COST-EFFECTIVE DESIGN AND OPERATION OF URBAN STORMWATER CONTROL SYSTEMS: DECISION-SUPPORT SOFTWARE

by

John W. Labadie, Neil S. Grigg, Dennis M. Morrow, and David K. Robinson

Colorado Water
Resources Research Institute

Completion Report No. 135

Colorado State University

COST-EFFECTIVE DESIGN AND OPERATION OF URBAN STORMWATER CONTROL SYSTEMS: DECISION-SUPPORT SOFTWARE

by

John W. Labadie
Neil S. Grigg
Dennis M. Morrow
and
David K. Robinson

Department of Civil Engineering
Colorado State University
Fort Collins, Colorado 80523

Research Project Technical Completion Report
Project No. 90037-U
Agreement No. 14-34-0001-0507C

Submitted to:

Office of Water Research
Bureau of Reclamation
United States Department of the Interior
Washington, D.C. 20242

The research on which this report is based was financed in part by the U.S. Department of the Interior as authorized by the Water Research and Development Act of 1978 (P.L. 95-467)

Colorado Water Resources Research Institute
Colorado State University
Fort Collins, Colorado 80523

Norman A. Evans, Director

October 1984

ABSTRACT

Although the microcomputer revolution has made powerful computer hardware available at low cost, there is still a severe lag in the availability of computer software that can aid urban water managers in finding cost-effective solutions to complex design and operational problems in stormwater and combined sewer control. A Stormwater Control Package (SWCP) is presented with user manual for introducing automation into urban stormwater control systems. The package contains state-of-the-art technology in storm inflow forecasting, fully dynamic hydraulic routing, and dynamic programming optimization. It is designed for "simulated" real-time experimentation on application of automation to storm and combined sewer control for achieving improved performance. In addition, it is believed that application of the SWCP at the planning level can potentially save large amounts of capital in sizing of facilities through optimum regulation in real-time of storage and conveyance of stormwater. Computational experience with the North Shore Outfalls and Channel Outfalls Consolidation projects of the Clean Water Program of San Francisco is presented.

In addition to operational software, an optimal sewer design package called CSUDP/SEWER is presented with user manual which also employs dynamic programming. As a screening tool, Program CSUDP/SEWER can find least-cost vertical layouts and sizings of storm drainage systems. Preliminary experience with an optimal horizontal layout procedure which combines nonlinear programming, network flow theory and CSUDP/SEWER is also presented, but more testing with large networks is needed for this algorithm.

The eventual goal is to combine the SWCP and CSUDP/SEWER package together to develop optimal designs with full consideration of the potential cost-effectiveness of employing automation in real-time. Future work should also proceed to apply previous research at CSU on solving large-scale, city wide stormwater control problems by using the SWCP at the subbasin level and optimally integrating it into city-wide controls. Recommendations are included on combining Kalman filtering and dynamic programming together for the stormwater control problem under risk by using a new procedure developed at CSU called dynamic programming in objective space.

ACKNOWLDGEMENTS

The work upon which this report is based was supported in whole by funds provided by the U.S. Department of the Interior, Office of Water Research and Technology, as authorized under the Water Research and Development Act of 1976. The Office of Water Research and Technology has since been terminated effective August 25, 1982, by Secretarial Order 3084 and its programs transferred to other bureaus and offices in the Department of the Interior. The Water Conservation Research Program was among those transferred to the Bureau of Reclamation.

Dr. David K. Robinson provided major contribution to this research with his work on optimal storm sewer system design while serving as Visiting Professor at Colorado State University, during sabbatical leave from the University of New South Wales, Australia. Thanks are extended to Dr. Harry G. Wenzel from the Department of Civil Engineering, University of Illinois at Urbana-Champaign who contributed greatly to preliminary work on the horizontal layout and design problem while also here at CSU as Visiting Professor. The work of several graduate students in Civil Engineering is gratefully acknowledged, including work by Steffen Meyer and Jay Regenstreif on the optimal horizontal layout problem, and contributions by Mark Calabrese on the Stormwater Control Package. We are also grateful to Lee Ann Mitchell for her skillful word-processing work on this manuscript.

DISCLAIMER

This report has been reviewed by the Office of Water Research of the Bureau of Reclamation, U.S. Department of the Interior, and approved for public dissemination. Approval does not signify that the contents necessarily reflect the views and policies of the Department of the Interior, nor does mention of the trade names or commercial products constitute endorsement or recommendation for use.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

A. OVERVIEW

The new emerging CAD/CAM (computer aided design and manufacturing) technology is seen by many as the salvation of American industry in substantially increasing productivity to remain competitive on the world scene. In the public sector, there is also a productivity crisis, but of a different kind. The product is public service, which is of course, difficult to measure in economic terms. However, in construction of wastewater management alone, it has been estimated that over 100 billion will need to be spent before the end of the century to properly develop new systems and upgrade and maintain reasonable service levels of existing systems (Grigg, 1982). Most agree that this level of expenditure is unattainable. Capital intensive solutions to our urban water problems may be a thing of the past. The key to the future may be the full development of a CAD/CAM counterpart in the public works area, with the goal of reducing construction costs while maintaining adequate service levels.

The explosion in sophisticated and powerful computer hardware technology and its availability at reasonable cost is perhaps one of the greatest American success stories of this century. Developments in applications software are also dramatic, but lag behind hardware development. In particular, adequate software is still not available in a readily usable form for dealing with complex problems associated with metropolitan water services. Back in 1970, the American Public Works

Association surveyed the state of the art in public works computer applications, including guidelines for installation and operation of computerized process control systems (APWA, 1970). Poertner (1972) followed this with a survey of "Existing Automation, Control and Intelligence Systems for Metropolitan Water Facilities." Since that time, further progress in implementation has been modest at best. With the major advances in computer technology that have occurred since these reports, it is imperative that these surveys be updated with particular focus on developing productivity and cost-effectiveness data, including surveys of available models, software, and specification of future software needs.

McPherson (1971), perhaps the pioneer in introducing innovative concepts of computerized automation and control in the urban water services, has emphasized that widespread development and application of computer technology can only be justified to the extent that all the water-related metropolitan services are integrated together using networked computer hardware and data management systems, as well as integrative software that can exploit the overlap between the various services.

B. RESEARCH OBJECTIVES

With the ultimate need for these integrative approaches clearly in mind, the focus in this research is on development of general computer software for optimal design and real-time operation of urban flood control, including combined sewer systems. The urban flood control problem is of course closely related to the problem of pollution of receiving waters. It is believed that inclusion of computerized control concepts into planning and design can result in considerable capital investment savings for urban drainage and flood control systems. It may be possible, for example, to reduce sizing requirements for stormwater conveyance, storage, treatment and pumping facilities through recognition that storm inputs are generally spatially and temporally nonhomogeneous over an urban area. Portions of a combined sewer system may be overloaded, whereas at the same time, other areas may have sizable unused capacity. In addition, such factors as storm movement in relation to the direction of sewer flow and time lags between storm input and ultimate passage to interceptors can be exploited (McPherson, 1981). Through investment in such technology as radar and storm tracking, advanced warning systems, automated raingage networks, forecasting models, sewer system models, flow level sensors, automated regulators and control devices, and control decision software, it may be possible to more effectively utilize the total storage capacity of a storm or combined sewer system. Metropolitan Seattle's 3.1 million CATAD (Computer Augmented Treatment and Disposal) system has been reported as saving the City 70 million in construction costs for an alternative separated sewer system (Brown and Caldwell, 1978). They reported that CATAD had reduced stormwater overflows by 80% since its 1973 installation by simply more effectively using in-line storage capability through computer controlled regulators.

Previous work by Labadie et al. (1975) and Trotta et al. (1977) has shown that urban runoff control is most effectively accomplished on a totally integrated, city-wide basis. Labadie et al. (1980) have emphasized the importance of including fully dynamic hydraulic system modeling in computer control schemes. Trotta et al. (1977) have demonstrated the importance of including rainfall forecasts and anticipated storm flows into real-time decisionmaking which considers the inherent stochasticity of the control problem and the risks associated with control decisions. Morrow and Labadie (1980) have attempted to demonstrate how these elements can be integrated together through a software development called the Stormwater Control Package that can be used for operational planning as well as possibly adapted for actual real-time control. Labadie et al. (1981) have gone a step further and actually documented the value or "worth" of real-time storm inflow forecasting in computer control systems. It was found that attempts to forecast can be valuable, even in the presence of somewhat large forecast errors. This latter work was supported under the research covered by this completion report. Though the City of San Francisco has served as the primary case study for most of this research, attempts have been made to generalize the software development for use in other cities.

What this research has shown is that real-time computer control and decision support systems, even in the presence of prediction uncertainties, can enhance the performance of urban drainage and flood control systems. Again, this has implications both in an operational and a planning content. At the operational level, it means we are maximizing performance of existing facilities, and hence improving cost effectiveness. In a facilities planning content, it means that incorporation of these control concepts into capital improvement alternatives can possibly replace sizing requirements.

In addition to work in computer control and real-time operations, considerable effort has been carried out at Colorado State University on optimal sizing and design of urban stormwater drainage facilities in order to achieve cost effectiveness. As an early output of the research covered in this completion report, Robinson and Labadie (1981) presented a dynamic programing algorithm for optimal vertical layout and sizing of a storm sewer system. This model is designed to be an effective screening process whereby optimal pipe selections and vertical layout under steady flow assumptions can be followed by further refinement and, hopefully, sizing reductions through use of the Stormwater Control Package using dynamic, unsteady flow routing and "simulated" real-time forecasting and control.

Considerable work resulting from this research has been previously published and is readily available. The purpose of this final completion report is to:

1. Provide a concise overview of the Stormwater Control Package (SWCP) developed at Colorado State University.

2. Present previously unpublished results of application of the SWCP to the San Francisco Stormwater Control System using the Bayside Facilities Planning Project as a case study.

3. Include new documentation and a user's manual for the general SWCP software to facilitate its possible use in other cities.

4. Present a new procedure for integrating the problem of optimal vertical and horizontal layout of a storm or combined sewer system, along with some computational experience. Again this is a screening procedure which can provide layouts and sizings that can be further refined by the stormwater control package, with possible inclusion of automatic control capability and considuation of realistic hydraulic routing. This technology is not only applicable to new system, but also for finding optimal expansion and improvement plans for existing systems.

