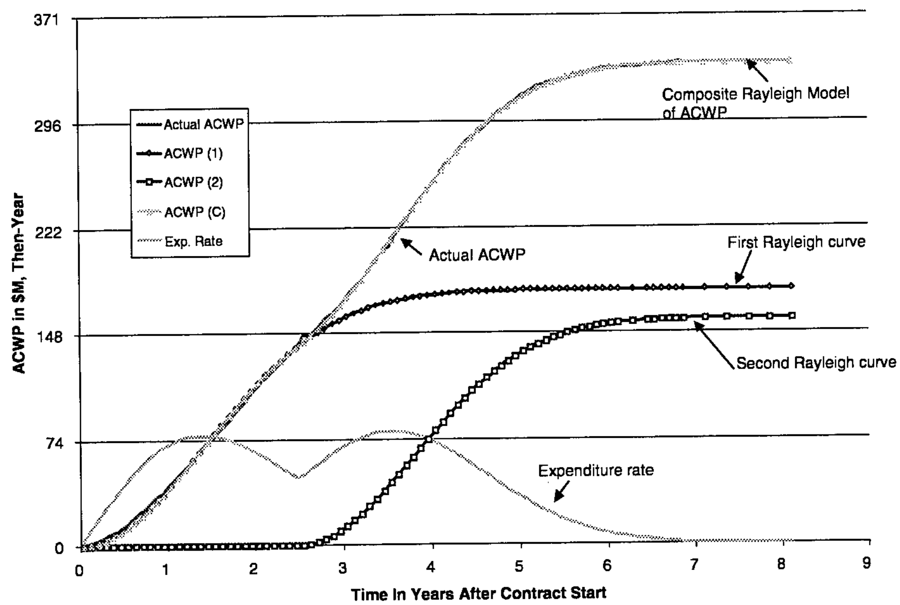
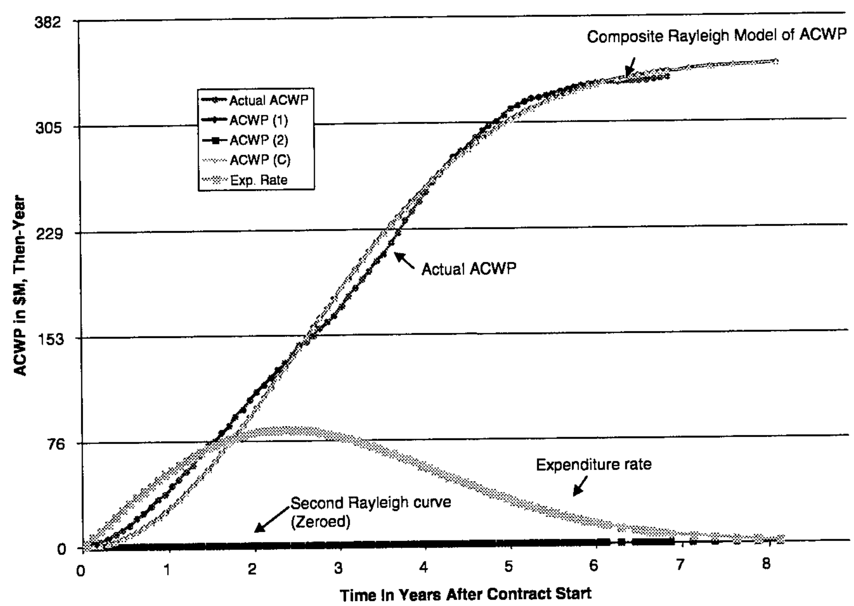


Figure 3-3 shows the same program as fit by a single Rayleigh curve. More precisely, the two-Rayleigh spreadsheet in the *Rayleigh Analyzer* generated this curve fit when the magnitude of the second curve was set to zero. The graph shown in Figure 3-2 and Figure 3-3 lead to two observations. The first observation is that the estimate-at-completion of both models agree quite well; the more accurate curve fit of the two-Rayleigh model does not necessarily provide a more accurate estimate. Second, the single Rayleigh fit below was generated by "tricking" the two-Rayleigh model into believing that the second curve has a zero magnitude.

Obviously, because the same model can be used to generate two such obviously different solutions, it may generate other optimal solutions that do not accurately reflect contract numbers. The MMAE algorithm generally will find a much better estimate than a simple single Rayleigh optimization, and, in most cases will provide as good an estimate at completion as the two-Rayleigh model, even when the project contains two distinct expenditure rate peaks. Therefore, the user should use of the two-Rayleigh model only for contracts in which the accuracy of the MMAE estimate is questionable. The user should always know the rough magnitude and duration of the two phases of a specific contract to reduce the likelihood of inaccuracy in the two-Rayleigh model. If the user does not have this information but wants to use a two-Rayleigh model, it is best that he generates a spline fit of the data and its derivative (expenditure rate). If the expenditure rates generated by the spline routine and the two-Rayleigh model match reasonably closely over time, then some confidence can be taken in the two-Rayleigh estimate at completion.

Figure 3-3. The Same Missile Program Shown in Figure 3-2, Extrapolated with a Single Rayleigh Curve



MODIFIED RAYLEIGH COMPOSITE MODEL

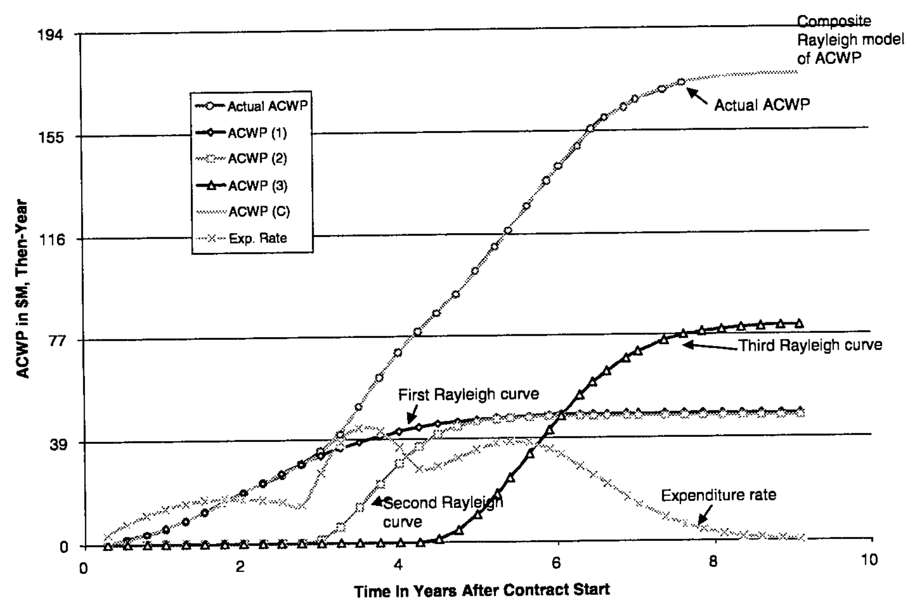
Using the modified Rayleigh model requires computing the maximum value of a Rayleigh curve that starts at $t = 0$ and ends at some completion time t_f and a Rayleigh curve with a larger magnitude that starts at some offset time and ends at the same t_f. The assumption is that the development team responds to an expanded set of requirements by increasing their level of effort without a traditional ramp-up period. Because most major expansions in requirements need some ramp-up time, this model is less applicable than the two-Rayleigh model. Although the qualitative information we received during the program office interviews suggest that the model has merit, we were unable to find a project that fit a modified Rayleigh profile as well as it fit either a single or two-Rayleigh profile.

THREE-RAYLEIGH COMPOSITE MODEL

The three-Rayleigh composite model sums three distinct Rayleigh curves to create a composite curve that approximates a development project. This model is based on the assumption that a project has three distinct phases, such as conceptual design, detailed design, prototype, and test. The idea for this particular model comes from the perception that development programs behave more like a sum of several Rayleigh curves than a single Rayleigh curve. An arbitrary number of Rayleigh curves can be used to *fit* the accumulated cost data of development project at least as well as a single curve. A better fit usually leads to a better *extrapolation* of the data, which is needed for a better estimate at completion. However, as the discussion of the two-Rayleigh composite model states, the greater responsiveness to the data of the extra Rayleigh curves can, in certain cases, lead to a less accurate extrapolation. The extra Rayleigh curves may be more sensitive to perturbations in the most recent cost reports. Because the three-Rayleigh model is based on physical phenomena that have been confirmed during our program office interviews, the potential for enhanced accuracy exceeds the risk of introducing artifacts in the curve fit.

Figure 3-4 shows a three-Rayleigh model of a missile program. The three-Rayleigh model predicts a final cost within 5 percent of the actual final cost, while a single-Rayleigh model estimate differs from the actual cost by 42 percent. The MMAE algorithm predicts the final cost to within 0.3 percent at 50-percent confidence, and to within 9 percent at 90-percent confidence. Clearly, the MMAE model and the three-Rayleigh model produce roughly the same quality of estimate, while the simple one-Rayleigh model produces a far inferior estimate. This is no doubt due to the superiority of the MMAE algorithm to the simple linear programming algorithm used to generate the one- and three-Rayleigh fits. If the three-Rayleigh model is incorporated into an MMAE algorithm, that accuracy is expected to greatly improve.

Figure 3-4. Three-Rayleigh Extrapolation of a Missile Program



The user should be aware that the three-Rayleigh model is particularly sensitive to perturbations in the data set because of its additional degrees of freedom. Therefore, the user should have a good idea of the size and location of each Rayleigh curve that makes up the three-Rayleigh composite. This will aid the user in verifying the quality of the fit. If the user is not certain about the exact size of each phase but still wishes to use this model, it is best to obtain a *spline fit*[14] of the cost history data and its derivative (the expenditure rate). If the model's expenditure rate matches the spline-generated expenditure rate reasonably well over time, then confidence can be taken in the model's estimate at completion.

THE EFFECTS OF ACQUISITION REFORM ON NEW DEVELOPMENT PROJECTS

Acquisition reform is a broad initiative for improving the way DoD development projects are procured, monitored, and funded. Acquisition reform has prompted the implementation of a number of streamlined accounting and contract-structuring practices throughout the last five years. Because the initiative is broad enough to change how DoD projects progress, the Norden-Rayleigh model may no longer apply to modern development programs. Our research, therefore, focused on a number of programs initiated after the acquisition reform was instituted to test this possibility. Our results show that although acquisition reform may benefit the overall outcome of a program, it does not affect the assumptions of the Rayleigh model.

The "Invisible Benefit"

Program office personnel echoed a common theme when we asked about the effectiveness of acquisition reform: they believe in its effectiveness, and they believe it saves time and money. However, they refused to place a dollar value on money saved by acquisition reform. One program office representative referred to acquisition reform as the "invisible benefit." He did not imply that the benefit was illusory or insignificant; he simply meant that measuring its impact directly would be difficult, because savings are interspersed across several activities. Only one program manager placed a concrete dollar value on savings from acquisition reform: he cut the estimate at completion for his airframe development project by 2 percent.

Although program office personnel were universally reticent about correlating a bottom-line savings to acquisition reform, they were equally unified in stating that acquisition reform helps maximize the chance for success of modern development contracts. The major benefit of acquisition reform is that the projects are better scoped and managed than before. The benefits, although intangible, obviously affect the development team's ability to deliver a product that meets it full set of requirements on time and under budget.

The Effects of Acquisition Reform on the Rayleigh Model

Reviewing the model's basic assumptions helps with understanding how acquisition reform affects the Rayleigh model. First of all, the Rayleigh model is based

[14] Chapter 4 describes the theory and use of splines.

on the assumption that a development project starts with a small group with little existing work to build on. The group expands in size and expends greater and greater amounts of resources until they begin to solve more problems than they uncover. At this point, the spending for the project decelerates as the development team continues to solve remaining problems. Eventually, the work concludes when there are no problems left to solve.

Acquisition reform borrows from best commercial practices, encourages the use of off-the-shelf products, and streamlines the acquisition process. Each of these goals makes pursuing the development process easier for engineers. It also improves the efficiency of project performance assessment, so potential problems can managed more timely and cost-effectively. The characteristics, while critical for improving the potential for success, do not change the fundamental development process: projects still start with a small team that expands to investigate new problems, then subsequently contracts as the number of problems left to investigate diminishes. As a result, the cost history of modern development projects should remain essentially the same as their pre-acquisition reform predecessors. Acquisition reform does not change the gross cost history of the development, but it may well make the money spent more effective.

An Exception to the Rule: Single Objective Development

The major question to consider when deciding whether the Rayleigh model will apply after a change in procedure, then, is not how well the change improves the effectiveness of the development, but whether it introduces or accompanies a fundamental change in the process itself. While the information uncovered by our research shows that acquisition reform generally is used to enhance the effectiveness of the engineering development process, it has potential for facilitating a change in the process if there is motivation to make one.

For example, large DoD development projects often use two contracts for two different phases of development. The first contract, or set of contracts, funds an initial R&D effort. The second funds the full-scale development of the commodity. Theoretically, the work from one contract should build on the work of another, so both contracts should be modeled by one Rayleigh curve. This is not the case, however. Historically, the start of the full-scale contract begins after the end of the first contract, forcing the development team to scale down and then rebuild from scratch. As a result, the two contracts must each be modeled by their own Rayleigh curve. This practice demonstrates a paradigm shift, made possible by acquisition reform, actually improves the applicability of the Rayleigh model.

The land-based mobile weapons system program listed earlier in this chapter has such a contract structure. However, in trying to reduce the scale down and subsequent ramp-up that occurs at the end of the initial contract and at the start of the full-scale contract, the program office scheduled an overlap between the two.

This contracting strategy is known as *single objective development*. Because of the tighter coupling between the initial and full-scale development afforded by this strategy, the MMAE algorithm was able to estimate reasonably accurately the cost of the entire development on the basis of data from the initial contract. The program office hypothesized that once sufficient data from the second contract is available, a two-Rayleigh model should be able to accurately estimate the development's total cost and completion time according to the accumulated cost of both contracts. The program office maintains that the second contract will have a Rayleigh-like expenditure history, so the second contract can be modeled independently as before. However, being able to consider both the initial contract and the full-scale contract together should make the estimate more accurate earlier in full-scale development.