5. Provide an updated user's manual for software developed under this project for optimal vertical layout and sizing of a storm drainage system.

6. Propose a new procedure for fully incorporating risk and uncertainty into the operational aspects of storm drainage systems.

As mentioned previously, the early work of this research focused on the Stormwater Control Package and the value or "worth" of real-time storm inflow forecasts and has been published. A copy of the paper is presented in Appendix A of this report, with a user manual for the SWCP included in Appendix B. Due to changes in budgetary priorities for the City of San Francisco, it has not been possible for Colorado State University to actually see the implementation of the methodologies developed in this research to the San Francisco stormwater management system, as was originally proposed in this research. As a result of this, it was decided that it would be beneficial to attempt to generalize both the operational control packages and sewer design and layout programs so as to be applicable to other urban areas. Published work resulting from this research on optimal vertical layout and sizing of storm drainage systems is included in Appendix C. The user manual for this program is found in Appendix D. Though full demonstration of these tools on an actual city-wide basis is yet to be achieved, it is believed this research has generated the basic operational and design tools necessary for such an effort. It is hoped that future opportunities will arise for a full demostration of this software, if not in San Francisco, then perhaps in another U.S. city or even abroad. Information on access to the computer codes developed in this research can be obtained directly from the principal investigator, Dr. John W. Labadie.

An additional goal of this research was to provide a conference and workshop for transferring this technology to other cities. Dr. Labadie is co-chairman of a large 2-day symposium on "Computer Aided Decision Support Systems in Water Resources," at the 1985 Spring National

Convention of the American Society of Civil Engineers (ASCE) in Denver, Colorado. He has also been asked to present an extensive tutorial on "Real-Time Control of Water Resource Systems," with focus on urban water systems, at the 1985 ASCE Specialty Conference in Buffalo, New York, with the theme of "Computer Applications in Water Resources."

These sessions should be well attended by managerial and engineering staff from many large and small municipalities around the U.S., and will afford an excellent opportunity for presenting the results of our research.

C. ORGANIZATION OF REPORT

This completion report is organized as follows: the following chapter focuses on cost-effectiveness analyses in urban water systems in general, and stormwater control systems in particular. It is shown that inclusion of computer-aided decisionmaking and automatic control are critical needs for achieving cost-effectiveness in urban flood control and wastewater management, and therefore enhancing productivity in urban water services. This sets the motivation for the following chapters which present the operational and design software which we believe can help meet these needs. A final chapter is devoted to a proposed methodology for performing integrated real-time control with consideration of risk and uncertainty. The Appendices provide foundational work associated with this research, along with user manuals for the computer software.

CHAPTER II

COST-EFFECTIVENESS IN URBAN WATER SYSTEMS VIA COMPUTERIZED DESIGN AND CONTROL

A. COMPUTERIZATION IN PUBLIC WORKS: YESTERDAY AND TODAY

Colorado State University published a report in 1972 with a description and inventory of existing computer control and automation systems at that time (Poertner, 1972). This followed a thorough review by Murray McPherson and the ASCE Urban Water Research Program of the concept of "Metropolitan Water Intelligence Systems" (McPherson, 1971). The contention was that computerized automation in urban water services is absolutely essential to achieving desired levels of cost-effectiveness and productivity, and that this technology in turn needs to be fully integrated into urban water system planning and design. It is valuable to go back to some of the municipalities surveyed in 1971 and 1972 by Poertner and find out (a) what their experiences have been in applying computer technology since then, (b) if indeed there have been productivity improvements, and (c) to determine what some of the current trends in the field are. The focus of this survey is on operational aspects rather than planning and design since we are at this time unaware of any attempt to directly include consideration of automatic computer control as an important element of facilities planning and capital improvement.

Although urban water systems can be lumped together for the purpose of general description, there are substantial differences between water supply, wastewater and drainage systems. Integration of these systems can result in scale economies, but fragmentation of systems rather than integration remains the rule. The range over which computer-aided decision support can occur includes all of the components of water supply, wastewater, drainage and flood control systems, and all of the stages of computer control and automation: data collection, supervisory control, partial automation and total automatic control.

A.1 The Setting in 1971

Ten years ago we had computers, automated instrumentation and adequate control equipment to implement control systems. Hopes were high for the transfer of NASA technology to the environmental sector, including urban water systems.

The CSU project published a report on computer and control equipment in 1971 which concluded that "...costs associated with automation are quite variable, being dependent on a number of factors. A quite approximate basis for estimating automation costs is $800 per control loop, but each situation is best evaluated on an individual basis. The associated computer system will generally be a minicomputer selling for less than $50,000" (Medearis, 1971).

On the face of it, the situation in 1971 seems not unlike what we find today; minicomputers were available then and costs per control loop had already fallen to presumably reasonable levels. The stage was set for progress in applying this technology to urban water facilities, along with other process systems in industry and in utilities.

The biggest change since 1971 has been, of course, the remarkable development of the microprocessor. This technological development may have more to do with trends in computer control and decision support than almost any other single event.

A.2 Specific Applications in 1971

Poertner surveyed over 60 utilities in 1971 to learn of their experiences. Several have been selected from each of the water services to see what has happened since then.

1. Water Supply

The Denver Board of Water Commissioners reported in 1971 that they had completed a "power conservation phase" and were starting a "load shifting phase" for their water supply system. A third phase, called the "total demand phase," would seek to provide in real-time a complete operating picture of the system. The next phase would couple the computerized operation of the filter plants with the distribution system and raw water supplies, with all of this to be operated under computer control with appropriate software.

The Philadelphia Water Department was engaged in water quality monitoring as well as control of the water distribution system through a "Load Control Center." At this center, they were remotely monitoring 145 points in the system with the capability to exercise supervisory control over pumps and control valves. They were in the third phase of automating the Load Control Center and information was being processed by a minicomputer. They had decided to automate their water treatment plants and envisioned that this would be completed within three years (by about 1975-1976). They were considering utilizing distributed processing in the three water treatment plants. It should be noted that Philadelphia Water Commissioner Sam Baxter and his research chief, Joe Radzuil, both have died since their 1971 response on which the above information was based.

The Dallas Water Utilities Department was considering improvements in water system control in 1971. Their consultants had completed a study which recommended expanding the existing control center to include data handling and supervisory control equipment and to begin a study of specific data needs from each control location. They already had an off-line simulation program for the water distribution system in regular use as well as a simulation model for the raw water supply system.

2. Wastewater Treatment

The City of Atlanta reported in 1971 that their R.M. Clayton plant was to feature digital computer control connected to analog control loops. They intended to use the system for conventional types of process control and later for optimizing plant performance through process modeling. The City had a contract with Fisher and Porter for the instrumentation.

The City of Los Angeles was planning a high degree of process instrumentation and central automatic control for three new wastewater treatment plants. Direct digital control and data logging were to be handled by a computer which would optimize the operating parameters and provide data on effluent quality for the regulatory agency.

The City of Milwaukee had future plans for the use of a central control computer, minicomputers and data logging equipment at their South Shore plant where improvements were scheduled for completion in 1972. Expansion plans for the South Shore plant would increase its capacity from 60 to 120 mgd and upgrade it from primary to secondary treatment.

3. Combined Sewer Overflow Control

In 1971, the Municipality of Metropolitan Seattle already had in operation a computer augmented treatment and disposal system (CATAD) which sought to fully utilize in-line storage available in the combined sewer system to reduce overflows. They were developing a mathematical model of the collection system to allow for eventual full control by computer. At that time, the system was being operated under supervisory control with provisions for both local and remote control built into the system.

Minneapolis-St. Paul also had a computer controlled system for maximizing the use of in-line storage in a large interceptor sewer. Inflatable "fabridams" were in use to control sewer flows rather than standard gates. They were estimating that fully automated control could be attempted at that time (U.S. EPA, 1974).

San Francisco began developing an ambitious master plan for wet-weather control of overflows in the 1960's. A computer-based data logger was installed to collect precipitation and combined sewer flow data across the city. Plans were made for computer control of the system after the master plan concept was implemented. They envisioned off-line storage using a series of detention tanks, combined with a cross-town tunnel and expanded treatment of wet weather flows. These original plans went through considerable alteration in the late 1970's and early 1980's, with eventual abandonment of the detention storage concept and replacement with large shore-line interceptor tunnels. Regulatory standards have also released considerably since the early days of the Master Plan.

A.3 The Current Picture

Several of the organizations described above were contacted by phone, letter or visit by Dr. Neil S. Grigg, co-investigator for this project. The survey was by no means exhaustive. By comparing his findings with observations from the literature, it was hoped to discern some trends and an idea about the "state of the art."

The Denver Water Board has maintained a consistent interest in automation and computer control of their water treatment and distribution facilities. They currently have remote supervisory control of all treatment plants and pump stations. There are no sophisticated computer simulation or optimizing models in use for operating the systems but the Process Control Section appreciates the potential for them and will consider these advances in the future. Practical problems of control such as equipment maintenance and reliability, cost, and employee reaction continue to be large factors in decisions about computerization. (1)

The City of Philadelphia Water Department reported a continued interest in automation. They have contracts completed for automation of two filter plants. Controls will be distributed and based on microprocessors. They anticipate completion of the projects in 1985. (2)

Dallas continues their interest in computers and automation but has not implemented closed-loop automatic control with use of prediction models. They do have some direct digital control loops in their treatment plants. They continue their interest in off-line analysis by modeling and the application of scientific techniques to water system operation. (3)

An in-depth analysis of the status of computer control and automation in wastewater treatment plants has not been made. A paper about the Milwaukee system seems to represent successful applications, how-

---

(1) Based on a visit with Mr. Richard J. Fellows, Superintendent of Process Control, Denver Water Board.

(2) Letter from Patrick Cairo, General Manager, Planning and Engineering Division, Philadelphia Water Department.

(3) Based on a telephone conversation with Mr. Dan Brock, Dallas Water Utility.

ever, and some trends can be discerned from it (Dedinsky et al., 1982).