Summary

Our discussions with program offices have shown that the Rayleigh model remains as useful a tool for evaluating ongoing programs today as it has been historically. The consensus from program offices is that although acquisition reform is significantly beneficial, it does not fundamentally change the traditional Rayleigh-like shape of development contract cost histories. Because acquisition reform helps to streamline some of the administrative aspects of contracts, Rayleigh may be even more effective for post-reform projects than for pre-reform projects. This is because the Rayleigh model is primarily concerned with how engineers form teams and solve problems, not with how the acquisition process performs. When the acquisition process becomes more efficient, the development team will have more freedom to pursue the development. Some acquisition reform practices do show the potential to facilitate a change in engineering development processes. The single objective development strategy has shown that acquisition reform can remove contractual barriers to engineering problem-solving. However, in this particular case, the effect is to extend, rather than reduce, the Rayleigh model's applicability.

The main factor to consider appears to be the flow of the engineering design and development process. In the programs studied, the majority of deviations from the Rayleigh model were caused by changes in customer requirements or changes prompted by disruptive events outside the development process. The addition of new product variants, the imposition of new software standards, requirements rollover, Congressional budget decrements, and major feasibility studies have the potential to interfere with the engineering development process. Because the engineering activities that drive the overall cost history of development contracts, these disruptions can hinder or even remove the applicability of the Rayleigh model.

Major hardware purchases do not fit into the category of outside disruptions to development, but they have a similar impact to external disruptions. Whenever a commodity under development incorporates significant off-the-shelf hardware, a good possibility exists that the hardware will be purchased in a short period of time, resulting in a significant surge in expenditures. Although the expenditures are a normal part of the engineering the model, they can also reduce the effectiveness of the Rayleigh model because the model does not account for sudden spending departures.

Accounting adjustments and government-funded components of development efforts, for example government-funded missile testing, also cause departures from the underlying expenditure history. These accounting issues affect the data by introducing cost reporting periods in which the money reported does not accurately reflect the level of effort of the development team. Since the Rayleigh model relates cost history to level of engineering effort, the data points do not fit the expected spending patterns.

Despite providing the continued applicability of the Rayleigh model, this research has discovered some circumstances in which the model does not apply. One of these cases, modern software development, required actually changing in engineering development. The other cases involve contractual structures that break with the assumption of a single project starting and completing within one contract. The process for software development has changed in DoD programs to incorporate commercial practices. While this seems to have little effect on one-time software that is integrated with hardware, pure software projects may follow a distinctly non-Rayleigh pattern. Like commercial off-the-shelf software, these projects have rolling requirements, periodic version updates and service patches, and continuous manning of architecture, development, and testing teams. As a result, the projects may be better modeled as level-of-effort maintenance projects instead of traditional development projects.

The software development issue has an additional structural element that also may apply to large, long development efforts. For the sake of accounting considerations, the contracts may fund the same development project with several consecutive contracts. Even if the underlying pattern of the total project is Rayleigh-like, the cost reports of each contract may not appear Rayleigh-like because each contains only a part of the total expenditure history. Usually, a Rayleigh curve can be fit to each contract in the project, and, in some cases, (particularly when modeling the last contract of a series) this may lead to an accurate estimate at completion for the entire development project. However, all but the last contract in such a series will have an arbitrary termination point. By definition, arbitrary termination points cannot be modeled by this or any other estimating tool. Finally, umbrella contracts do not necessarily follow a single Rayleigh pattern.

For example, if a contract funds five essentially independent projects, then each project should be modeled separately. Usually the subprojects in these contracts must be integrated at some point. If this is the case, and the integration effort is sufficiently substantial, then the contract may well fit a single Rayleigh model. However, if minimal integration is required, the individual magnitude and dynamics of each subproject can give the overall expenditure history virtually any shape that can be imagined. The analyst should consider umbrella contracts carefully and decide whether each can be modeled by a single Rayleigh model, a multiple Rayleigh model, or by modeling each sub project separately.

Chapter 4
Spline Theory

BACKGROUND AND THEORY

The previous chapter discussed several types of programmatic events that can greatly affect the accuracy of a Rayleigh estimate at completion. In a few limited cases, especially when contracts are structured in ways that do not meet with the assumptions for applying the model, the model may not be valid to apply. It is difficult to tell if a model is valid solely by looking at a set of periodic cost reports. The data must be compared to a non-parametric curve that can be used for comparing with a Rayleigh curve and its derivative. Such a curve must be flexible enough to capture the features of the data faithfully, yet stable enough to avoid inserting artifacts such as "phantom" peaks and troughs into the data. In addition, because the cost data may be noisy and contain local oddities, such as accounting adjustments and one-time purchases, a curve that can provide "local control"—that is, isolate the effect of a single data point to a small region of the curve—would be beneficial. *Polynomial splines* provide these features.

Polynomial splines are piecewise polynomial functions. By piecewise, we mean that a given function (curve) can be divided into segments, and the individual segments can be represented by distinct polynomials. At each endpoint, the polynomials must be "smooth" and continuously differentiable. A polynomial is an algebraic function that maps the set of real numbers onto itself, often represented in the form:

$$P_n(x) = a_0 + a_1x + \cdots + a_nx^n \qquad \text{[Eq. 4-1]}$$

where: n = a nonnegative integer $a_0, \ldots, a_n$ = real constants. Polynomials have the important property of uniformly approximating continuous functions.[15] In addition, derivatives of polynomials are easy to determine and are polynomials themselves.

Three sets of items are needed to define a spline: a *knot sequence*, a set of *basis functions*, and a set of *coefficients* to be interpolated. The knot sequence is a set of non-decreasing real numbers that define a set of intervals that map to each segment in the piecewise curve. The basis functions are a set of linearly independent interpolating functions that span the space of the spline. The coefficients are values that map to a particular knot. The number of the basis functions depends on the order of the spline and the number of intervals. The order of the

[15] Burden, R.L. and J.D. Faires, "Interpolation and Polynomial Approximation," *Numerical Analysis*, 3rd Edition, PWS Publishers, 1985, page 79.

spline depends on the degree of the piecewise functions that comprise it. For example, a second-order spline consists of piecewise linear segments (or lower); third-order, piecewise quadratic; fourth-order, piecewise cubic, etc. (i.e., the order of the polynomial is equal to n+1 in Eq. 4-1). For most practical applications, fourth order splines are the highest order used.

A detailed presentation of spline theory is given many texts, one being Farin's book on computer-aided curves and surfaces.[16] The value of a spline at any given value x is a linear combination of the basis functions and the coefficients. Mathematically, a spline is expressed as

$$S_{N,k}(x) = \sum_{i=1-k}^{N-1} c_i F_{i,k}(x) \qquad \text{[Eq. 4-2]}$$

where: $S_{N,k}$ = k-th order spline containing N segments; c_I = the i-th coefficient to be interpolated; $F_{I,k}(x)$ = the i-th spline basis function of order k.

Several basis functions have been formulated for splines. The *basic splines* form a convenient set of polynomial basis functions. The first-order basic splines are piecewise constant and have the form

$$M_{i,1}(x) \equiv \begin{cases} \dfrac{1}{x_{i+1} - x_i}, x_i \le x < x_{i+1} \\ 0 \text{ otherwise} \end{cases} \qquad \text{[Eq. 4-3]}$$

The process of defining higher order splines is recursive; the definition of a k-th order basic spline depends on two basic splines of order k-1:

$$M_{i,k}(x) \equiv \frac{k}{x_{i+k} - x_i} \int \left[M_{i,k-1}(u) - M_{i+1,k-1}(u) \right] du \qquad \text{[Eq. 4-4]}$$

Notice that the splines are only non-zero when $x \in [x_I, x_{I+k}]$. This means that changing an interpolating coefficient will only affect k curve segments around it.

The *Rayleigh Analyzer* uses spline smoothing instead of spline interpolation for defining the set of coefficients. Therefore, the spline segments do not necessarily coincide with actual data points. The user specifies the order and number of intervals for each spline generated; the knots are supplied automatically by the *Rayleigh Analyzer* and comprise a uniform sequence. When attempting to visualize the interrelationships of the knot sequence, the number of intervals, the spline order, and the data when creating a basic smoothing spline, it is best to vary

[16] Farin, Gerald, *Curves and Surfaces for CAGD,* 4th edition, Academic Press, Inc., 1997.

these items and view the results graphically. The next series of graphs (Figure 4-1 through Figure 4-3) show the effects of varying the order when fitting a one-interval spline to a quintic (5th order) function.

Figure 4-1. Comparison of 2nd Order Spline with Actual Data

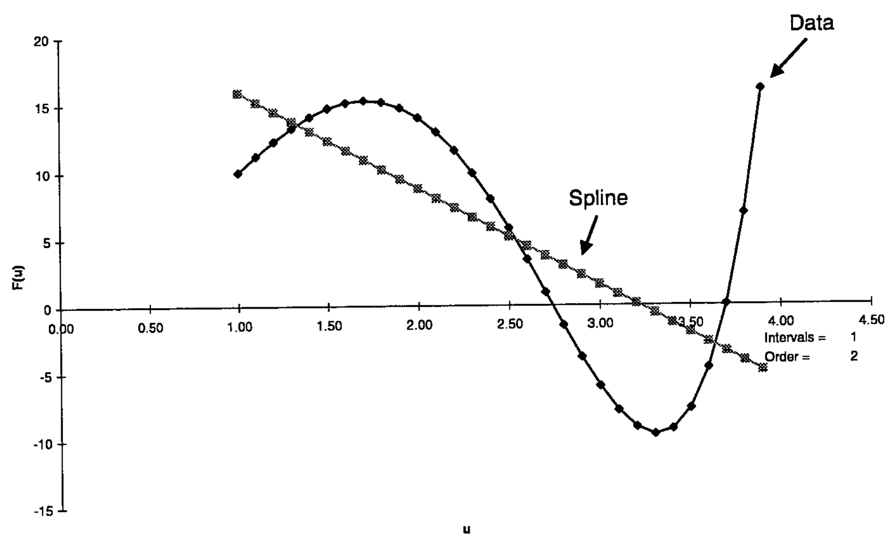


Figure 4-2. Comparison of 3rd Order Spline with Actual Data

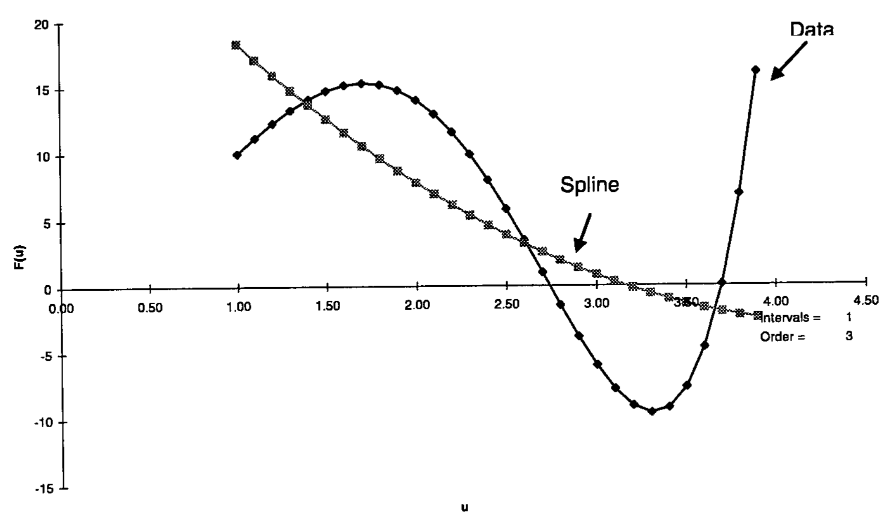
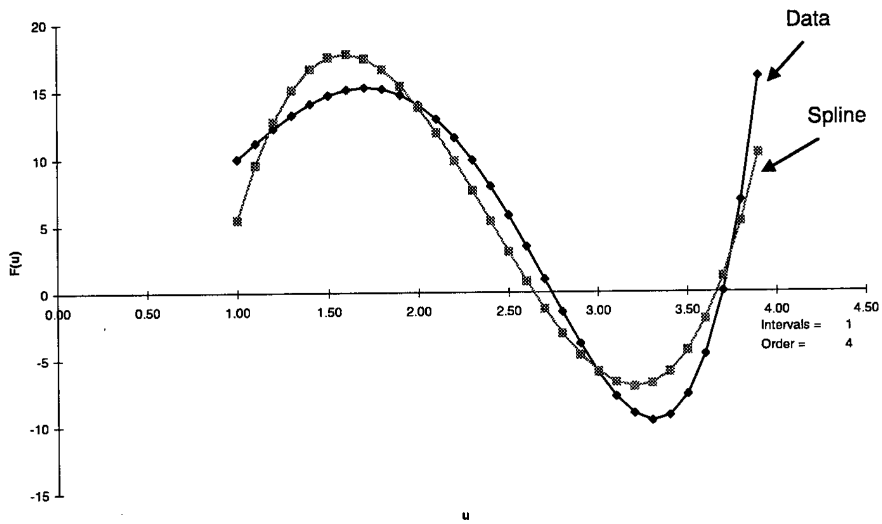


Figure 4-3. Comparison of 4th Order Spline with Actual Data



With only one interval, the effect of fitting a spline is to fit the best-fit curve of the specified degree. A first-order spline fits a constant to the data, a second order spline fits a line to the data, etc. Because a single-interval spline offers the fewest degrees of freedom with which to fit the data, the fit is not very close. To see how well the fit improves by adding more intervals (and thus adding more curve segments), the following graphs show an eight-interval spline that uses the same order and knot sequence as the one-interval splines in Figure 4-1 through Figure 4-3.

The splines in Figure 4-4 through Figure 4-6 fit the data far better than the one-interval splines because the eight individual curve segments in the e-interval splines have far more degrees of freedom with which to fit the data. In fact, the splines fit the data so well, that in Figure 4-5 and Figure 4-6 the two curves. As a result, the piecewise linear (order = 2) curve with eight intervals fits the data more closely than the cubic spline with one interval. However, the increased closeness of fit comes with a potential risk. Splines with too many degrees of freedom may contain artifacts, particularly when used to fit noisy data such as cost history data for development programs. This means that the spline can introduce extra peaks and troughs in the topology of the curve that aren't warranted by the trends in the data. If these splines are differentiated, the problem can become far worse, because functions that are quite close in value everywhere may still have widely different derivatives at some points. Using a spline of an excessively high degree also can introduce artifacts in the fit that damage the accuracy of the derivative. For a mathematically rigorous discussion of the use of spline-smoothing numerical differentiation of noisy data, see Appendix C.