Milwaukee completed the installation of two 16-bit minicomputers at its South Shore Water Treatment Plant in 1977. As of this date, they have nine control loops in operation, ranging from raw sludge pumpage and dissolved oxygen control to sampler flow pacing. They report consistent improvement of plant performance without increased manpower. Observations about necessary conditions for successful computerization include: effective instrumentation, coordination of data transmission with instrumentation, simple control loops (at least initially) and good definition of control set-points.

The situation with combined sewer overflow (CSO) control is, as always, the most complex. In spite of the enthusiasm with which some agencies approached CSO in the 1970's, the lack of federal funds combined with a relaxation of regulatory tensions has resulted in little progress with automation and computer control.

The best success story encountered so far continues to be Seattle. They were known in the 1970's for their innovative CSO approach and their story was highlighted in many reports and papers as well as at least one film. They currently continue to operate the CSO system with computer control of 20 key regulator stations. The control uses the rule curve approach. Work on more sophisticated prediction models has been halted and they currently see no advantage in the more complex models. At the present time, they are considering upgrading their control hardware, but financial analyses are still underway and federal support has not been secured. One important factor in Seattle's success was the vision and drive of the Executive Director when the system was being implemented. (4)

No reports appear to be available on the use of models to control other CSO systems such as Minneapolis-St. Paul. San Francisco has not yet moved head with its plans for computer control and automation as of this date. It seems safe to say that any actual use of prediction models to operate CSO systems is still in the future.

A.4 Some Trends

The picture that we are left with is that progress in computerized water system control during the last decade has been slow compared to the expectations we had in 1971. Implementation of computer control and decision support is proceeding incrementally, but nowhere do we find a utility that has thrust forward with a comprehensive effort.

In these times we are hearing about use of automation in other technological fields. Terms such as CAD/CAM (computer-assisted design and manufacturing) are seen everywhere. A review of news articles and periodicals suggests that computer control and automation has even been slow in some industrial applications as well and that CAD/CAM has a long way to go before we reach the so-called "Automated Factory."

A review of the status of automation in water agencies and in industry suggests that problems can be combined into the following three categories:

1. Management Support and Incentives. There is a lack of top management support for innovation and automation. Incentives are weak. Why conserve water and energy in a non-regulated public utility? Why do

---

(4) Based on a telephone conversation with Mr. Bill Nitz, Municipality of Metropolitan Seattle.

more than the minimum in a wastewater treatment plant? Why control CSO if there is no penalty?

2. Human Factors. Automation threatens jobs. What is the incentive to operators? Where are the skilled and trained operators coming from?

3. Technological. Although we have heard much about the explosion in computers there are still many technological barriers to computer control and automation. They include non-standardized data systems, inadequate or overly expensive instrumentation, and lack of good software. It is this latter area that we have particularly addressed in this research.

A.5 Conclusion

In looking back over progress during the past ten years, it is surprising not to see more. However, an analysis of the reasons reveals trends that also apply to industrial sectors other than water. The prescription for improvement in the future seems clear: effective management, responsible regulation, more research and development and a good overall climate for technological innovation. The water industry offers a good market for consultants and firms with promising approaches to computer control and automation.

B. COST-EFFECTIVENESS IN URBAN STORMWATER MANAGEMENT

In effect, cost-effectiveness analysis is a creative technique used to determine the relative value, in economic terms, of alternative problem solving approaches. In systems analysis, the use of economic data, simulation models and optimization allow examination of many aspects of system performance. The resulting cost-effectiveness analysis is somewhat a work of art rather than a straightforward application of economic principles. In a problem as complex as the performance of an automated combined sewer control system, cost-effectiveness analysis will necessarily be complex.

The basic steps in cost-effectiveness analysis were presented by Zazanowski (1968) and they still apply today. The ten steps suggested by that source are repeated here:

1. Define the desired goals, objectives, missions, or purposes that the systems are to meet or fulfill.
2. Identify the mission requirements essential for the attainment of the desired goals.
3. Develop alternative system concepts for accomplishing the missions.
4. Establish system evaluation criteria (measures) that relate system capabilities to the mission requirements.
5. Select fixed-cost or fixed-effectiveness approaches.
6. Determine capabilities of the alternative systems in terms of evaluation criteria.
7. Generate systems-versus-criteria arrays.
8. Analyze merits of alternative systems.
9. Perform sensitivity analyses.
10. Document the rationale, assumptions, and analyses underlying the previous nine steps.

From the steps listed, it is apparent that the biggest challenges to us are the determination of the merit of the alternatives, along with sensitivity analysis to determine the reliability of the conclusions. A significant challenge of cost-effectiveness analysis is how to integrate the many real world considerations into the analysis.

On the face of it, cost-effectiveness analysis should be based on minimizing the cost of maintaining fixed standards. In other words, for a given array of environmental standards, what is the minimum cost solution, or how much capital and operating cost can be saved with automatic computer control and decision support? As difficult as these questions are analytically, additional considerations must be included in the analysis such as reliability, redundancy, political cost of failure and regulatory ambivalence, in order to guarantee credibility of the study.

1. Regulatory uncertainty. In the early 1970's, there was a growing awareness of the "moving target" of regulatory agency changes. This has not really improved; in fact, it has likely worsened. With the Federal Government generally withdrawing from the construction grants picture, we will see somewhat relaxed standard, under state primacy. This is consistent with advice given San Francisco and EPA as to the value of the large San Francisco city-wide program, as provided by a panel of experts (Pirnie, 1980). What might occur is a tendency for the regulatory agencies to adapt lower standards through permits and agreements in order to be consistent with the technology levels available to cities such as San Francisco, with little planned implementation automatic computer control capability.

It is impossible to forecast shifts in regulatory policy in cost-effectiveness studies, and it would be counterproductive to try. It is probably best to fix a set or range of standards and provide an analysis based on those standards.

2. Technological risk. One of the reasons for failure to implement computer control so far is technological risk. In the past, this has been a source of embarrassment to the space program, transit programs, and other areas where high technology has been used. A critic might say that a cost-effectiveness analysis is not valid without building in the risk of failure. This is difficult to do since we lack

reliability data, and we are dealing with an area where development is still proceeding rapidly. This should probably be handled by development of cost-effective configurations under various assumed reliability levels.

3. Adequacy of cost estimates. One of the obvious reasons for the slow implementation of computer control is the fear of escalated costs in an industry that is conservative by nature. The literature contains many references to the inadequacy of the capital base of the water industry and the rate structure to generate capital. Thus, investment strategies tend to be ultra-conservative. On the other hand, we have little actual experience with the true and total costs of automatic control and decision support systems for public works.

4. Operator skill levels. What is the real capability of a municipality to mount a computerized decision support or automated control strategy when considering difficulties in maintaining highly skilled staff. This question must be approached from an economic standpoint: if the benefits are worth the investment in the first place, they will be worth the investment in staff.

C. A RECOMMENDED COST-EFFECTIVENESS STUDY

The goal of a viable cost-effectiveness study should be to demonstrate the value of computerized decision support and automatic control. Estimates of the value of sophisticated simulation, optimization, and forecasting models should also be derived from the basic analysis. The software tools developed as a result of this research can provide a good foundation for such a study.

The first step should be to isolate the part of the system to be studied. Since there are often shifts in the design of a city-wide system, there are unresolved questions about performance objectives. It seems best then to take a subsystem which is already established.

The next step is apply operational and design software such as presented herein and insure they are functional and properly calibrated. When this is achieved, we can be confident that we can simulate and evaluate many situations for a cost-effectiveness study and sensitivity analysis.

The presumed output from using this software to evaluate various design and control options will be performance data with and without control under various levels of forecast accuracy. Control scenarios and assumptions about control capabilities must be derived from a sound knowledge of the real system, along with any future possible control implementations. To facilitate the development of control scenarios it is critical that analysts work closely with agency personnel in order to jointly develop the scenarios to be tested. It is only after the physical system is properly identified and understood that the control schemes and design features can be developed.

At this point, an appropriate definition of cost-effectiveness must be agreed upon. Two obvious candidates for effectiveness criteria are: savings in capital cost and savings in O&M cost under fixed regulatory standards.

The more dramatic argument is in presumed capital savings due to automated control and decision support. The ultimate goal is to be able to suggest that a smaller facility, at lower cost, could meet the standards with use of automated control and optimal design or that expansion can be delayed. This approach has the difficulty that: 1) it can be criticized as based on unproven technology; and 2) that it implies criticism that the facility was originally planned to be too large. Research designs are needed that can mitigate both of these arguments.

Perhaps the best approach is to base the analysis on the planned size of the system and present results in two forms: first, improved performance due to automated control, and second, on capital savings to meet higher standards. For the latter, it is necessary to approximate cost estimates for the system at various levels of higher capacity.

D. CONCLUSION

In conclusion, the software tools developed as a result of this research can be effectively employed to improve cost-effectiveness and productivity in urban flood control and pollution statement. We have attempted to show a strong linkage between automation and computerized decision support systems with achieving high levels of cost-effectiveness. A procedure has been suggested for effectively demonstrating the value of this technology and quantifying that value in economic terms. The following chapters present this technology in such a way that, hopefully, it is immediately useful for such a demonstration.

CHAPTER III

STORMWATER CONTROL PACKAGE FOR OPERATIONAL COST-EFFECTIVENESS

A. INTRODUCTION

A Stormwater Control Package (SWCP) is presented in this chapter as a methodology for application of realistic computer models for automatic sewer storage regulation in "simulated" real-time for operational planning purposes. It is believed that with the powerful, low cost microprocessors currently available, the model could be adapted to actual real-time use. We maintain that sophisticated modeling techniques can indeed be implemented for real-world problems and can offer in many cases improved cost-effectiveness. Individual components of the SWCP can and should be upgraded and improved in future work. The uniqueness at the SWCP lies in the integration of these components into a workable package.

What follows is a general overview of the SWCP, followed by a more detailed look at the forecasting, routing, and optimization components. A summary of important results obtained using the North Shore Outfalls consolidation project of San Francisco as a case study is also included. Appendix A provides work conducted under this project on use of the SWCP for evaluating the "worth" of storm inflow forecasts for automatic control, and the sensitivity of that value to forecast error. Appendix B contains user information for the SWCP showing data input requirements, formatting, and sample data output.

B. STORMWATER CONTROL PACKAGE: OVERVIEW

The Stormwater Control Package (SWCP) incorporates state-of-the-art techniques in (1) inflow forecasting, (2) unsteady flow sewer routing, and (3) multidimensional dynamic programming optimization for controlling storm sewer flows. These three models and their lines of interaction are illustrated in Figure III-1. These models are briefly summarized, followed by more detailed discussion in the following sections.

B.1 Real-Time Forecasting Model

Inflow forecasting capability for storm or combined sewer systems enables anticipation of peak flows and storage requirements and therefore allows a certain lead-time in order to actuate control strategies such as starting up wet-weather pumps and controlling gates. The runoff forecasting model available in the SWCP is based on Box-Jenkins statistical concepts; that is, pattern recognition of inflow data by time series regression analysis techniques (Box and Jenkins, 1976). This autoregressive-transfer function model was adapted from a stand-alone model developed by Trotta (1976).

The model is composed of two parts: (1) off-line historical parameter generation and (2) on-line real-time forecasting, based on historical parameters and measured data on the storm in-progress. The former component assumes that relevant historical rainfall data have been passed through a watershed model which has been properly calibrated for basins with trunk sewers leading into the interceptor(s). It is the resulting inflow data which are directly used in the forecasting model. Future research on the SWCP will attempt to include a watershed model directly in the SWCP. The second component allows model parameters to be updated automatically as the real event unfolds in real-time or "simulated" real-time. A lead time option for the forecasting model

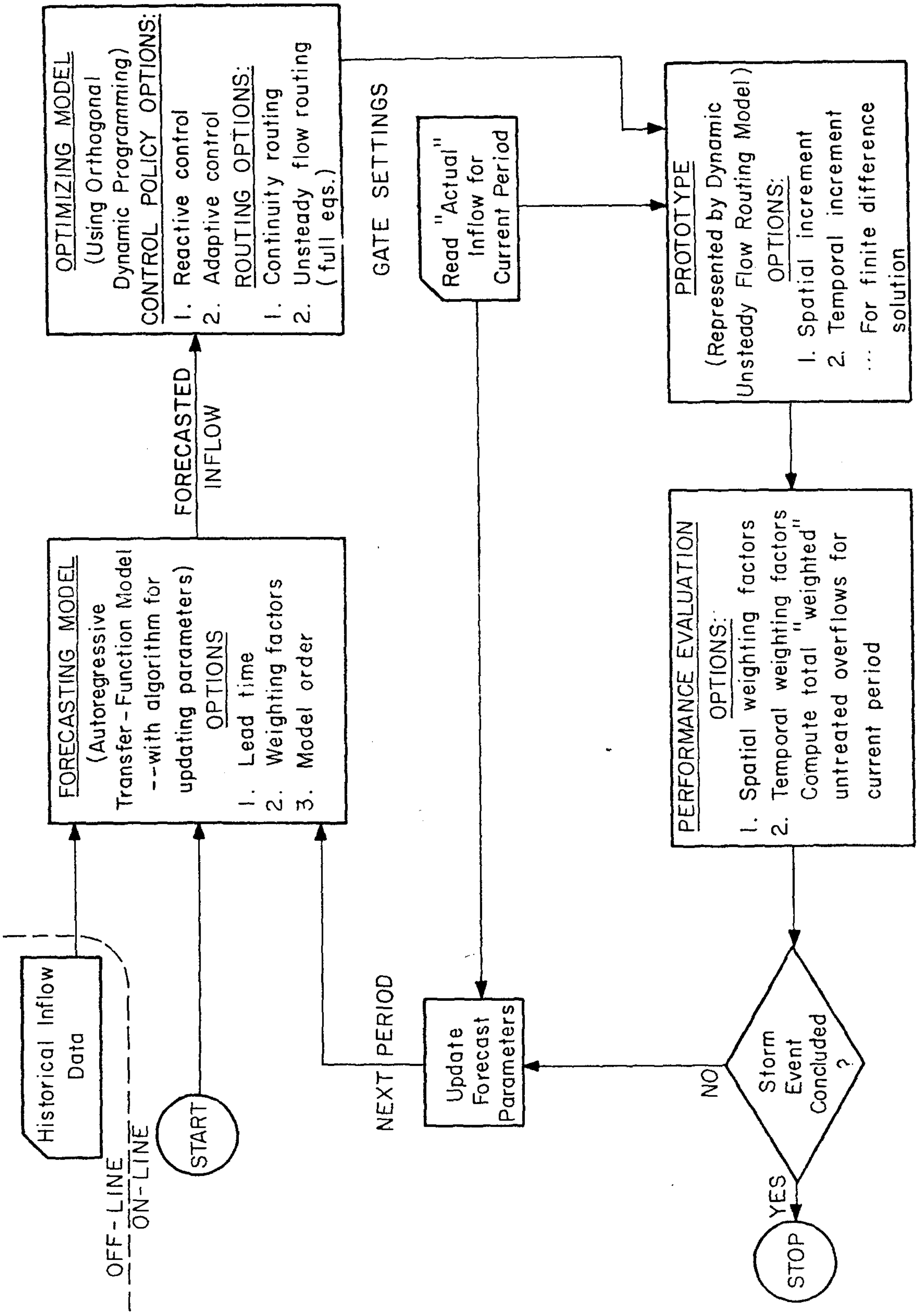


Figure III-1. Major components of the Stormwater Control Package (SWCP) (Labadie et al., 1981) specifies the number of future time periods over which forecasts will be made. For historical parameter generation, weighting factors may be selected for each storm and for each gaging location. Atypical storms should be weighted lower; typical storms should be given a higher weighting factor. The weighting factor selected for the real-event storm determines to what degree the historical base-line parameters will be modified. Too great a weighting factor on the real event will result in persistence forecasting. Weighting factors for gaging locations may be selected to favor remote gages serving as advance indicators of storm conditions.

The autoregressive model order determines the number of preceeding time periods which must occur before a forecast can be made. Forecast accuracy may improve when the order is increased since more is revealed about the storm in progress. The lead time required to effect control strategies limits the maximum model order. The model power determines whether the model structure is linear, quadratic, etc. Further experimentation is necessary to ascertain the effect of model orders greater than one on forecast accuracy. Further work should focus on including moving average terms into the model and providing radar input data on storm tracking for increasing forecast accuracy.

B.2 Unsteady Flow Routing Model

A river routing model developed by Chen (1973) which solves the full unsteady flow equations or St. Venant equations was adapted for the SWCP. The full equations are necessary to describe backwater and reverse flow conditions caused by downstream control. The continuity and momentum equations are modified to allow for reverse flows, control gates, lateral inflows, and overflows or flooding.

The solution of the continuity and momentum equations first requires a conversion from a continuous to a discrete solution space. Linearization of the St. Venant equations is accomplished by a fully implicit finite difference scheme. The nonlinear friction slope term, equation for control gates, and weir equation for overflows are all linearized by a first order Taylor Series approximation. The resulting sparse matrix of linear simultaneous equations is solved by an "interrupted double sweep" technique which can correct for gate linearization errors. Supercritical flows generated by estimated initial depth/flow conditions or falsely induced by linearization errors may also be ameliorated by an automatic reduction in the finite difference time step.

The interrupted double sweep technique is particularly applicable for solving the unsteady flow equations. The forward sweep involves the calculation of internal arrays which define linear relationships between the unknowns (i.e., flow and depth) at every section of the reach. The upstream boundary condition, which is usually an inflow hydrograph, defines the starting point for the forward sweep. Upon completion of the forward sweep and after definition of downstream boundary conditions (i.e., sewer outfall, pump rating curve, etc.) the backward sweep proceeds upstream and solves for the unknown variables.

The backward sweep is interrupted at each control point, transition region, or other discontinuity. A calculation is made to check if the resulting depths and flows satisfy the actual nonlinear equation for flow at that point, such as the equations for flow under a gate. If the error exceeds a preset tolerance, flows are corrected and an informative error message is printed out by the computer program. The backward sweep continues upstream with interruptions to check accuracy at each control point until the upstream section is reached. This method allows sections upstream of a control gate to accommodate any corrections for linearization errors.

The same unsteady flow routing model is actually used in two ways in the SWCP by reading in two different sets of input data:

1. The routing model simulates the prototype sewer system and predicts results of the control strategies as specified by the dynamic programming (DP) optimization algorithm.

2. The routing model is also applied independently for each control section, which represent stages in the DP, and used within the DP algorithm to constrain the objective function. Stated another way, the optimization of the objective function is constrained by the flow routing dynamics of the gradually varied unsteady flow routing model. A simple continuity routing approach has been incorporated in the SWCP as a user option for obtaining rough initial policies.

These two different approaches to unsteady flow routing can differ in accuracy. The staged DP routing model is called a number of times in the optimization, and as such, must use a coarser finite difference grid of Δx and Δt in order to save computer execution time. The prototype simulation model is called only once for each forecast control interval, and can incorporate shorter time steps and spatial increments and serve as a check on policies derived by the optimization model.

Note that currently the SWCP can be applied to branched systems only one branch at a time. Internal boundary conditions at branch junctions must be satisfied iteratively by the user by running the SWCP several times for each branch. A version of the unsteady flow model has been modified to include branches, but the staged control structure under branching systems has not been formalized as yet. This is left for future research.

B.3 Optimizing Model

The optimizing model combines a heuristic dynamic programming (DP) algorithm with an unsteady hydraulic routing model to find optimal control decisions for gates, pumps, valves, adjustable weirs, etc. The model directly optimizes with respect to flows at the control point, and then calculates the resulting gate setting. This means that other control elements besides gates, such as pumps and adjustable weirs, can be simulated using a "dummy" gate equation. Stages in the DP are separated at the control points. The DP stages must be defined spatially in order to allow feasibility checks for the correct combination of the flow through a control element and sewer system heads as predicted by the unsteady flow simulation model. Infeasible combinations of heads and flows are tagged as unacceptable policies.