Figure 4-4. Comparison of 8-Interval, 2nd Order Spline with Actual Data

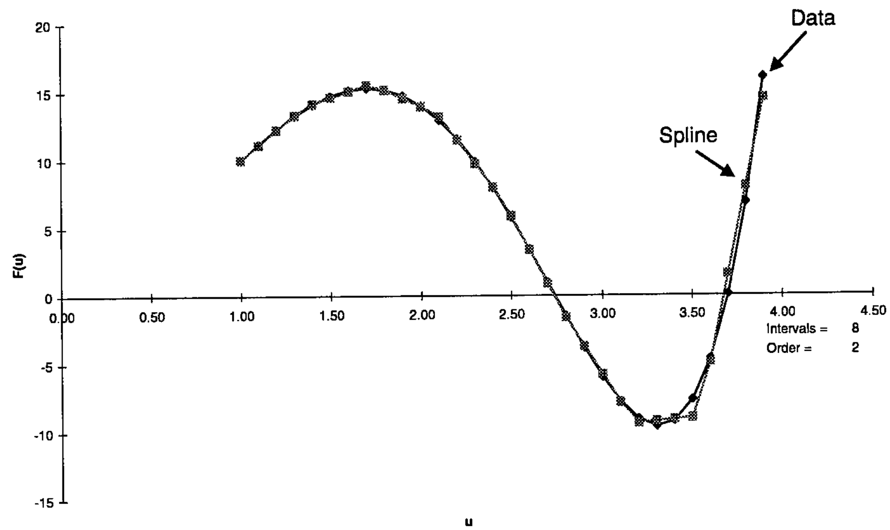


Figure 4-5. Comparison of 8-Interval, 3rd Order Spline with Actual Data

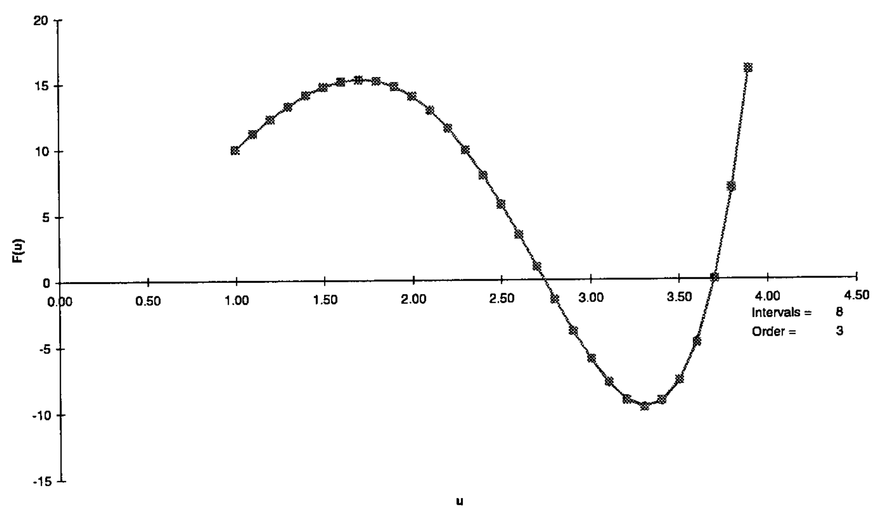
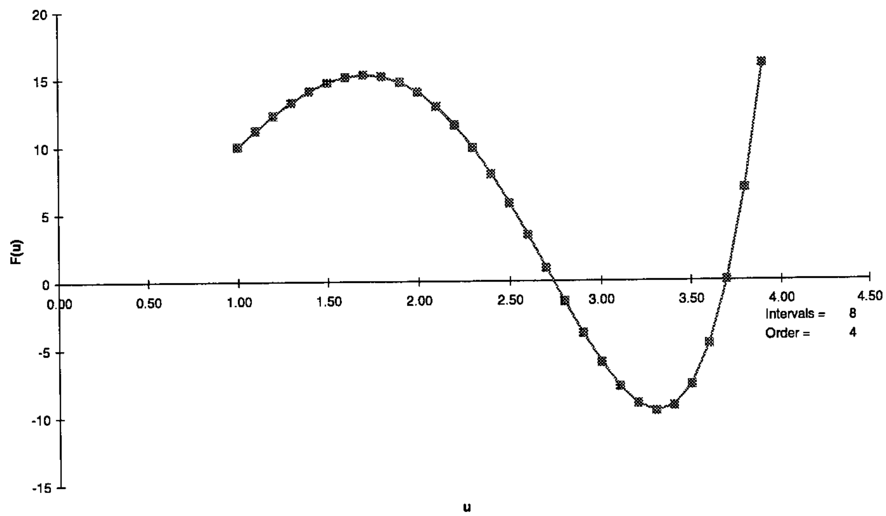


Figure 4-6. Comparison of 8-Interval, 4th Order Spline with Actual Data



EXAMPLE

By using real program data we can demonstrate how the number of intervals influences the quality of the spline fit and the accuracy of its derivative. Figure 4-7 represents a missile program with noisy data and a two-interval spline used to model the data.

In Figure 4-7, the spline fit uses too few intervals to fit the noisy data. The spline fit does not capture a slight dip in the actual data near the t = 3.00 range, and does not capture a subsequent peak near the t = 4.00 range. These discrepancies do not seem to be too large when looking at the data. However, the derivative of the spline fit (the expenditure rate curve) shows a very gradual, smooth curve with a single peak. This shape differs greatly from the expected shape; the ripples in the actual data imply that even larger ripples should be present in the derivative. Hence, the derivative should display pronounced perturbations at the t = 3.00 and t = 4.00 positions. Therefore, we conclude that two intervals are insufficient to capture the actual features of a data set of this size. In Figure 4-8, we see the effect of increasing the number of intervals to eight.

When the intervals are increased to eight, the spline has sufficient flexibility to closely match the data's local peaks and troughs. As a result, three distinct peaks are evident (a small peak at t = 1.5, a larger peak at t = 3.75, and a third peak at t = 5.75). According to the program office for this program, the peaks match actual events in the development project that caused temporarily accelerated spending.

Figure 4-7. Spline Fit of Noisy Program Data—Two Intervals

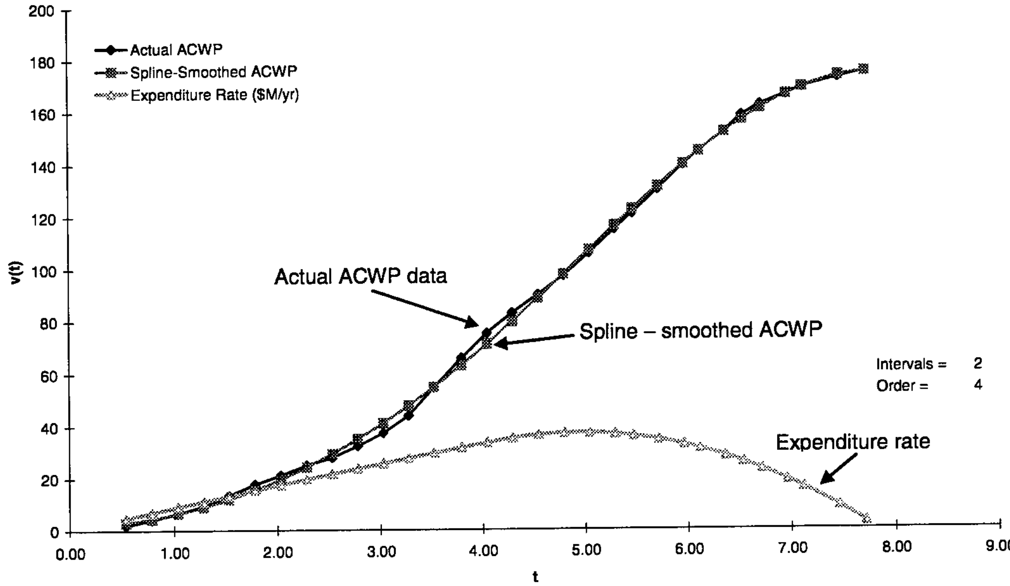
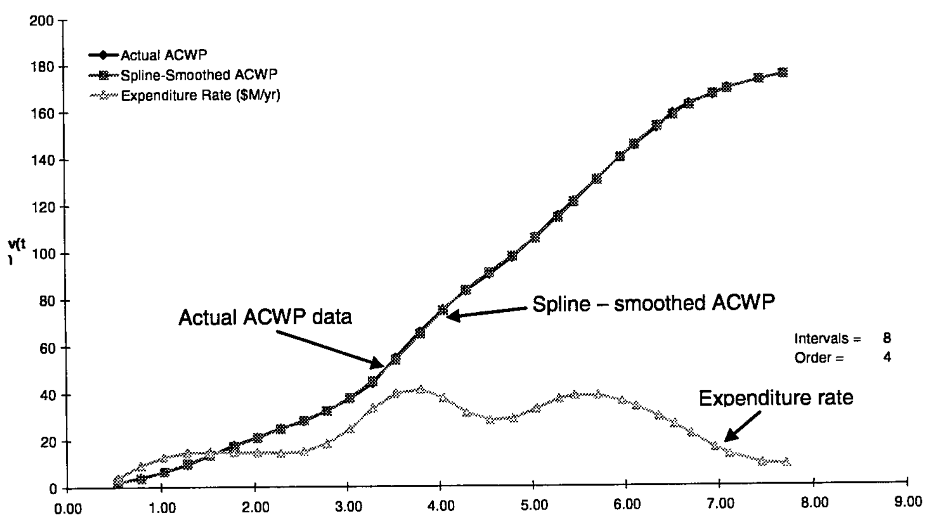
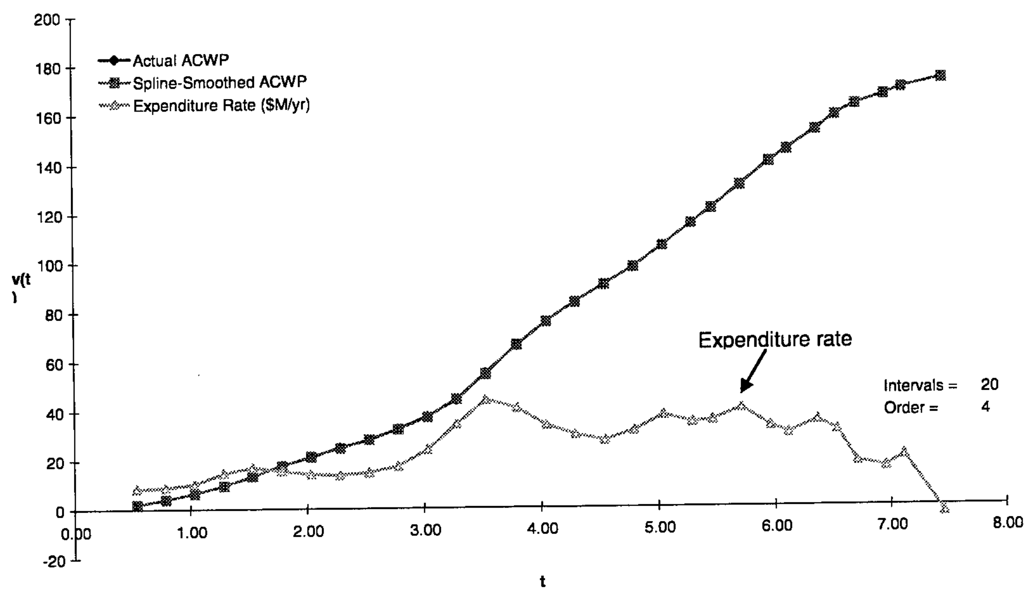


Figure 4-8. Spline Fit of Noisy Program Data—Eight Intervals



Because increasing the intervals from two to eight improves the spline fit and the derivative curve, increasing the number of intervals more is tempting. Figure 4-9 shows the error of such an idea.

Figure 4-9. Spline Fit of Noisy Program Data—20 Intervals



With twenty intervals, the spline does not seem to fit the data much better than it does with eight. However, the derivative curve is a lot more sensitive to noise. So much so, in fact, that the two distinct peaks evident in the eight-interval fit are obscured by noise in the twenty-interval fit. Appendix C, "Numerical Differentiation of ACWP Data," provides a thorough mathematical explanation of the issues involved with choosing appropriate parameters for modeling noisy ACWP data with splines. Clearly, the analyst needs to choose the number of intervals carefully. No explicit rule exists that the analyst can use for choosing intervals that fit every case. However, our research has provided the following guidelines:

- A choice of intervals corresponding to three or four data points per interval seems to produce the derivative curves that best match actual program events.

- More than three distinct peaks is a clear indication that too many intervals exist. After analyzing more than 150 DoD development programs by using four data points per interval, we seldom saw the derivative curves seldom showed evidence of more peaks than two, and almost never saw more than three peaks.

Thus, we see that splines can be a useful tool, if applied by a knowledgeable analyst.

Chapter 5
Summary and Recommendations

SUMMARY

The Norden-Rayleigh model for estimating total cost and completion time for development programs is based on the assumption that the cost histories of development programsS follow a Rayleigh cumulative distribution. This distribution quantifies what many cost analysts will recognize as a "classic S-shape" curve: the accumulated cost of work performed increases gradually over the first part of a development project, builds momentum until it reaches a peak in expenditure rate at about 40 percent complete, then gradually tapers until the expenditure rate finally becomes zero. Peter Norden first related this phenomenon to the development process in the early 1960s. He postulated that the staffing levels of a project could only be expanded or reduced gradually to effectively use the available manpower. Because spending for a development project is dominated by the relative staffing level of that project, the expenditure rate also should follow the staffing level of the development team. Historically, the Rayleigh model has enjoyed success in modeling major DoD development projects and software projects of all sizes.

MAJOR FINDINGS

With the advent of acquisition reform came doubt about whether the Rayleigh model would still apply. LMI researchers investigated ten programs that have been heavily affected by acquisition reform and fourteen development contracts to determine if the model remains applicable.