The dimension of the overall control problem is one greater than the forecast lead time; i.e., future time periods for which a forecast is made. Optimal control policies which use forcasts over a lead time greater than one are termed "adaptive" policies. Reactive policies only use the current period inflows in the optimization. The DP model specifies the flow vector through each control element as a perturbation from the user-supplied initial flow trajectory. This corridor-type approach to decompose the multi-dimensional DP problem employs orthogonal polynomials to approximate control flow trajectories in the temporal state. The result is a sequence of one-dimensional DP problems which successively optimize the coefficients (i.e., surrogate state variable) associated with each successive term in the orthogonal polynomial. The usual approach is to begin with the lowest order terms and move higher; however, the reverse approach has proved successful in some cases

(Labadie et al., 1980). This approach in effect performs a more rapid perturbation around a current gate flow trajectory than can be accomplished by methods such as incremental dynamic programming or discrete differential dynamic programming (Heidari et al., 1971) since the entire trajectory is perturbed rather than just one state increment at a time. Convergence to a local optimum is not guaranteed; however, rapid calculation of improved operating policies is usually of more importance for real-time control applications. A new approach called objective-space dynamic programming, developed by Fontane et al. (1981) offers great promise for effectively solving this problem with assurance of discrete global optimality. This is an important area for future research.

B.4 Operation of the SWCP

The SWCP simulates an on-line operational environment as follows:

1. Historical and/or synthetic runoff hydrographs representative of a range of interceptor sewer inflows are evaluated off-line using rainfall data and an appropriate watershed outflow prediction model. Parameters are derived from the historical data for use by the identification algorithm in the forecasting model. The historical hydrographs may be examined and categorized as to seasonal influences, storm duration, or storm depth, and a separate set of historical parameters derived for each category. Each historical storm hydrograph also requires a weighting factor which determines the relative importance this storm has in the historical data base.

2. Entire runoff hydrographs for each input location are read into the SWCP and designated as a real event to be simulated; but they are actually used by the model sequentially over each time interval. Data describing the sewer system are read into the optimizing model and the prototype simulation model. Initial flow discharge and depth are

specified for each sewer system reach. The downstream boundary condition is also specified (i.e., a pump rule curve or depth-discharge relationship).

3. Options are next selected for the forecasting model, which include: model order, weighting factors for the real storm event, in contrast with the historical base-line events; gaging location weights, and the number of time periods into the storm before the first forecast is made. The model then forecasts the inflow hydrographs for each location along the interceptor sewers over the specified lead time. These forecasted inflow hydrographs are then passed to the optimizing model.

4. The optimizing model determines the optimal flows through each control section and then computes the resulting settings for gates so as to minimize the occurrence of overflows, or, if overflows are unavoidable, to minimize the pollution impact on receiving waters, as represented by the objective function in the dynamic programming algorithm. Control policy options include the level of control (reactive or adaptive), storage routing or full unsteady flow routing between actuators, weighting factors for overflows in space and time, and discretization intervals for space and time for the finite difference solution of the unsteady flow routing. Initial trial settings of the control section flows are specified by the user for the storm duration. The optimizing model then proceeds to improve this initial policy. As mentioned previously, control elements other than gates can be simulated.

5. When significant improvement in operating policy cannot be found, the actual incoming storm hydrograph and improved strategy for the control actuators, up to and including the next time period only, are simulated by the prototype model and the performance of the system is measured. The operator is given the option of intervening and

specifying control settings for the actuators if it is felt that a more intuitive strategy will prove superior.

6. The SWCP then proceeds to the forecasting model which updates the forecasted hydrographs based on the measured inflows (i.e., the actual incoming storm) from the just elapsed time interval. The updated, forecasted hydrographs are then input to the optimizing model and the entire process continues until the storm event is concluded.

Again, the SWCP is currently not applicable to branched sewer systems. However, branched systems can be indirectly modeled using the SWCP by iterative processes that seek to satisfy internal boundary conditions at branch junctions, with the SWCP applied to one branch at a time.

C. FORECASTING MODEL

C.1 Autoregressive-Transfer Function Model

Successive observations or measurements of rainfall or runoff time series are highly correlated. Time series regression analysis techniques which attempt to account for dependencies between elements in a time series are generally referred to as Box-Jenkins (1976) models.

Such models assume that a time series in which successive observations are dependent can be modeled as a linear combination of independent random disturbances or shocks drawn from a stable distribution. Such a series of disturbances is called a white noise process (see Graupe (1976)). Let $\varepsilon(k)$, $\varepsilon(k-1)$, $\varepsilon(k-2)$, ..., represent these random components and a_0, a_1, a_2, ... represent the weighting coefficients associated with them. The dependent sequences of rainfall or inflows $R^i(k)$ at location i and time k can then be represented as

$$R^i(k) = a_0\varepsilon(k) + a_1\varepsilon(k-1) + a_2\varepsilon(k-2) + \cdots \tag{1}$$

Such a stochastic model process is usually called a linear filter. Successive observations of $R^i(k)$ are dependent because they are drawn from the same previous realizations of $\varepsilon(k)$. This model actually transforms a dependent time series into a white noise process.

Forecasting models derived from white noise process models are capable of representing both stationary and nonstationary time series. Stationary processes are those in which the series fluctuates around some constant mean level, while nonstationary processes have no such mean level.

The above model has, however, an infinite number of terms in its definition and consequently is of little use. Various approaches are available to find an efficient or parsimonious model which adequately represents the process for the purpose of forecasting.

Autoregressive models assume that the independent random variables are the previous members of the considered series. Such a model using p terms back in time is abbreviated as AR(p). It is written as

$$R^i(k) = b_1R^i(k-1) + b_2R^i(k-2) + \cdots + b_pR^i(k-p) + \varepsilon(k) \tag{2}$$

where $\varepsilon(k)$ is white noise and the coefficients are again unique. Using backshift notation and the function $\varphi(B)$ defined as

$$\varphi(B) = b_1 + b_2B + \cdots + b_{p+1}B^p \tag{3}$$

enables the AR(p) forecast model to be written as

$$R^i(k) = \varphi(B)R^i(k-1) + \varepsilon(k) \tag{4}$$

Another type of model which can be used for forecasting might be termed a transfer function or input-output model. In this case, $R^i(k)$ is assumed correlated with other measurable inputs. For the situation considered herein, these other measurable inputs might be inflow measurements at each location j adjacent to location i for various past times. The structure of such a forecasting model would be:

$$R^i(k) = \sum_{j \varepsilon J(i)} \sum_{l=1}^{S} c_{jl}[R^j(k-l)] \tag{5}$$

where the coefficients are unique for each location, and where

J(i) = set of pertinent locations adjacent to i (the number of elements in each set is assumed equal to r).

S = number of time periods backward which the model considers; we will assume S = p.

Introducing the back shift function $\theta(B)$

$$\theta^j(B) = \sum_{l=1}^{S} c_{jl}B^l \tag{6}$$

enables a simpler representation of the forecasting model:

$$R^i(k) = \sum_{j \varepsilon J(i)} \theta^j(B)R^j(k-1) \tag{7}$$

A mixed autoregressive-transfer function model combines the features of the above models:

$$R^i(k) = \phi(B)R^i(k-1) + \sum_{j \varepsilon J(i)} \theta^j(B)R^j(k-1) \tag{8}$$

This model is needed for each location i. Assuming the parameters are available, or can be derived, it is a straightforward matter to use (8) for forecasting purposes. The time series we are modeling is highly nonstationary (i.e., a runoff hydrograph or rainfall hyetograph), so it was decided that the moving average terms would not improve forecast accuracy. Further research is needed to confirm this observation.

If the autoregression-transfer function model order is properly selected and the coefficients optimally determined, then the error series generated by the difference between the forecast and the actual occurrence should be a white noise process. Various techniques are available for deriving the best order (Graupe, 1976). In real-time forecasting, however, it may be necessary to alter the model order if it appears that the current event differs significantly from the historical data base.

We can also add additional terms to (8) with the $R^j(k-1)$ terms raised to integer powers. The final form of the model is

$$R^i(k) = \varphi(B)R^i(k-1) + \sum_{n=1}^{N} \sum_{j \varepsilon J(i)} \theta^{jn}(B)[R^j(k-1)]^n \tag{9}$$

The rainfall or runoff data from other locations adjacent to a specified location are designated as other measurable inputs within the transfer function portion of the model. Hence, we assume that previous measurements at adjacent or local locations can be as important for generating accurate forecasts as previous measurements at the location being considered.

The order p of the autoregressive part of the model should be selected with consideration of its impact on the total number of model parameters. For 10 locations, there are a total of 100p parameters for the complete model, assuming forecasts are generated at each location and all locations are correlated with each other.

C.2 Parameter Estimation

The parameter identification or estimation of the previously described model is carried out as follows. For notational convenience, let all the variables assumed to be correlated with forecasted rainfall or runoff by designated as u_i, i = 1, ..., pm(i), where m(i) is the number of elements in the set J(i) and the model order is p. Also for convenience, let us assume that m(i) = M, for all locations i. Since a separate forecasting model is needed for each location, the complete model is written as follows:

$$
\begin{aligned}
R^1(k) &= a_{1,1}u_1 + \cdots + a_{1,j}u_j + \cdots + a_{1,pM}u_{pM} \\
R^i(k) &= a_{i,1}u_1 + \cdots + a_{i,j}u_j + \cdots + a_{i,pM}u_{pM} \\
R^M(k) &= a_{M,1}u_1 + \cdots + a_{M,j}u_j + \cdots + a_{M,pM}u_{pM}
\end{aligned}
\tag{11}
$$

or, in vector form:

$$\underline{R}(k) = A\underline{u}$$

The a_{ij}'s then are the coefficients to be identified. Isolating the j-th row, it may be written as:

$$R^i(k) = [\underline{a}^i]^T\underline{u} \tag{12}$$

where

$$[\underline{a}^i]^T = [a_{i1}, a_{i2}, \cdots, a_{ij}, \cdots, a_{iM}] \tag{13}$$

Therefore, M separate identification problems are isolated with each set of parameters $\underline{a}^i$ being identifiable by sequential regression methods. The estimation of parameters is performed such that the estimated parameter vector $\underline{a}^i$ minimizes a squared error index J_s (the s denoting the estimation iteration) defined by the equation:

$$J_s = \sum_{k=1}^{s} q_k\{R^i(k) - [\tilde{\underline{a}}_s^i]^T \underline{u}_k\}^2 \tag{14}$$

where q_k are weighting factors on measurement error and $\underline{u}_k$ represents the $\underline{u}$ vector associated with period k (i.e., observations previous to period k at location i and all adjacent locations).