- We determined that the model is as applicable today as it was in the past, except for mature, pure software programs that produce periodic releases.
- The cyclical nature of software programs makes their activities resemble maintenance or production rather than development, but they are typically called development projects by convention. The Rayleigh model applies to software projects that are not cyclical (in other words, those that produce a finished product rather than versions and patches) and software-intensive hardware projects.
- In addition to the software exception, our investigations of the program office also identified several situations in which the Rayleigh model will not be applicable. These situations involve contract structure issues that invalidate the model's underlying assumptions. Multiple contracts that fund

one development project (unless each contract funds a complete, individual development, such as the combat suite for a warship of the engines for an aircraft), umbrella contracts that fund several non-integrated projects, and contracts that are suspended or interrupted all invalidate the assumption of a single contract that funds a single, continuous, integrated development project. These circumstances apply to both pre- and post-acquisition reform programs.

- After the program office investigations, we formulated three composite Rayleigh models: the two-Rayleigh model, the three-Rayleigh model, and the modified Rayleigh model. These models address specific issues that emerged from interviews with program office representatives. These models significantly more accurate than a single Rayleigh model in limited situations.

- The research team used polynomial splines as a non-parametric means of identifying trends in the expenditure rates of the programs investigated. Splines are piecewise polynomial functions that are continuous at their junction points. By fitting a set of cost report data using a spline, and differentiating the spline, we could obtain an accurate expenditure rate curve, even when noise exists in the data.

FURTHER RESEARCH

Our research shows that the MMAE method of estimating parameters is a robust, useful method for finding the best Rayleigh curve for extrapolating a set of noisy cost report data. The method is far superior to simple curve-fitting techniques such as linear programming optimization. Currently, researchers have only implemented a single Rayleigh model within an MMAE framework. However, MMAE is a general Bayesian-state estimation technique that can be applied to several models, including the composite Rayleigh models developed as a result of our program office investigations. Because composite Rayleigh models can more accurately model certain expenditure profiles than a single Rayleigh model can, implementing these composite models into MMAE or another Bayesian state estimation model would be valuable. Ideally, the user should be able to choose from any of the models mentioned above and execute each using MMAE.

MMAE represents only one Bayesian state estimation method that may be used to create a framework for Rayleigh-based estimate at completion software. The MMAE method is computationally intensive, and the computations can become prohibitive as new degrees of freedom are added to the underlying model. Because the composite models have two to three times the number of degrees of freedom as a single Rayleigh model, investigating other parameter-identification methods that are not as computationally intensive may be valuable.

Appendix A
Abbreviations

ACWP	Actual cost of work performed
CAIV	Cost as an independent variable
DoD	Department of Defense
E&MD	Engineering and manufacturing development
FSD	Full-scale development
G&A	General and administrative
LMI	Logistics Management Institute
MMAE	Multiple model adaptive estimation
OSD	Office of the Secretary of Defense
R&D	Research and development

Appendix B
Bibliography

Burden, R.L. and J.D. Faires, "Interpolation and Polynomial Approximation, Numerical Analysis," 3rd ed., PWS Publishers, 1985, page 79.

DoD Secretary William Perry, "Specifications and Standards—A New Way of Doing Business," Memorandum, (Washington, DC, June 29, 1994).

Farin, Gerald, *Curves and Surfaces for CAGD*, 4th ed., Academic Press Inc., Chestnut Hill, MA, 1997.

Gallagher, M. A. and D. A. Lee, "Final-Cost Estimates for Research & Development Programs Conditioned on Realized Costs," *Military Operations Research*, Vol. 2, No. 2, (1996): pp. 51–65.

Lee, D. A., J. A. Dukovich, *Using the Rayleigh Analyzer®*, Report AT701C1 Logistics Management Institute, McLean, VA, March 1998.

Lee, D. A., *The Cost Analyst's Companion*, Logistics Management Institute, McLean, VA, 1997.

Lee, D.A., M.R. Hogue, and D.C. Hoffman, "Time Histories of Expenditures for Defense Acquisition Programs in the Development Phase—Norden-Rayleigh and Other Models," Presented at the 1993 Annual Meeting of the International Society of Parametric Analysis.

Norden, P. V., "Useful Tools for Project Management," *Operations Research in Research and Development*, B. V. Dean, Editor, John Wiley and Sons, 1963.

Putnam, L.H. and W. Myers, *Measures for Excellence-Reliable Software onTime within Budget*, Prentice-Hall, Upper Saddle River, NJ, 1992.

Appendix C
Numerical Differentiation of ACWP Data

INTRODUCTION

Numerical differentiation is well known as a serious challenge to the analyst. The basic difficulty is that two functions may be quite close to one another, yet have derivatives that differ hugely at some points. Observations always involve errors, so at first glance inferring derivatives from data seems hopeless. Indeed, for many years mathematicians considered problems whose solutions amounted to differentiating functions known only through observations as "ill-posed," and, therefore, not suitable as models of physical processes. (A well-posed problem has a unique solution, and the solution depends continuously on the data. Differentiation doesn't depend continuously on data.)

Over the past few decades, the mathematical community regarded problems whose solutions require estimating derivatives of functions that are known only from observations as members of the class of "regularizable ill-posed problems"[1]. Problems of this class are suitable models of physical systems. Nevertheless, their treatment requires special care. Estimating the derivative of a function from noisy data, like solving regularizable ill-posed problems in general, is characterized by the necessity to trade resolution–the ability to resolve fine structure in the derivative–against sensitivity to noise[2].

This appendix explains the care that we took, in inferring values of the rate of change of earned value with respect to time. We believe that careful work is particularly important in this part of our study so that we may be confident that the features we infer are truly in the data, and not artifacts of our differentiation scheme.

We begin our explanation with an example, the simple centered-difference scheme, with which many readers will be familiar. When the scheme is applied to noisy data, its results are random variables. We show that the expected values are "windowed" averages of the desired derivative, with a rectangular kernel whose width is the data spacing.

This averaging "smoothes out" higher frequency components of the derivative. We see that the expected values of the results of the centered-difference scheme

[1] Sabatier, P.C., editor, *Tomography and Inverse Problems*, Hilger, Bristol, 1987.

[2] Readers familiar with numerical treatment of regularizable ill-posed problems may wish to skip to the discussion of our specific method, which begins in the section, "Spline Smoothing Numerical Differentiation" on page C-12.

are the output of a low-pass filter, whose input is the desired derivative. The bandwidth of this filter may be computed, and that gives a convenient quantitative characterization of the resolution achieved by the centered-difference scheme. We find that the bandwidth is inversely proportional to the data spacing.

The standard deviations of the dispersion of the centered-difference scheme's results are inversely proportional to the data spacing. Thus, as the data spacing decreases, the scheme becomes more sensitive to noise. This result, combined with the bandwidth result, quantifies the resolution and noise sensitivity for the centered-difference scheme. We show an example of how this trade works out in practice. We also use the example to illustrate a method for estimating an optimal data spacing.

We do not use the simple centered-difference scheme to estimate time rates of change of earned value because evidence that exists more sophisticated schemes give better results. The remainder of the appendix describes the scheme that we do use, spline-smoothing numerical differentiation (SSND), and our particular application of it.

Expected values of the SSND scheme's results, like those of the centered-difference scheme, are windowed averages of the desired derivative. These averages also are outputs of low-pass filters operating on the desired derivative. The resolution of SSND may be characterized by the bandwidth of the filter, which we compute.

Standard deviations of SSND results increase, as resolution increases. We introduce a procedure for finding the best practical resolution for our SSND, for given data sets. We used this procedure to determine rates of change of earned value for the acquisition programs we studied. The appendix concludes with an illustration of the procedure, using data for an airframe development program.

NUMERICAL DIFFERENTIATION OF NOISY DATA: AN EXAMPLE

We illustrate the general characteristics of numerical differentiation with a discussion of the familiar centered-difference scheme,

$$\hat{f}'(t_m) \equiv \frac{\tilde{f}(t_0) - \tilde{f}(t_1)}{t_0 - t_1} \qquad \text{[Eq. C-1]}$$

In equation C-1, $\hat{f}'(t_m)$ denotes the estimate of the derivative of f(t) at t_m, the midpoint of (t_0, t_1). (Tildes denote noisy values.)

Expected Value

With the standard assumptions that

$$\tilde{f}(t_i) = f(t_i) + \varepsilon_i, < \varepsilon_i \varepsilon_j >= \sigma^2 \delta_{ij} \qquad \text{[Eq. C-2]}$$

(we use the notation <Q> for the expected value of the quantity Q) one sees that the expected value of $\hat{f}'(t_m)$ is given by

$$< \hat{f}'(t_m) >= \frac{f(t_0) - f(t_1)}{t_0 - t_1} = \int_{x_0}^{x_1} \frac{1}{t_1 - t_0} f'(u)du \qquad \text{[Eq. C-3]}$$

Equation C-3 shows that the expected value of our centered-difference scheme is the average of the desired derivative, over the interval (t_0, t_1). If we let R denote the range of data for f, the centered-difference scheme's expected value at a point t can be written as

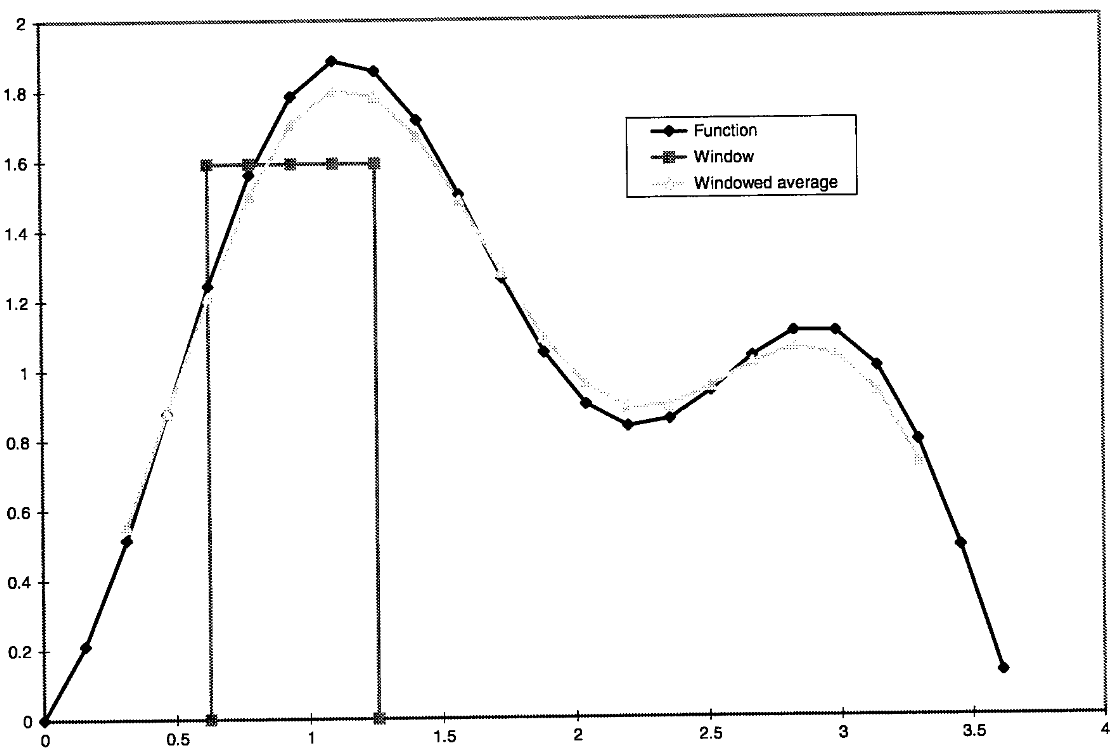
$$< \hat{f}'(t_m) >= \int_R W(t_m, t) f'(t)dt = \int_R K(t_m - t) f'(t)dt \qquad \text{[Eq. C-4]}$$

where

$$K(t) = \begin{cases} 1/\Delta, |t| < \Delta \\ 0, otherwise \end{cases} \qquad \text{[Eq. C-5]}$$

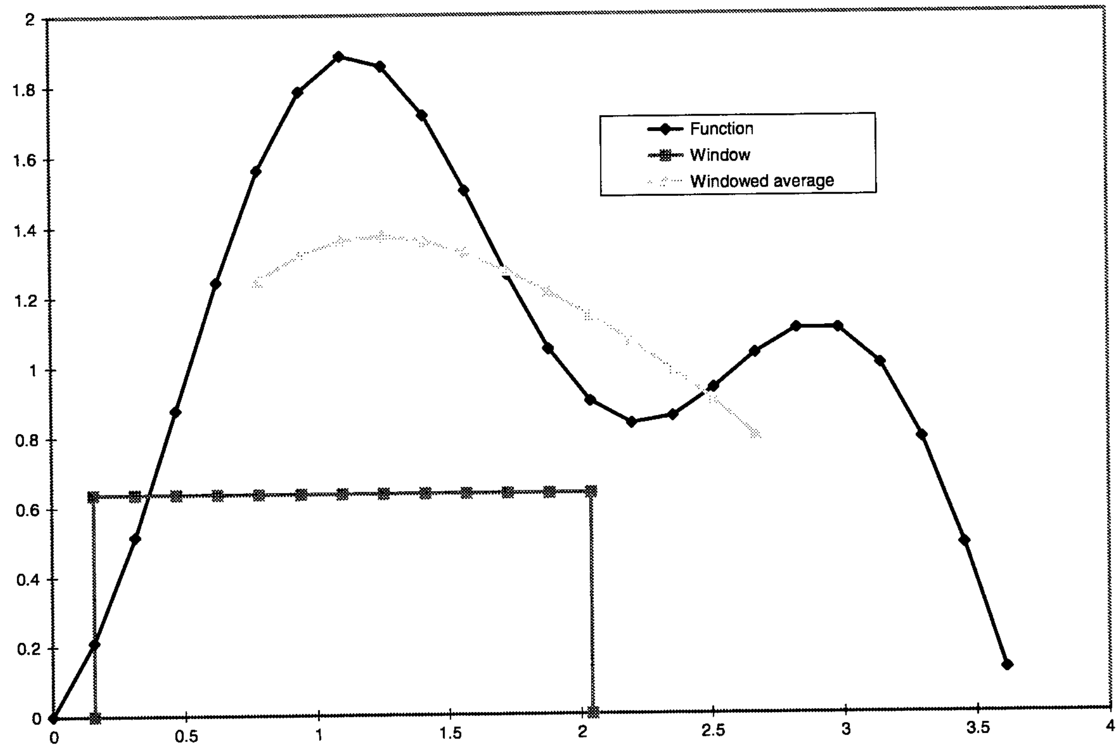
and where we write Δ for the data spacing t_1 - t_0. Thus we see that $< \hat{f}'(t_m) >$ is a windowed average of the desired derivative. The window is rectangular, and its width is Δ. Also, the kernel of the average is a difference kernel, so the average is a convolution. Figure C-1 illustrates the situation.

Figure C–1. Function, Window, and Windowed Average



Features of f'(t) that occur on time intervals smaller than Δ will be smoothed out by the averaging that the simple centered difference scheme imposes. In the example of Figure C-1, the window is sufficiently narrow–that is, the data spacing is sufficiently small–that the windowed average preserves the bimodal character of the function. By contrast, in Figure C-2, the window is so broad that this feature is lost. This is the sense in which increasing Δ decreases the ability of our approximation to resolve "fine structure" in f'(t).

Figure C–2. Function, Window, and Windowed Average Using Broad Window



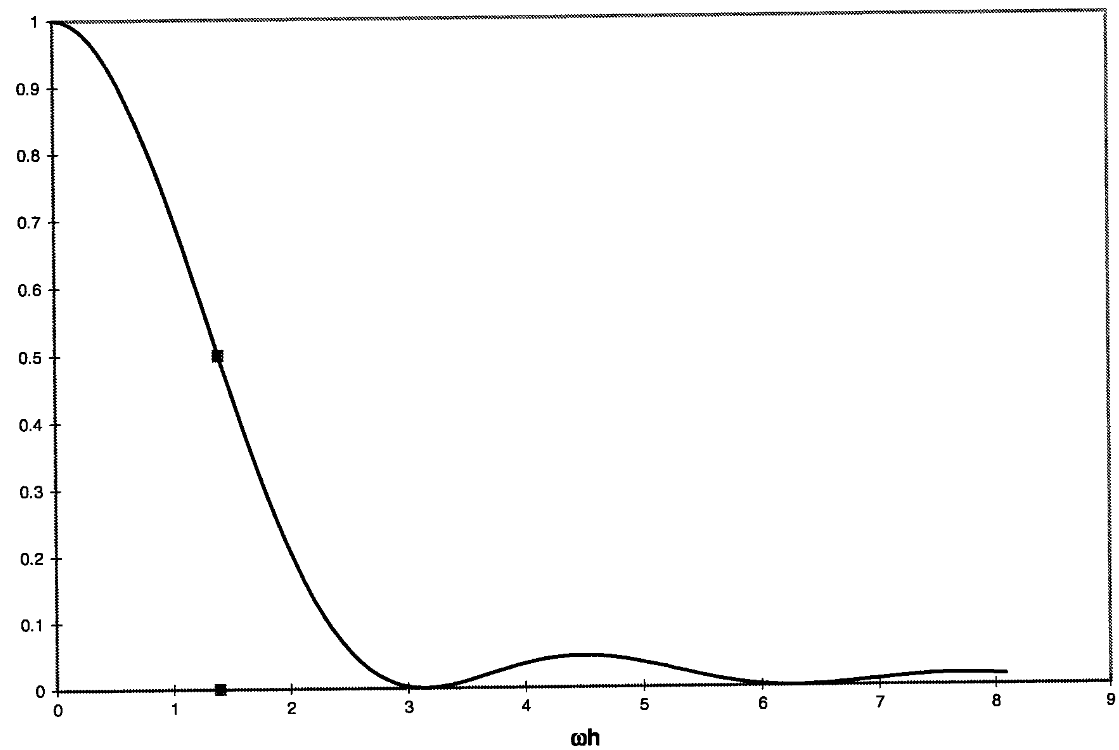
Bandwidth of the Centered-Difference Scheme

We can make this resolution concept more precise, and quantitative, by frequency-domains. If f'(t) has Fourier transform g(ω), then the Fourier transform of the $< \hat{f}'(t_m) >$ corresponding to f'(t) will have Fourier transform $\hat{g}(\omega)$, where

$$\hat{g}(\omega) = \frac{\sin(\omega h)}{\omega h} g(\omega) \qquad \text{[Eq. C-6]}$$

where h is half the data spacing, i. e., Δ/2. The function sin(ωh)/ωh has the value 1 when ω is zero, and decreases with increasing ω out to ω = π/h. Figure C-3 shows the filter's (power) transfer function, which is the square of the square of sin(ωh)/ωh.

Figure C–3. Power Transfer Function for Divided-Difference Filter



Note: the emphasized abscissa and point show the filter's half-power point

As illustrated in Figure C-3, the half-power frequency of this filter is at $\omega_p(\Delta)$, where

$$\omega_p(\Delta) = \frac{2.78}{\Delta} \qquad \text{[Eq. C-7]}$$

This means that the averaging performed by the centered-difference scheme will essentially remove components of f'(t) corresponding to frequencies greater than $\omega_m(\Delta)$, while passing lower frequencies. Thus, the expected value of the simple centered-difference scheme gives the output of a low-pass filter applied to the desired derivative. The bandwidth $\omega_p(\Delta)$ of this filter, or, equivalently, the shortest period of the components that the filter passes, which is $2\pi/\omega_p(\Delta)$, gives a convenient quantitative measure of the resolution of a given centered-difference scheme.

Variance of the Centered-Difference Scheme

Turning to the dispersion of the centered-difference scheme, we see that

$$\mathrm{var}(\hat{f}'(t_m)) = < \frac{(\varepsilon_0 - \varepsilon_1)^2}{\Delta^2} > = \frac{2\sigma^2}{\Delta^2} \qquad \text{[Eq. C-8]}$$

so that the standard deviation $\hat{\sigma}_m$ of $< \hat{f}'(t_m) >$ is given by

$$\hat{\sigma}_m = \sqrt{2}\frac{\sigma}{\Delta} \qquad \text{[Eq. C-9]}$$

where we have written Δ for $t_1 - t_0$. We see from equation C-8 or C-9 that the dispersion of the simple centered-difference approximation increases as Δ decreases for given σ.

Equations C-8 or C-9 and C-7 give helpful insights into using of the centered-difference scheme. To capture features of f'(t) for frequencies as large as some given frequency ω_0, the noise in the data (measured by σ), must be sufficiently small so that the dispersion it causes (shown by equation C-8 or C-9) is small compared with the values of f'(t) that are of interest, when Δ is small enough that $\omega_p(\Delta)$ (shown by Equation C-7) is less than ω_0. The discussion above tells workers how good the data must be, i.e., how small σ must be, to achieve a given resolution for an experiment.

Dealing with existing data, if σ and a typical value of |f'(t)| are known, equations C-8 and C-7 taken together show that the greatest resolution that can be achieved with the centered-difference scheme, i.e., the largest frequency such that components of f with that frequency can be inferred with roughly 10 percent uncertainty, is close to

$$\omega_{\max} = \frac{2.78|f'|_{typ}}{10\sqrt{2}\sigma} \qquad \text{[Eq. C-10]}$$

Inferring Best Practical Resolution From Data

If little is known about the desired derivative, but σ is reliably estimated, one may proceed in this way to determine the resolution justified by a given set of data, with a corresponding estimate for f'(t). Begin with a sufficiently large value of Δ that, for all points t_m of interest, the dispersion of $\hat{f}'(t_m)$, determined by Equation C-8, is small enough to give confidence in the features of $\hat{f}'(t_m)$. Decrease Δ, until the dispersion of $\hat{f}'(t_m)$ first becomes so large that this is not true. This is the smallest practically useful value of Δ. Equation C-7, for this value of Δ, then gives the resolution that the data justify.

We illustrate this procedure with an example. Figure C-4 shows exact and noisy values of a certain function. Figure C-5 shows the function's derivative, together with results of Equation C-1 applied to machine-accurate values of the function. This confirms that the scheme is capable of evaluating f'(t) quite accurately, given accurate values of f.

Figure C–4. Exact and Noisy Values of a Function

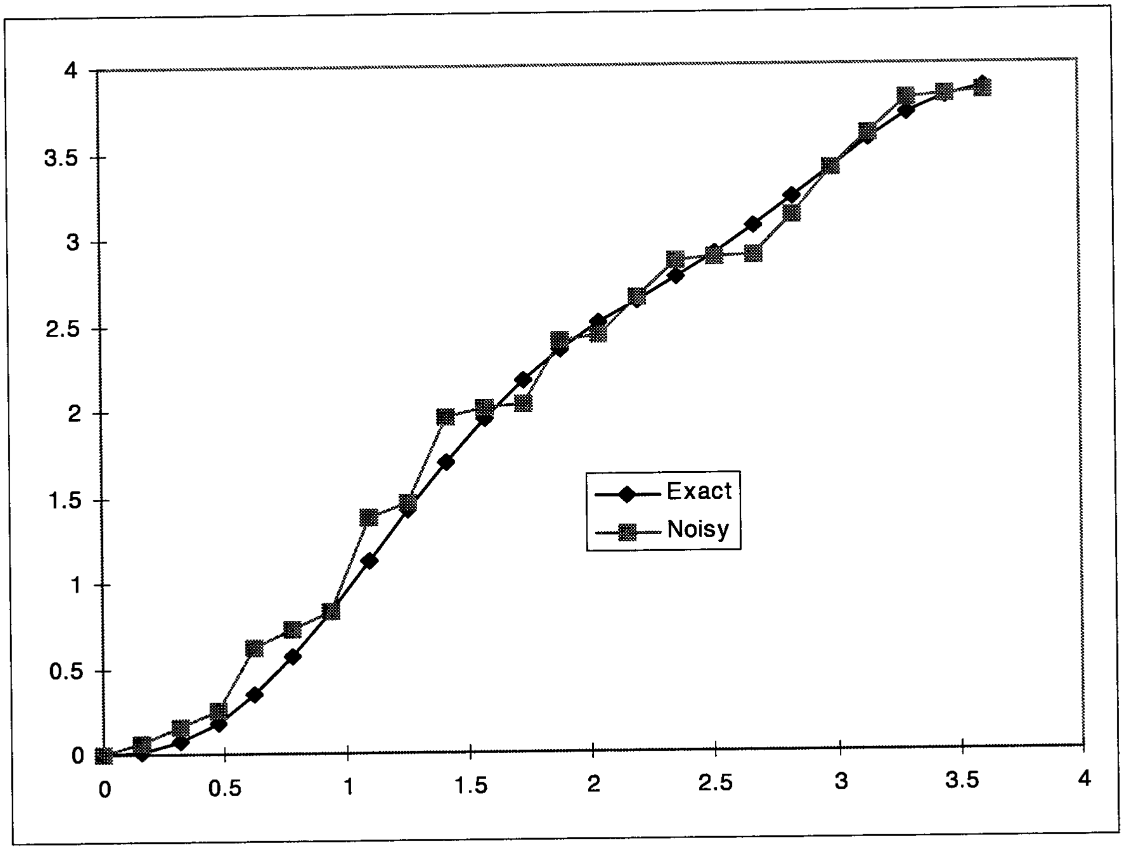
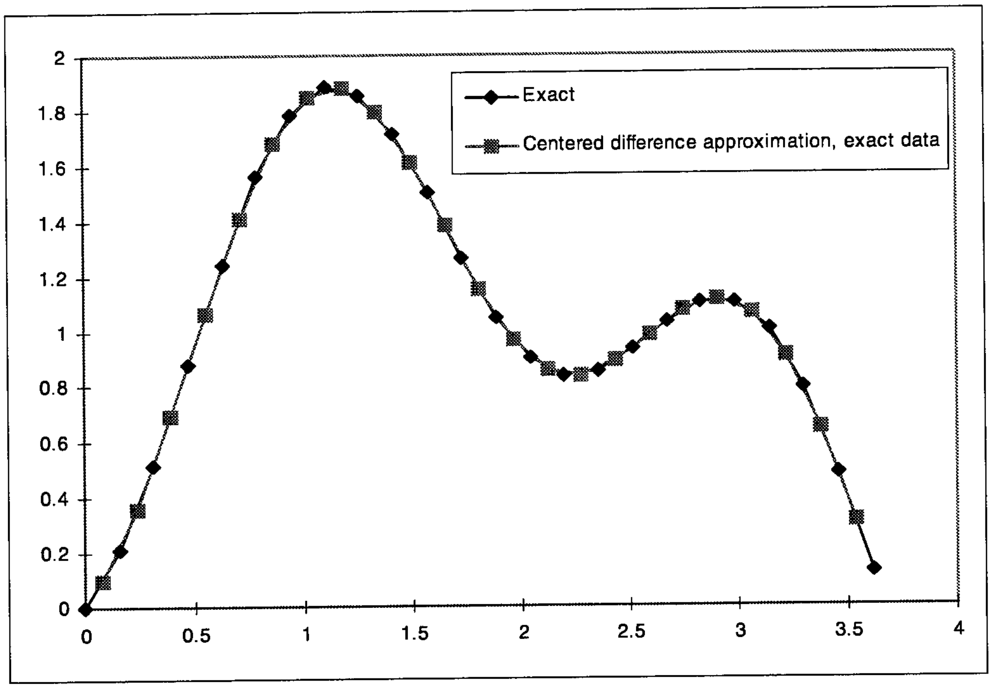


Figure C–5. Exact Derivative and Results of Difference Scheme with Exact Data

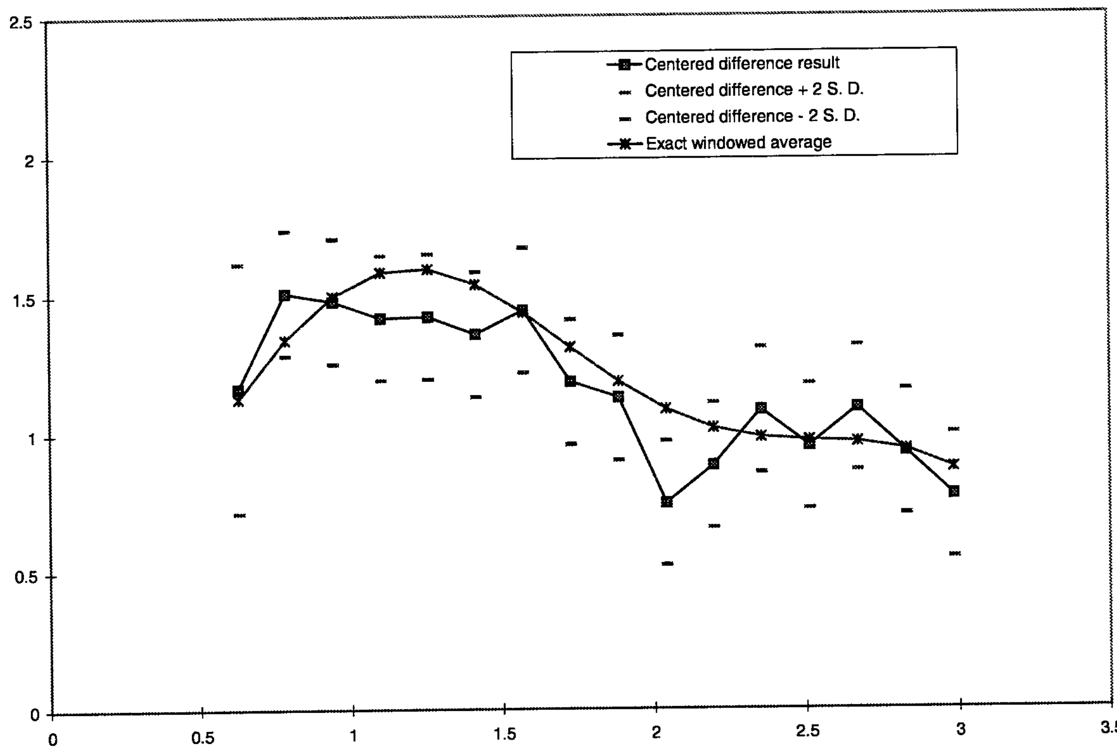


Now we consider the procedure, applied to the noisy data of Figure C-4. For this contrived example, we know the standard deviation of the errors exactly, so we can use Equation C-9 to generate error bars on our estimate of the windowed average that the centered-difference scheme generates (*not* on the derivative itself).