It can be shown (see Graupe (1976)) that for each model (dropping the index on the parameters indicating location):

$$\tilde{\underline{a}}_s = \tilde{\underline{a}}_{s-1} + P_s q_s \underline{u}_s (R^i(s) - \underline{u}_s^T \tilde{\underline{a}}_{s-1}) \tag{15}$$

where

$$P_s^{-1} = \sum_{k=1}^{s} q_k(\underline{u}_k \underline{u}_k^T) \tag{16}$$

Therefore, $\tilde{\underline{a}}_{s-1}$ may be derived sequentially from the previous estimate $\tilde{\underline{a}}_{s-1}$ as well as the previous measurements and weights, as long as P_s can also be computed sequentially. The matrix P_s is computed sequentially according to the relation:

$$P_s^{-1} = P_{s-1}^{-1} + q_s \underline{u}_s \underline{u}_s^T \tag{17}$$

where

$$P_0^{-1} = 0 \tag{18}$$

Instead of inverting the matrix P_s, the matrix inversion lemma is used to facilitate the recursive derivation of P_s, yielding:

$$P_s = P_{s-1} - P_{s-1}\underline{v}_s(1 + \underline{v}_s^T P_{s-1}\underline{v}_s)^{-1}\underline{v}_s^T P_{s-1} \tag{19}$$

where

$$\underline{v}_s = \sqrt{q_s \underline{u}_s} \tag{20}$$

$$\underline{v}_s \underline{v}_s^T = q_s \underline{u}_s \underline{u}_s^T.$$

Since $(1 + \underline{y}_s^T P_{s-1} \underline{y}_s)$ is scalar, no matrix inversion is involved in deriving P_s and consequently $\tilde{\underline{a}}_s$.

C.3 On-Line and Off-Line Identification/Forecasting

The inflow forecasting model has two main parts: the off-line historic base line parameter identification model, and the on-line adaptive (current event) parameter updating model with inflow forecasting. The relationship between these functions is illustrated in Figure III-2.

The off-line, historical base line parameter identification model analyzes past storm records and estimates the parameters of the inflow forecasting model. These parameters could be estimated in such a way that more relevant storms are given higher weighting. Relevance, in this case, would be defined by how recent the record is and the similarity of current meteorologic conditions to those existent at the time of the historic event (e.g., season, wind direction, barometric pressure, etc.). The off-line, base line parameter identification model should be run periodicially to adjust the base line parameters to more currently relevant conditions.

As data describing the current storm event in progress become available, the on-line parameter estimation model can update the base line parameters. The weighting of these data will depend upon the sensitivity of the model. A weighting factor on current data which is too high will result in an erratic model which ignores trends identified in the historic data. Weighting factors on current data which are too low will, however, ignore the evolving structure of the storm currently being experienced. The determination of the proper balance requires extensive testing on a prototype system.

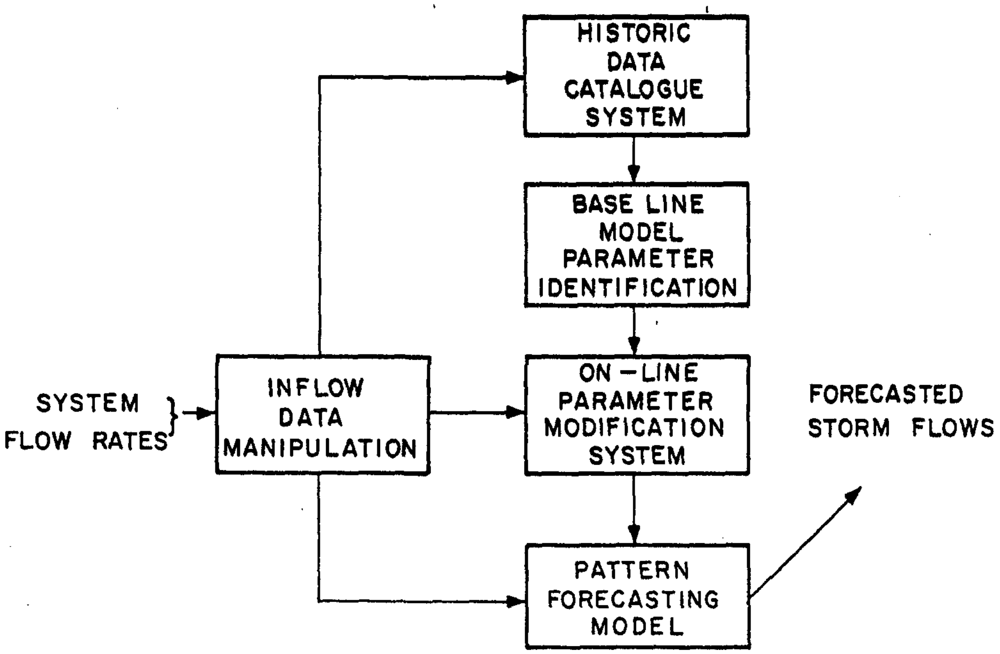


Figure III-2. Components of the forecasting model.

The pattern predictor simply uses the previously identified parameters and available data which describe the unfolding event and forecasts the subsequent inflows for the chosen lead time. Again, the lead time is the number of time periods into the future that we desire to forecast. The estimated coefficients of the autoregressive transfer function forecasting model are simply multiplied by the appropriate value of the previous inflows (or previously forecasted inflows) to obtain the forecasted inflow for the desired location.

The model which identifies the parameters needed for each forecasting model is the same algorithm used in the on-line parameter modification system since they both must be of the same order or dimension. They are illustrated as separate, however, to accentuate the fact that the off-line identification may be used experimentally by the system operators. Working off-line facilitates the search for the most efficient model configuration. The off-line model must, however, retain in some storage facility the appropriate information needed by the current on-line model to be used during storm events. The algorithm developed simply effects the iterative identification procedure developed in the previous section after appropriately arranging the needed data into prediction model inputs and outputs. After the parameters have been defined to the extent possible from the historic data, and any data available on the current event, the inflow forecast portion of the model then forecasts the progressing event for a given lead time at a particular location. This is accomplished by the repeated application of the updated model for that location.

C.4 Computational Experience

The forecasting model employed in the SWCP has been tested using data from a portion of the dense raingage network of the City and County of San Francisco, as well as data from outlying gages to the northwest in Marin County (see Figures 1 and 2 in Appendix A). Appendix A presents the results of application of the model to forecasting four actual historical storm events capable of producing untreated overflows; two of them are intense and of short duration, whereas the other two are less intense but of longer duration. A data base for off line analysis includes 29 preceeding events spanning a three year period, with deliberate exclusion of these four events from the data base. This allows as fair testing of the ability of the forecasting model. These 29 events are used to establish the base-line parameters of the forecasting model. As the additional four storm events unfold in "simulated" real-time, the forecasting model is allowed to update in accordance with these new data.

Examples of convergence of the recursive off-line parameter estimation procedure can be found in Figures 3 and 4 of Appendix A. Figures 5 to 8 of Appendix A provide comparisons of actual and forecasted hourly rainfall for the four selected events. It is evident that the short duration, highly intense events like Storm #33 provide the most difficulty for the model. Tables 1 and 2 of Appendix A give the autocorrelation and crosscorrelation functions, and Tables 3 and 4 the average absolute forecast errors for Storms #31 and #33. These latter results are graphically presented in Figure III-3 of this section.

Notice from Figure III-3 that average forecast error usually increases with the lead time, since we are trying to use the model to forecast further into the future. An exception would be time period 4 for both storms.

Notice also that for the intense storm (#33), forecast error decreases dramatically in real-time as the model is "learning" more

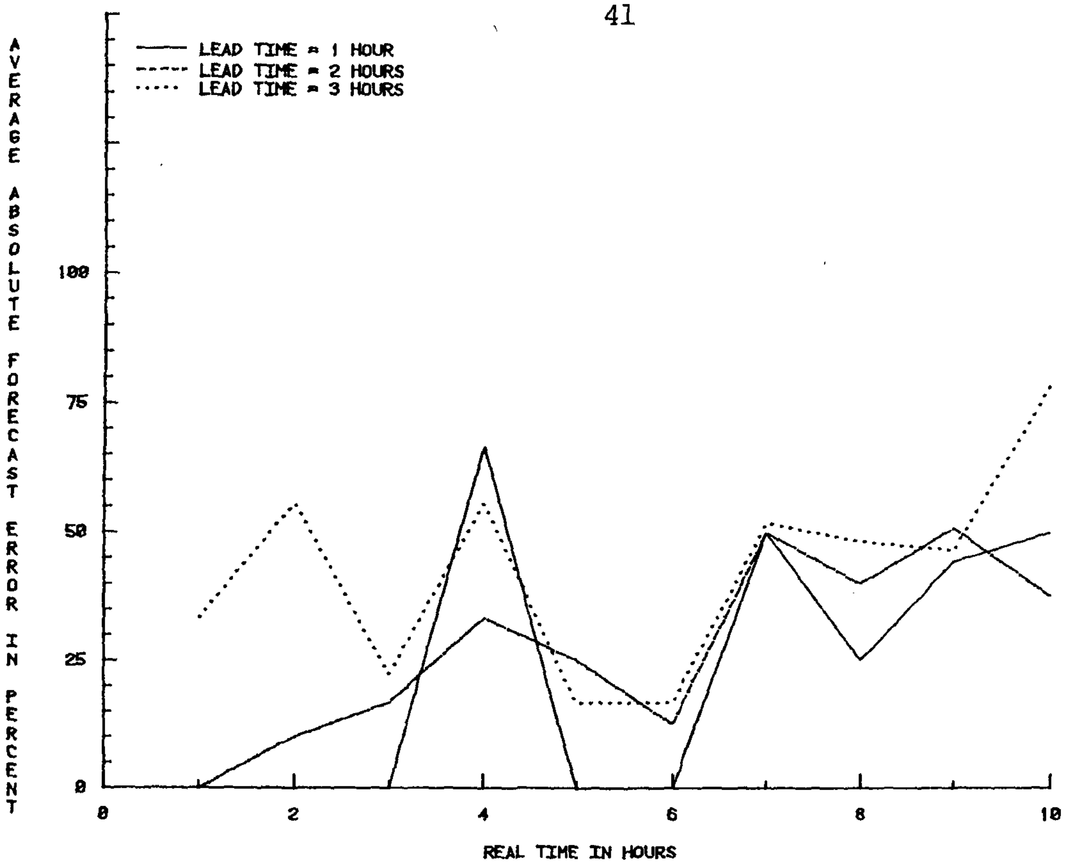


Figure III-3a. Average absolute forecast error in percent for Storm #31.