Figure C-6 shows the result for a fairly large spacing. The error bars indicate that the result's general features probably are reliable. The resolution isn't good enough to show the bimodal character clearly, however.

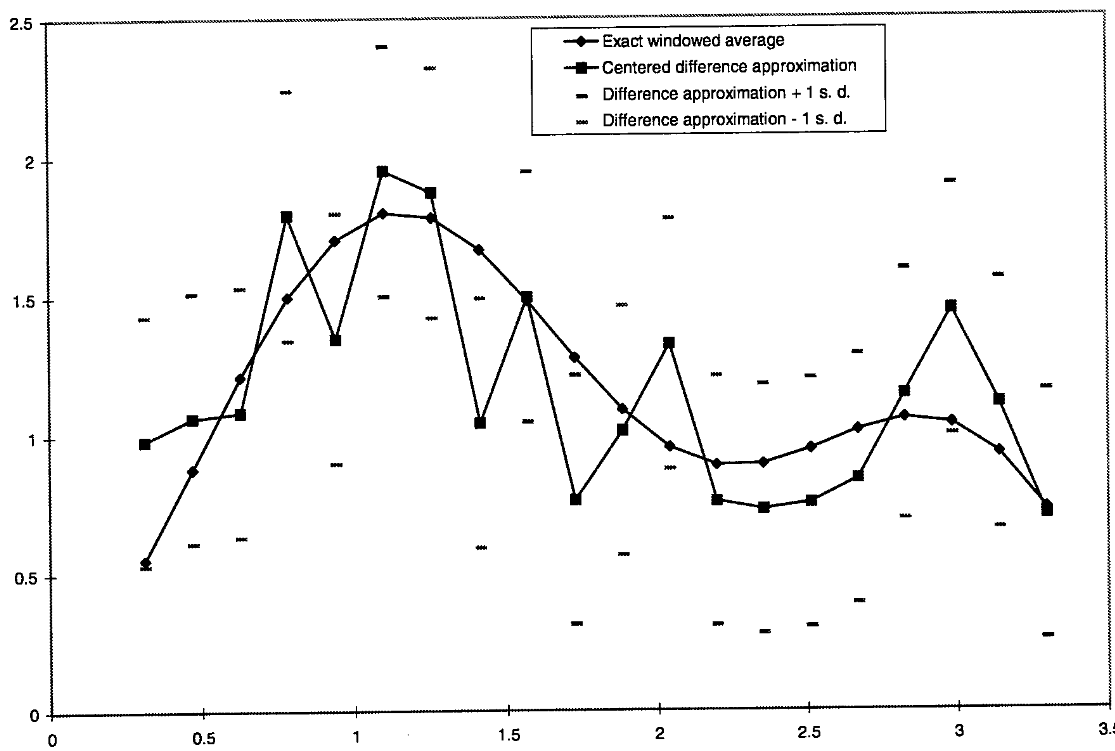
For reference, Figure C-6 shows exact values of the windowed average generated by the difference scheme. We included the reference to help the reader understand the results; the curve wouldn't be available, of course, if the procedure is applied to an actual case.

Figure C–6. Difference Scheme, Largest Spacing



In Figure C-7, the data spacing is reduced to the point at which the error bars suggest that further reduction would generate unreliable results. At this resolution, the bimodal character of the derivative is apparent, although the result is significantly affected by noise.

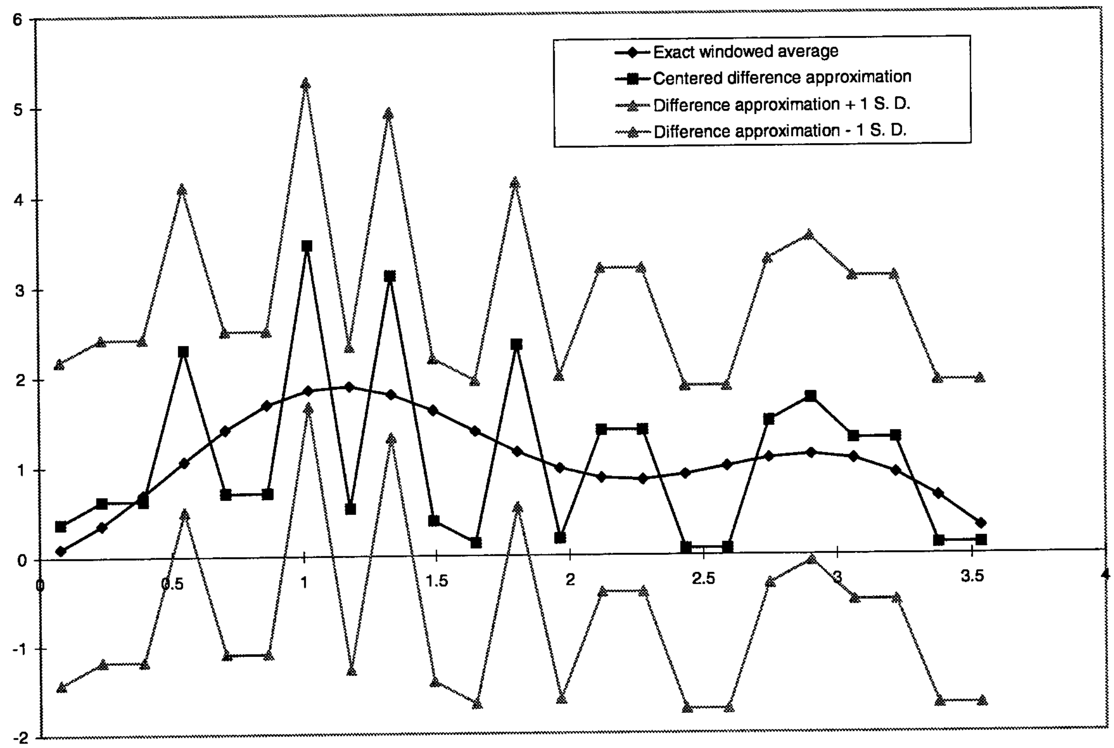
Figure C–7. Difference Scheme, Near-Smallest Useful Spacing



The bandwidth of the centered-difference scheme for this near-optimal spacing, computed from Equation C-7, is 4.1 inverse time units. This implies that features of f′(t) that take place over roughly 2 time units can just be inferred by the scheme when applied to these particular noisy data. This is, of course, consistent with what we have seen: the just-discernible bimodality occurs over that approximate amount of time.

The curves in Figure C-8 show the result of applying the centered-difference scheme at too small a spacing. Here, while the windowed average is capable of representing the salient features of f′(t) quite well, noise overwhelms the result.

Figure C–8. Difference Scheme, Data Spacing Too Small



SPLINE SMOOTHING NUMERICAL DIFFERENTIATION

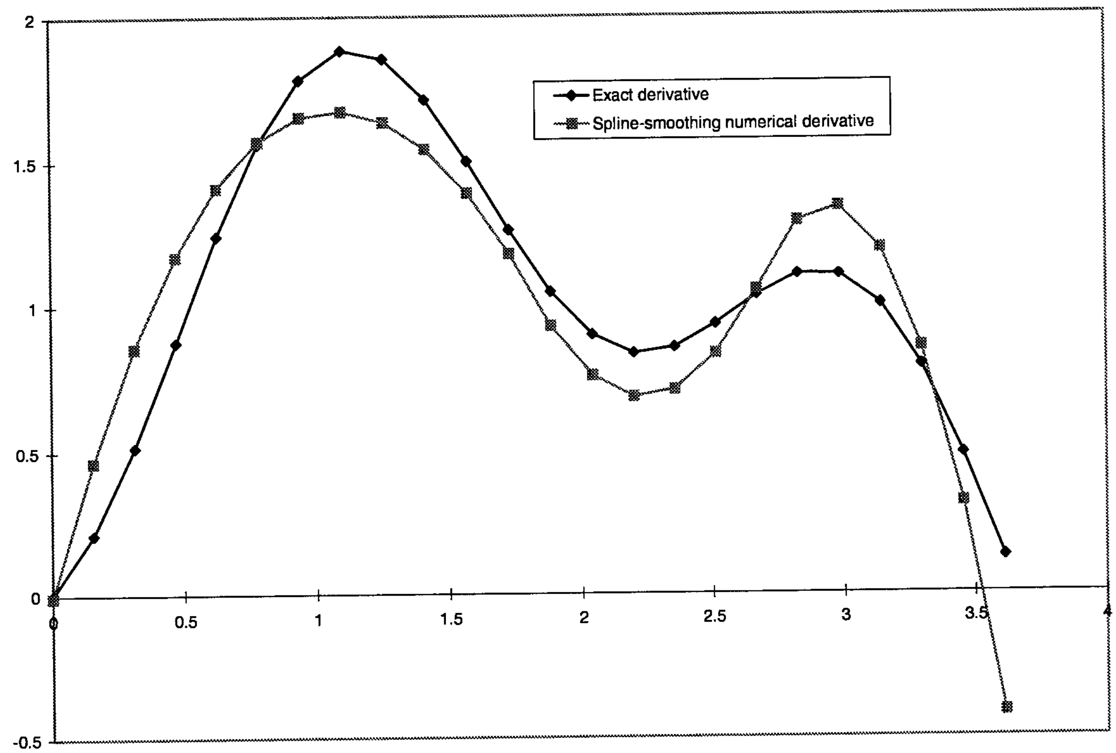
Numerical differentiation may be carried out by fitting the data to some class of differentiable functions, and taking the derivative of the fitted curve as an estimate of the derivative of the data function. This idea underlies several advanced methods of numerical differentiation.[3] One method of this type comes from fitting the data to a polynomial spline[4] and using the derivative of the spline as an estimate of the data function's derivative. We refer to this method as "spline-smoothing differentiation."

SSND uses many data points for estimating the derivative at a single point. By using many data points, we may cancel or reduce several sources of errors. For example, Figure C-9 shows the result of applying SSND to the same noisy data that generated Figure C-5.

[3] Press, W. H. et al, *Numerical Recipes in C*, 2nd Edition, Section 5.7. Cambridge University Press, Cambridge, U.K., 1992

[4] Polynomial splines were discussed in Chapter 4, "Spline Theory."

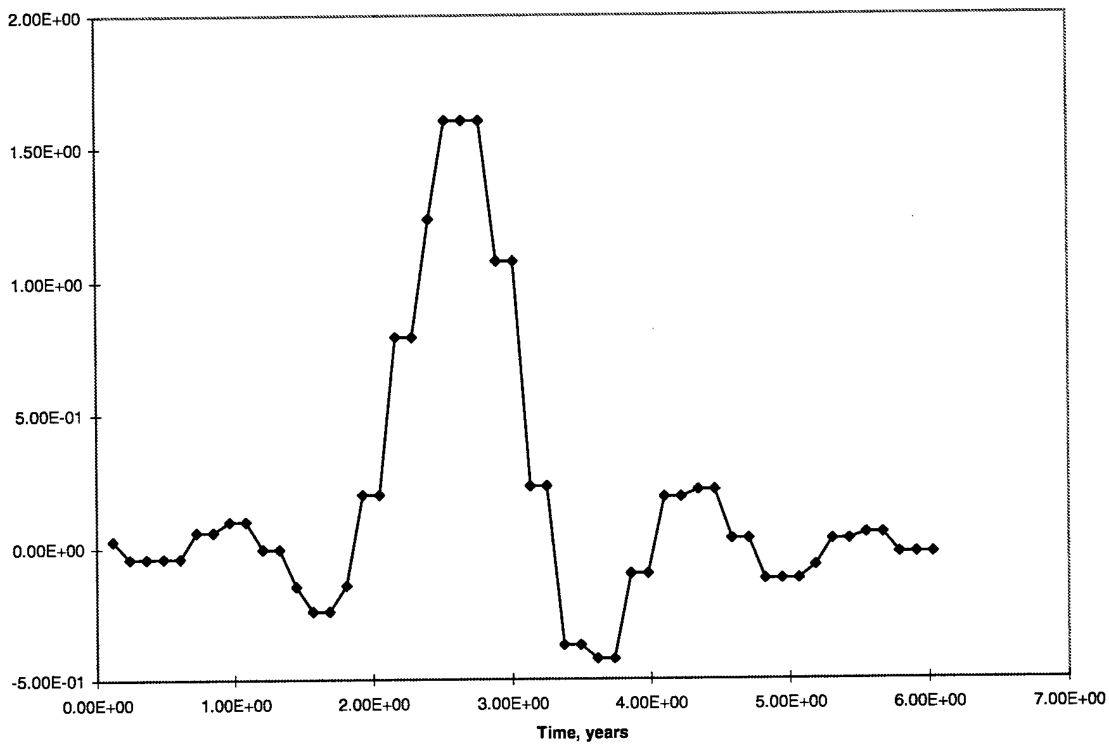
Figure C–9. Values of Exact Derivative and Spline-Smoothing Numerical Derivative



We used SSND to estimate values of the time rate-of-change of earned value, dv/dt. Our method was to find the best least-squares fit to constant-dollar ACWP data of a polynomial spline of a specified order *k*, on an evenly spaced knot sequence that covered the entire interval on which earned-value data were given with a specified number *N* of equal-length intervals. We then took values of the spline's derivative to estimate dv/dt for various t. We refer to these derivative estimates as "N-interval, k^{th} order" SSND estimates.

Like other numerical differentiation schemes, SSND generates a windowed average of the desired derivative. Taking larger values of N, and smaller values of k, makes narrower windows and increases the effects of noise. Figure C-10 shows an example of the windowing kernel for a spline-smoothing numerical differentiation. For that figure, the data times are those of the development program. As we will explain below, the piecewise constant nature of the curve is real, not an artifact. The width of the kernel's main lobe suggests that this scheme can resolve features on time scales of about two years. A different SSND scheme with four intervals and 4th order splines produced the result in Figure C-9.

Figure C–10. Windowing-Kernel for N=8, k = 4, For Derivative at t = 2.524



We are looking for features of dv/dt as a function of v that are common to acquisition programs of a given class. We don't want to neglect features on any time scale (i.e. of any frequency) that we can infer accurately from our data. Thus, we would like to carry out bandwidth-noise sensitivity trades like the one illustrated in figures C-6 through C-9. To do this, we need the analogue of equation C-8 or C-9 for spline-smoothing differentiation, as well as an estimate for the common variance of the noise in the ACWP data. The following considerations provide these things.

The Method In Detail

Our data are *m* noisy values of earned value, $\tilde{v}_i$, and their associated times t_i. We determined minimum-sum-of-squared-residuals fits of polynomial splines of order k, defined on knot sequences that cover the data interval in N equal subintervals, by solving

$$\min_{\{c_j\}} \sum_{i=1}^{m} \left[\tilde{v}_i - \sum_{j=1-k}^{N-1} c_j M_{j,k}(t_i) \right]^2 \qquad \text{[Eq. C-11]}$$

In Equation C-11, $M_{j,k}(.)$ is the basic spline of order k, initial knot j.

The solution of the problem in Equation C-11 is straightforward (readers familiar with the general linear statistical model will recognize Equation C-11 as a particular case of that). If the t_i are monotone, increasing with i, the problem's unique solution is given by

$$c_j = \sum_{p=1-k}^{N} A_{jp}^{-1} b_p \qquad \text{[Eq. C-12]}$$

The matrix elements A_{ij} are given by

$$A_{ij} = \sum_{p=1}^{m} M_{i,k}(t_p) M_{j,k}(t_p) \qquad \text{[Eq. C-13]}$$

and the b_p by

$$b_p = \sum_{i=1}^{m} \tilde{v}_i M_{p,k}(t_i) \qquad \text{[Eq. C-14]}$$

The spline-smoothing estimate of dv/dt at time t is then given by

$$\hat{v}'(t) = \sum_{1-k}^{N-1} c_j M'_{j,k}(t) \qquad \text{[Eq. C-15]}$$

Expected Value

As noted above, the expected value of $\hat{v}'(t)$ is a windowed average of $v'(t)$, and Figure C-10 gives an example of the kernel of that average. The result follows from these considerations:

$$\begin{aligned} < \hat{v}'(t) > &= \sum_{1-k}^{N-1} < c_j > M'_{j,k}(t) \\ &= \sum_{j=1-k}^{N-1} \left(\sum_{p=1-k}^{N-1} A_{jp}^{-1} < b_p > \right) M'_{j,k}(t) \\ &= \sum_{j=1-k}^{N-1} \left(\sum_{p=1-k}^{N-1} A_{jp}^{-1} \sum_{i=1}^{m} v(t_i) M_{p,k}(t_i) \right) M'_{j,k}(t) \end{aligned} \qquad \text{[Eq. C-16]}$$

But

$$v(t_i) = \int_0^{t_i} v'(\tau) d\tau = \int_0^{T} \Omega_i(\tau) v(\tau) d\tau \qquad \text{[Eq. C-17]}$$

where we write T for t_m, and where

$$\Omega_i(\tau) \equiv \begin{cases} 1, 0 \le \tau \le t_i \\ 0, all other \tau \end{cases} \qquad \text{[Eq. C-18]}$$

Substituting the last integral of Equation C-17 for $v(t_i)$ in Equation C-16 and rearranging the sums gives

$$<\hat{v}'(t)>= \int_0^T \left\{ \sum_{j=1-k}^{N-1} \left(\sum_{p=1-k}^{N-1} A_{jp}^{-1} \sum_{i=1}^{m} \Omega_i(\tau) M_{p,k}(t_i) \right) M'_{j,k}(t) \right\} v'(\tau) d\tau \qquad \text{[Eq. C-19]}$$

The right side of Equation C-19 has the form

$$<\hat{v}'(t)>= \int_0^T W(t,\tau) v'(\tau) d\tau \qquad \text{[Eq. C-20]}$$

with the kernel function W(t, τ) given by

$$W(t,\tau)= \sum_{j=1-k}^{N-1} \left(\sum_{p=1-k}^{N-1} A_{jp}^{-1} \sum_{i=1}^{m} \Omega_i(\tau) M_{p,k}(t_i) \right) M'_{j,k}(t) \qquad \text{[Eq. C-21]}$$

In evaluating W(t, τ) we should note that the value of this kernel at (t, τ) is that of the SSND at time t, when the datum at the i^{th} time is the value of $\Omega_i(\tau)$.

Values of all the $\Omega_i(\tau)$ do not change as τ ranges over any (t_i, t_{i+1}). Consequently, for fixed t, W(t, τ) is constant while τ varies over those intervals. This explains the piecewise constant variation of W(t, τ) seen in Figure C-10.

Bandwidth

The windowing operation described by Equation C-20 and Equation C-21 is a low-pass filter because linear combinations of the $M'_{j,k}(t)$ can reproduce a constant function if we take k greater than 2, which we do. Thus, we may determine the filter's bandwidth by finding its upper half-power frequency. We may determine that from the spectrum of an individual $M'_{j,k}(t)$, because the filter's output is a finite linear combination of these functions. That spectrum may be computed exactly. $M'_{j,k}(t)$ is a linear combination of two basic splines of order k-1. Thus, the upper half-power frequency will be the same as that of one of those splines. The work is simplified since we use splines only on evenly spaced knots (the knots for the splines are not necessarily the same as the t_i). Accordingly, we compute the spectrum $g_{j,k}(\omega)$ of $M_{j,k}(t)$, as follows:

$$g_{j,k}(\omega) \equiv \int_{-\infty}^{\infty} e^{-i\omega t} M_{j,k}(t)dt = \int_{t_j}^{t_{j+k}} e^{-i\omega t} M_{j,k}(t)dt \qquad \text{[Eq. C-22]}$$

since $M_{j,k}(t)$ is zero outside $[t_j, t_{j+1}]$. The last integral can be evaluated in closed form; it is[5]

$$\int_{t_j}^{t_{j+k}} e^{-i\omega t} M_{j,k}(t)dt = k!\left[T_j, T_{j+1}, \ldots, T_{j+k}\right] \frac{e^{-i\omega t}}{(-i\omega)^k} \qquad \text{[Eq. C-23]}$$

where $[T_j, T_{j+1}, \ldots, T_{j+k}]$ denotes the k-th order divided-difference operator and the T_n are the knots on which the splines are defined (generally distinct from the t_j for which we have earned value data).

For our cases, the evenly spaced T_j are at times jh, where h is $(t_m - t_1)/N$. Recalling that for evenly-spaced data the divided difference operator takes the form

$$\left[T_j, T_{j+1}, \ldots, T_{j+k}\right] f(t) = \frac{\sum_{n=0}^{k} (-1)^n \binom{k}{n} f(T_n)}{k!h^k} \qquad \text{[Eq. C-24]}$$

we find after some manipulation that

$$\left| g_{j,k}(\omega) \right| = \left(\frac{2(1-\cos(\omega h))}{(\omega h)^2} \right)^{\frac{k}{2}} \qquad \text{[Eq. C-25]}$$

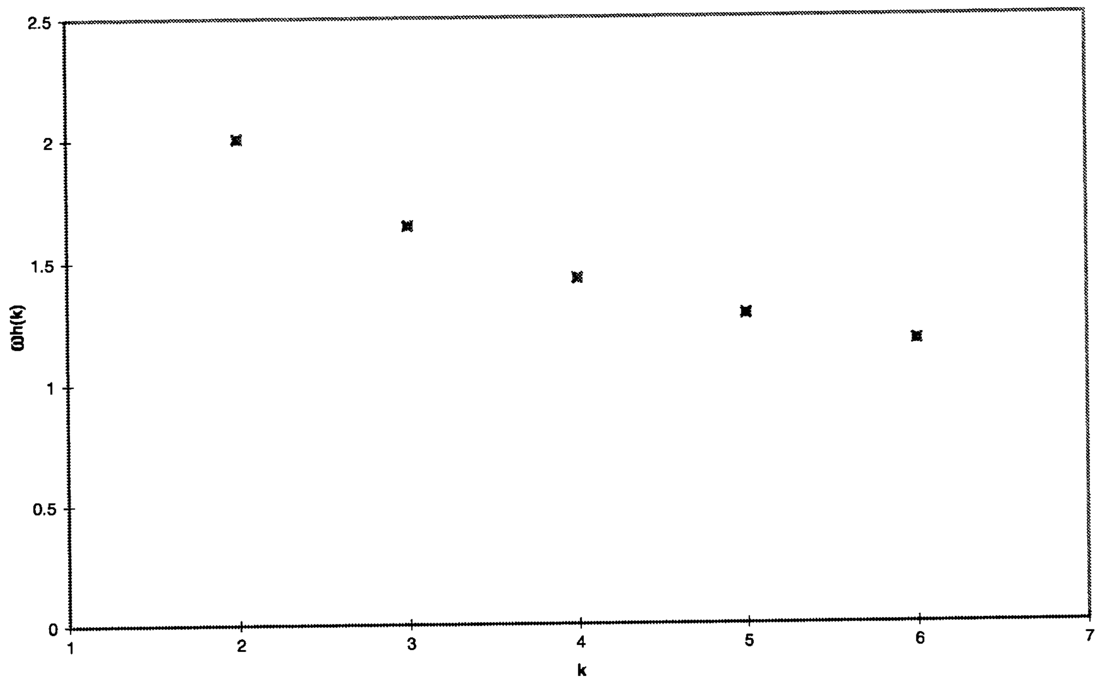
The half-power frequency of this spectrum is at the smallest value of ω for which

$$\frac{2(1-\cos(\omega h))}{(\omega h)^2} = \left(\frac{1}{2} \right)^{\frac{1}{k}} \qquad \text{[Eq. C-26]}$$

Equation C-26 determines the product (ωh) as a slowly decreasing function of k. Figure C-11 shows values for a few values of k.

[5] De Boor, loc. cit. ante, equation (8.1), page 29.

Figure C–11. Variation of ωh with k



With this result, and recalling that for a SSND scheme with N intervals covering the data interval

$$h = \frac{t_m - t_1}{N} \qquad \text{[Eq. C-27]}$$

we have our quantitative measure of the resolution of a (N, k) spline-smoothing numerical differentiation scheme: it is

$$\omega_{\max} = \frac{N\omega h(k-1)}{t_m - t_1} \qquad \text{[Eq. C-28]}$$

Variance

If we let α(t) denote the difference between $\hat{v}'(t)$ and its expected value, it follows directly from the linearity of SSND that

$$\alpha(t) = \sum_{j=1-k}^{N-1} \left(\sum_{p=1-k}^{N-1} A_{jp}^{-1} \sum_{n=1}^{m} \varepsilon_n M_{p,k}(t_n) \right) M'_{j,k}(t) \qquad \text{[Eq. C-29]}$$

where ε_n is the error of the data at t_n. With the usual assumptions that the ε_n are independent normal random variables with common standard deviation σ, we find after considerable manipulation that the variance of $\hat{v}'(t)$, which is the expected value of the square of $\alpha(t)$, is given by

$$\mathrm{var}(\hat{v}'(t)) = <\alpha^2(t)> = \sigma^2 \sum_{s=1-k}^{N-1} \left(\sum_{q=1-k}^{N-1} A_{sq}^{-1} M'_{q,k}(t) \right) M'_{s,k}(t) \qquad \text{[Eq. C-30]}$$

It is a great help in computing $\mathrm{var}(\hat{v}'(t))$ to note that, according to Equations C-30, this quantity is equal to the inferred derivative, evaluated when the b_q are set equal to $M'_{q,k}(t)$.

Inferring Best Practical Resolution From Data

In estimating time rates of change of earned value, we determined best practical resolution for SSND schemes using the procedure that we discussed above in connection with centered-difference numerical differentiation. In this section we explain that method, using data for the airframe development program.

ESTIMATING σ

Values of $\mathrm{var}(\hat{v}'(t))$ from Equation C-30 and of ω_{max} from Equation C-28 give two of the three variables required to find optimal resolution SSND schemes. The remaining requirement is an estimate of σ. Here is our method for estimating.

The basic idea of our estimates of standard deviation is a simple one: For an appropriate (N, k) combination we fit the ACWP data to a polynomial spline, and evaluate the sample variance as the sum of the squares of the residuals, divided by (m - N - k + 1). (This gives an unbiased estimate of the population variance, because the fitting equations impose N - k + 1 linear relations on the basic spline's coefficients.) But what is an appropriate (N, k) combination?

These considerations guided our search for such combinations: If we fit the data with a spline that has too little flexibility, that is, too small a value of N or too large a value of k, then the errors will show serial correlation. (In the limit of the stiffest spline, we would fit the data to a constant, and the errors would trace out values of the data themselves, decremented by that constant.) As N increases, and, perhaps, k decreases from such "too-stiff" values, the residuals will exhibit less serial correlation. As N and k vary to produce very "flexible" splines, serial correlation may again increase. For example, if N and k are chosen so that N - k + 1 = m, the spline will fit exactly, all the errors will be the same number, zero.