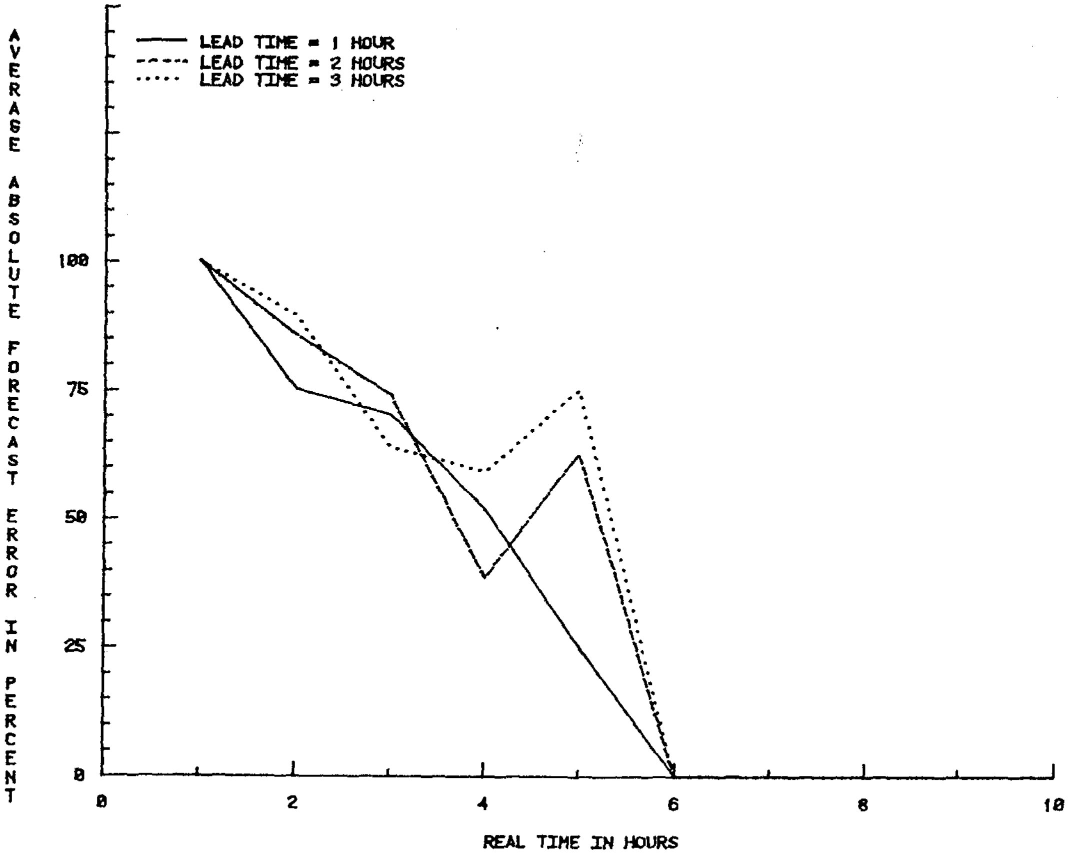


Figure III-3b. Average absolute forecast error in percent for Storm #33.

about the current event. For Storm #31, this is not really apparent. Here, the errors tend to remain within 0 to 50% over the first 10 hours. For these experiments, a homogeneous weighting structure was used. It may be that a different weighting could reduce the errors. Though these errors seem somewhat high, it is shown in Appendix A that forecast errors up to 70% can still be valuable in finding optimal adaptive control policies, when compared with no forecasting at all. Further research is needed on how to improve forecast performance by incorporating additional information, such as radar data and meteorological conditions, or modifying the structure of the model.

D. HYDRAULIC ROUTING MODEL

The continuity equation for the hydraulic model is written as

$$\frac{\partial Q}{\partial x} + T\frac{\partial y}{\partial t} = q_1 \qquad (21)$$

where x,t = spatial and temporal coordinates, respectively, and

y = depth of flow

Q = discharge rate of flow

$T = \frac{dA}{dy}$, where A is the cross-sectional area of flow

q_1 = net of lateral inflow and outflow per unit length

Lateral outflow in a sewer system is related to head by a weir equation. When the sewer becomes full, the change in cross-sectional area is negligible. Assuming that the water hammer generated is insignificant, we have

$$\frac{\partial Q}{\partial x} = q_1 \qquad (22)$$

The momentum or dynamic equation is written in the form of

$$gA \frac{\partial y}{\partial x} + |v| \frac{\partial Q}{\partial x} + v \frac{\partial |Q|}{\partial x} - v|v| T\frac{\partial y}{\partial x}$$
$$- v|v| A_x^y + \frac{\partial Q}{\partial t} = gA(S_o - S_f) \quad (23)$$

where

v = flow velocity ($|v|$ and $|Q|$ are used in order to consider flow reversals)

$A_x^y = (\frac{\partial A}{\partial x})_y$, which equals zero, except at channel transitions

g = gravitational acceleration

S_o = channel bottom slope

S_f = friction slope

Equations 21-23 are called the St. Venant equations, and consider friction and all significant inertial terms of the moving fluid. They are capable of representing waves moving upstream due to backwater effects. These effects become important in sewer systems when there are relatively flat slopes, tidal influences, and downstream control. They are, however, incapable of describing abrupt transitions, hydraulic jumps, or steep-front surges and bores.

The St. Venant equations are nonlinear hyperbolic partial differential equations which require two hydraulic boundary conditions (upstream and downstream) for describing subcritical flow, and also require initial conditions of depth and velocity (or discharge) for all discrete sewer sections. For this problem, the initial conditions are taken as dry weather flow.

Sewer systems generally include different sizes and types of pipe or channel. Flow through size transitions is described by the following equations:

$$Q_1 = Q_2$$

(24)

$$h_1 + \frac{v_1^2}{2g} = h_2 + \frac{h_2^2}{2g} + h_f \tag{25}$$

where subscripts 1 and 2 denote the sections immediately upstream and downstream of the transition, respectively; h_f is the energy head loss. If the energy loss and kinetic energy head are small compared to the water head, then equation 25 becomes

$$h_1 = h_2 \tag{26}$$

Flow through a control gate in the sewer system (Figure III-4) is governed by

$$Q_1 = Q_2 \tag{27}$$

$$Q_1 = K\,A\sqrt{2g\,|h_1 - h_2|} \text{ if } h_1 \geq h_2$$

with

$$Q_1 = -K\,A\sqrt{2g\,|h_1 - h_2|} \text{ if } h_1 < h_2 \tag{28}$$

where K is a gate coefficient and A is the cross-sectional area of the gate opening. Coefficient K actually varies with flow conditions, but is assumed as constant in the SWCP (U.S. Army Corps of Engineers, 1971).

Overflows to receiving waters are modeled using the broad crested weir formula

$$q_{OF} = K_w(y_j - z_s)^{1.5} \tag{29}$$

Since this assumes a rectangular cross section for the outfall (Figure III-5). Some error would occur if the outfall is circular. It would not be difficult to modify the SWCP to include circular outfalls.

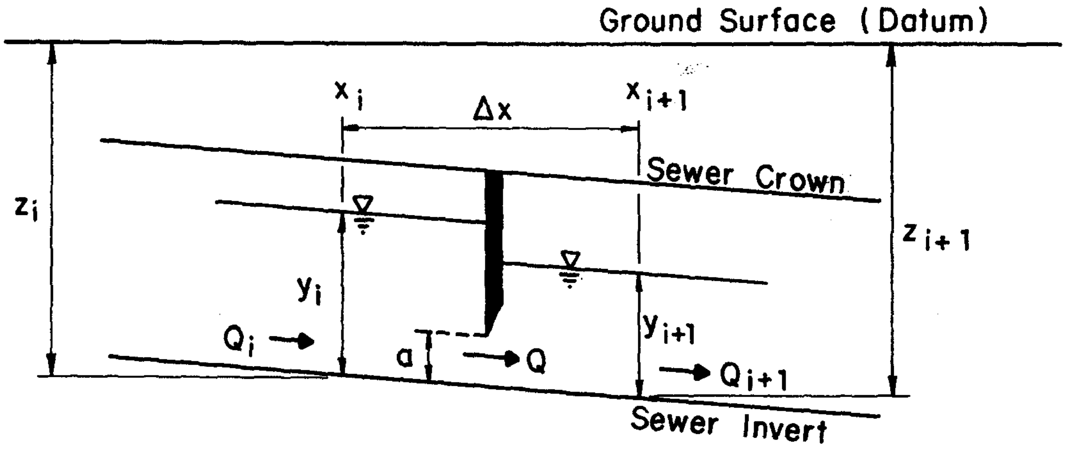


Figure III-4. Sewer section with a control gate.