With this in mind, we examined serial correlations of the residuals as N and k varied to make the fitted splines more flexible. Typically, we did this for fixed k and increasing N. We found that serial correlations would diminish sharply at first as N increased. We made our estimate of the variance of the ACWP errors from the smallest N that gave small measures of serial correlation. The autocorrelations in Table C-1 illustrate this for the airframe data.

Table C-1. Autocorrelations for Lag 1 and Lag 2, for Varying N; k = 4 In All Cases

N	Lag 1	Lag 2
4	.493	.055
6	.091	-.216
8	-.349	-.330

In view of these results, we used the N = 6, k = 4 case to infer our estimate of σ.

DETERMINING THE OPTIMAL N AND K

Given an estimate of σ, we used Equation C-30 to generate error bars on plots of $<\hat{v}'(t)>$. Our procedure in analyzing v(t) data was, to compute various (N, k) spline smoothing numerical differentiations, increasing N for fixed k until the error bars suggested that features found at the resolution for that N and k was affected too much by noise to make features reliable. We show an example, in Figures C-12 through C-15. In each of these figures, the error bars show estimates of the error in the windowed average of v'(t) generated by the SSND scheme (*not* the total error in the estimate of v'(t)). Each of the figures includes a bar showing the length of the shortest period resolved by that windowed average. The bars show a quantitative measure of the resolution achieved by the SSND scheme.

In Figure C-12, the error bars suggest we determined that the windowed average has been determined rather well, except near the ends of the interval. The "bulge" at the right side is likely to be an actual feature of the data.

Figure C-13 shows the result for a "more flexible" spline than the one in Figure C-12. The larger peak has narrowed. The right-side bulge is still apparent. Error bars suggest that the minimum near the zero time may be an artifact. This is the case from which, guided by Table C-1, we inferred the value of σ used for all the airframe cases.

Figures C-14 and C-15 show the results for fits of splines with increasing flexibility. Perhaps the results of Figure C-14 could be used; their essential features do not differ significantly from those of Figure C-13. For splines as flexible as those of Figure C-15, however, noise effects are sufficiently bad that the details stemming from increased resolution are not likely to be useful. We used the results of Figure C-13 as our estimate of the time rate of change of earned value for the airframe data.

Figure C–12. SSND Result, N – 4, k = 4

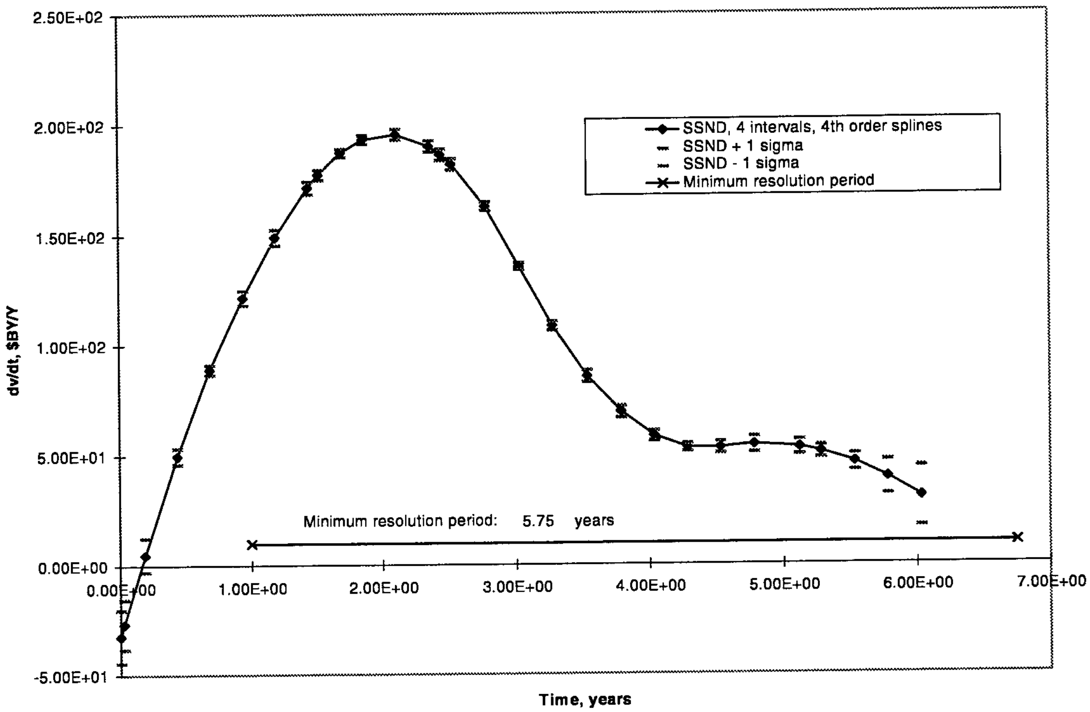


Figure C–13. SSND Result, N = 6, k = 4

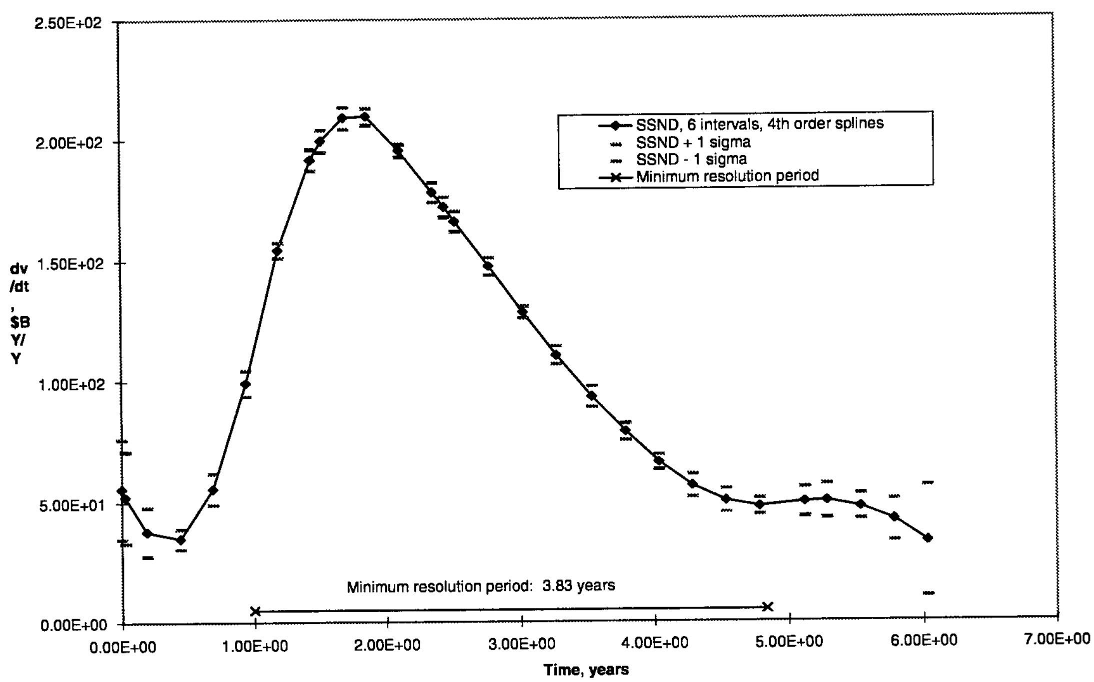


Figure C–14. SSND Result, N = 8, k = 4

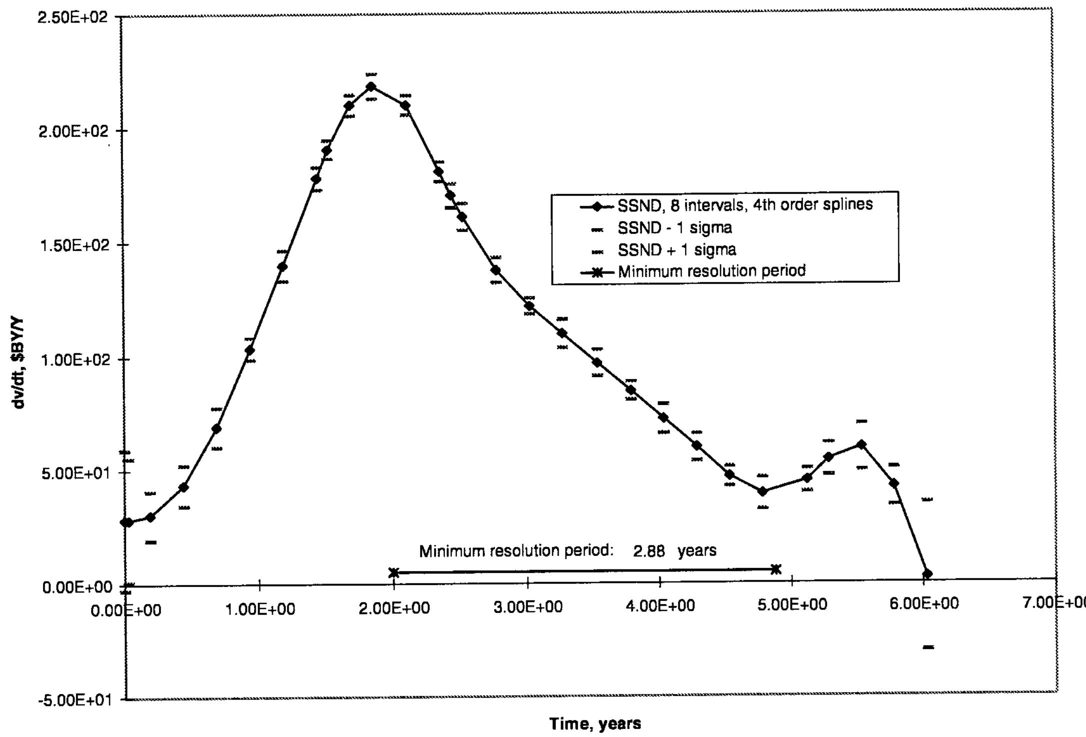
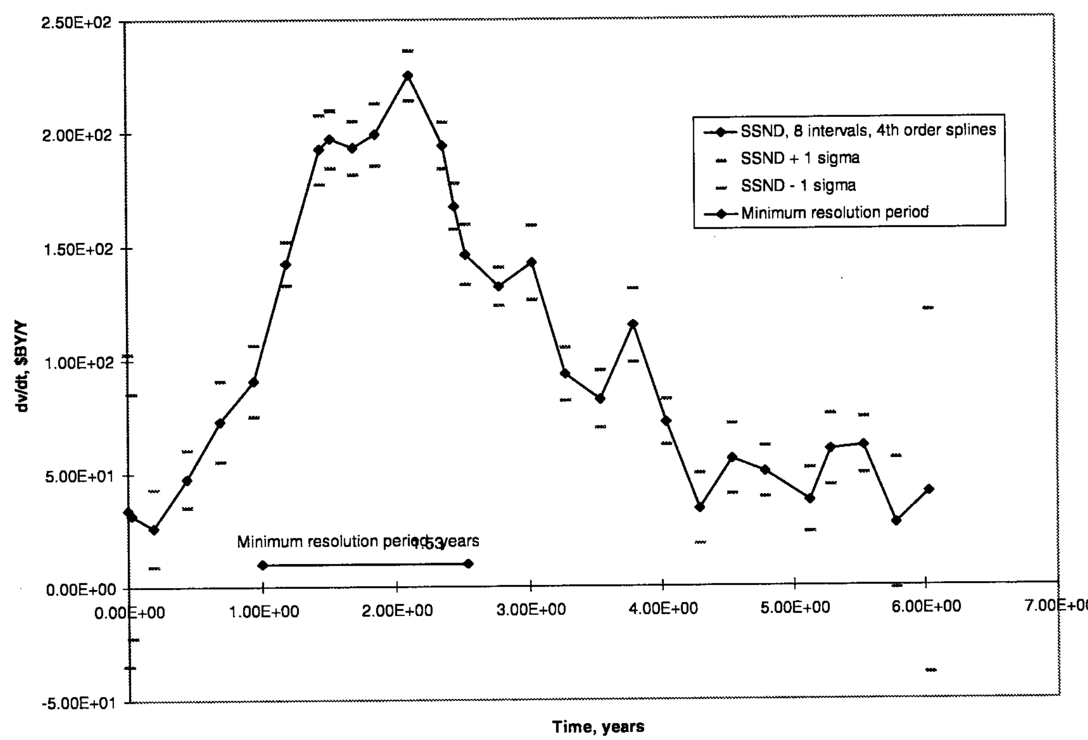


Figure C–15. SSND Result, N = 15, k = 4



REPORT DOCUMENTATION PAGE

Form Approved
OPM No.0704-0188

1. AGENCY USE ONLY ***(Leave Blank)***

2. REPORT DATE

Oct 99

3. REPORT TYPE AND DATES COVERED

Final

4. TITLE AND SUBTITLE

The Rayleigh Analyzer®: Volume I—Theory and Applications

5. FUNDING NUMBERS

C DASW01-99-F-1035

PE 0902198D

6. AUTHOR(S)

John Dukovich
Scott Houser
David A. Lee, Ph.D.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Logistics Management Institute
2000 Corporate Ridge
McLean, VA 22102-7805

8. PERFORMING ORGANIZATION REPORT NUMBER

LMI– AT902C1

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

Office of the Under Secretary of Defense (Acquisition and Technology)
Acquisition Program Integration Directorate
Pentagon
Washington, DC 20301-1800

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

A: Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

The historical expenditures and completion times of many United States defense research and development (R&D) programs have followed the Rayleigh probability distribution. The Rayleigh Analyzer® generates statistics of the completion time and the final cost for development programs, conditioned on actual cost of work performed (ACWP) data and on the Rayleigh model for the time variation of ACWP in development programs. The analysis uses the multiple model adaptive estimation (MMAE) method for system identification.

Volume I of the two-volume series presents Rayleigh theory and applications. A sample of development program offices was interviewed. The interviews served two purposes: 1) confirm the applicability of the Rayleigh model to current programs and 2) adjust the model according to several findings from the interviews, broadening the type of analyses that can be performed. Volume II is a users manual for operating the Rayleigh Analyzer model.

14. SUBJECT TERMS

Cost analysis, schedule analysis, cost estimating, statistical analysis, research and development, project management

15. NUMBER OF PAGES

50

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