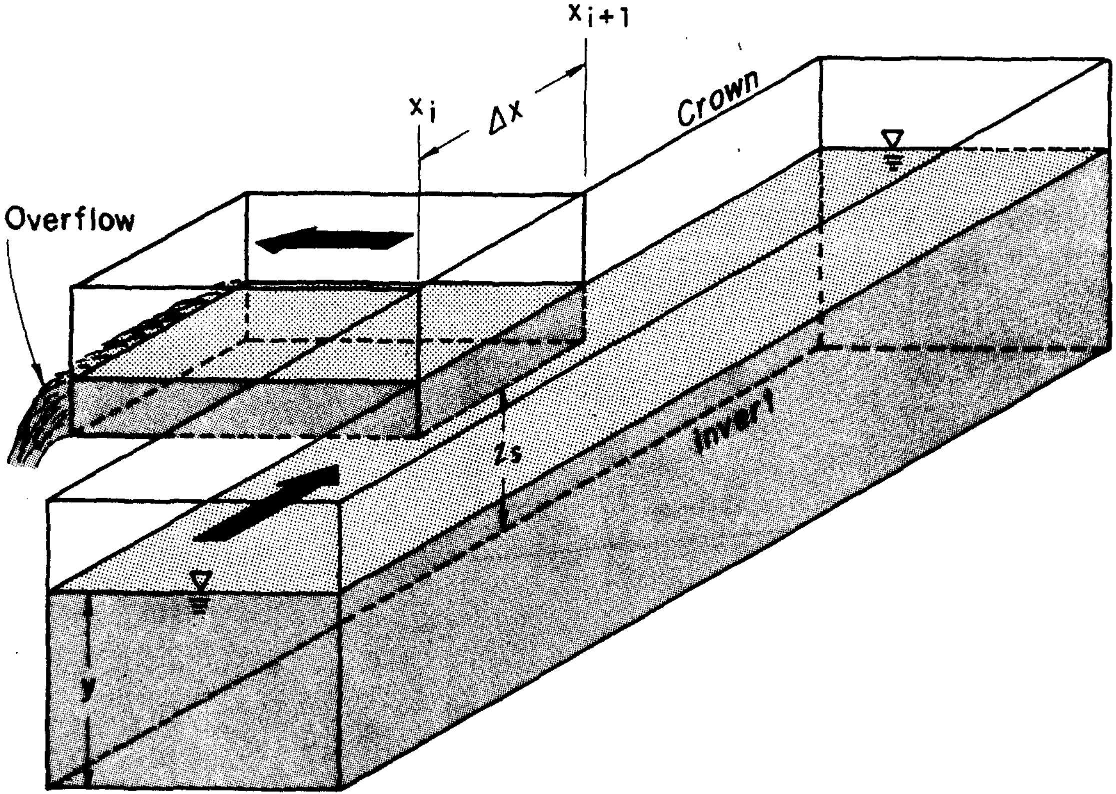


Figure III-5. Sewer section with an overflow weir.

Street flooding is assumed to occur in any section where the head in that section exceeds the ground surface less datum elevation. Note that for circular pipe, pipe full conditions are assumed to occur at a minimum water surface width of 3 cm.

Based on convergence and stability studies conducted by Chen (1973), Liggett and Cunge (1975), and Ponce et al. (1978), a fully implicit scheme for numerical solution of the partial differential equations was selected. A fully implicit scheme would use the approximation at point I on Figure III-6. The scheme is quite stable, which allows for selection of larger time steps and section lengths and therefore can result in significant savings in computer time and storage. This is an important consideration for real-time control applications.

The nonlinearity introduced by equations 28 and 29 and the friction slope

$$S_f = \frac{Q|Q|}{\left[\frac{1.486AR_h^{2/3}}{n}\right]^2}$$

where

R_h = hydraulic radius

n = Manning's roughness coefficient

can greatly increase computer time. Therefore, these relations are linearized using the truncated Taylor series. The continuity equation (21) is linearized by assuming T is known from the previous time step. For the momentum equation, area A and velocity v are also assumed to be known from the previous time step.

The sewer is divided into N sections for the numerical solution. This gives 2N simultaneous linear algebraic equations for the continuity

and momentum equations, along with the boundary conditions, in 2N unknowns (depth and discharge). The reader is referred to Labadie et al. (1980) for details on the structure of these equations. Since the numerical scheme is implicit, solution is accomplished by a double sweep process (during each time step) from the upstream boundary conditions to the downstream conditions, and back again. The technique is similar to Gauss Elimination, modified for reduced computer storage in order to take advantage of the sparse matrix structure.

Even though the algebraic equations are linear, nonlinearities are introduced through changes in flow conditions. For flow simulation under a specified gate control strategy, the upstream sweep is interrupted at each control gate in order to correct for linearization errors in the gate or orifice equation. A calculation is then made to check if the resulting depths and flows reasonably satisfy the original nonlinear gate equation. Calculations may also be interrupted at abrupt transitions in sewer size and shape. If the error exceeds a preselected tolerance, the flows are corrected and an informative error message is printed out.

E. OPTIMIZATION MODEL

E.1 Dynamic Programming Formulation

The unsteady flow model has been calibrated against physical scale model data by Morrow (1978), and with actual data by Book (19). These studies have confirmed the accuracy and applicability of the model.

The objective of the deterministic optimal control problem is to minimize total (weighted) untreated overflows at all overflow locations, and over a future forecast period. That is:

$$\text{minimize} \sum_{i=1}^{N} \sum_{t=\tau}^{\tau+L} [\omega_{it} O_i(t) + P\bar{O}_i(t)] \qquad (30)$$

where

N = the number of stages. In general, not every stage will have both a control point and an overflow location, but should have one or the other. An example 3-stage problem is shown in Figure III-7, taken from the North Shore Outfalls Consolidation Project of San Francisco.

τ = current discrete control time interval (integer valued)

L = number of lead time intervals over which storm inputs are forecasted (i.e., 0, 1, 2, ...). The selected interval may be as short as five minutes or as long as one hour, depending on the control precision required ($L=0$ is the reactive case).

ω_{it} = weighting factors used to set priorities on the timing and locations of overflows, if they are unavoidable.

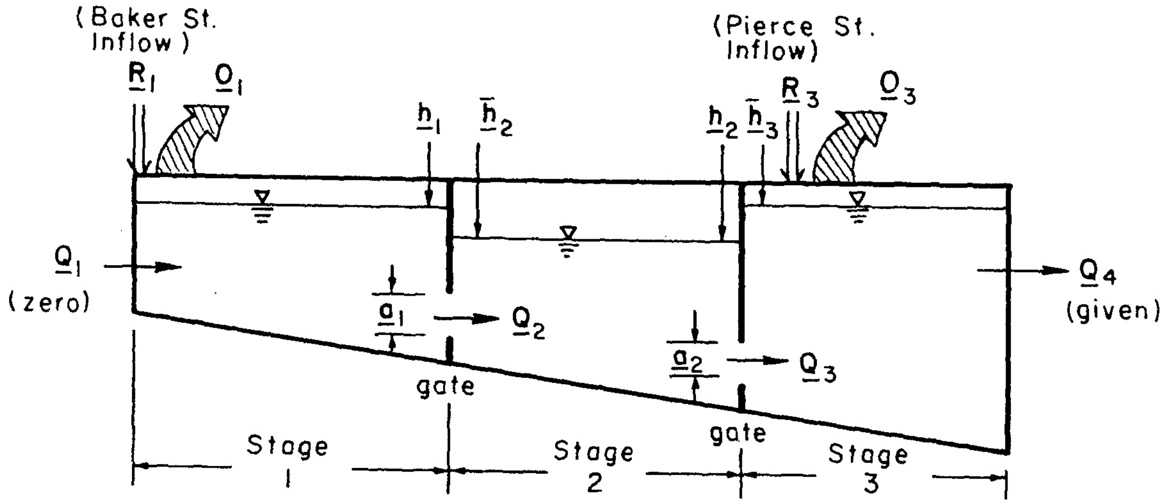
$O_i(t)$ = average rate of untreated overflows from stage i, during control period t.

$\bar{O}_i(t)$ = average rate of street flooding due to surcharged sewers, which is assigned an extremely high penalty P due to potential health hazards and nuisance factors.

The constraints include:

1. flow routing dynamics (i.e., the aforementioned unsteady flow model)
2. sewer capacity
3. hydraulic boundary conditions.

Although portions of the unsteady flow model have been linearized, it has also been necessary to introduce certain nonlinearities, as discussed previously, in order to deal with abrupt transitions, gate



<u>definitions</u>

$\underline{Q}_i$ = $(Q_i(\tau),\ldots,Q_i(\tau+L))$ = flow rate vector into stage i over lead time L from current period τ, defined at the *beginning* of each period t, t = τ,...,τ+L [$Q_i(t) < 0$ indicates reverse flow].

$\underline{R}_i$ = forecasted storm inflow rate vector over lead time L $(R_i(\tau),\ldots,R_i(\tau+L))$.

$\underline{O}_i$ = untreated overflow vector over lead time L $(O_i(\tau),\ldots,O_i(\tau+L-1))$, where $O_i(t)$ = average rate *during* period t, t = τ,..., +L-1.

$\underline{\bar{h}}_i$ = head vector over lead time L at the upstream end of stage i $(\bar{h}_i(\tau),\ldots\bar{h}_i(\tau+L)$.

$\underline{h}_i$ = head vector over lead time L at the downstream end of stage i $(h_i(\tau),\ldots,h_i(\tau+L))$.

$\underline{a}_i$ = gate opening size vector over lead time L $(a_i(\tau),\ldots,a_i(\tau+L))$.

Figure III-7. Sewer reach with two gates.

linearization errors, and supercritical flow. That is, certain conditional logic in the computer program is required. This precludes use of optimal control theory and the maximum principle since it would be difficult to handle the resulting nonlinear flow routing dynamics. When certain system nonlinearities cannot be conveniently linearized, then dynamic programming emerges as a potentially viable solution technique if the dimensionality problem can be resolved.

The dynamic programming recursion relation for stage i is defined as (underscored variables represent vectors):

$F_i(\underline{Q}_i)$ = the minimum total weighted overflows over forecast lead time L from stage i through final stage N, with controlled hydrograph $\underline{Q}_i = (Q_i(\tau),\ldots,Q_i(\tau+L))$ entering at the upstream end of stage i

$$= \min_{\underline{Q}_{i+1}} \left[\sum_{t=\tau}^{\tau+L} [\omega_{it} O_i(t) + \overline{PO}_i(t)] + F_{i+1}(\underline{Q}_{i+1}) \right] \tag{31}$$

(for i = 1, ..., N-1)

subject to:

$$\underline{Q}_{\min,i+1} \leq \underline{Q}_{i+1} \leq \underline{Q}_{\max,i+1} \tag{32}$$

$$[h_{it}(\underline{Q}_i, \underline{Q}_{i+1}) - \bar{h}^{*}_{i+1,t}(\underline{Q}_{i+1})] \cdot Q_{i+1}(t) \geq 0 \tag{33}$$

(for t = τ, ..., τ+L)

where

$\underline{Q}_i = (Q_i(\tau),\ldots,Q_i(\tau+L))$ = controlled inflow hydrograph to stage i. Notice that $Q_i(t)$ may be negative, thereby representing upstream flow.

$\underline{Q}_{\min,i+1}, \underline{Q}_{\max,i+1}$ = preset bounds on flow through the control point upstream of stage i+1, or downstream of stage i.

$h_{it}(\underline{Q}_i, \underline{Q}_{i+1})$ = head downstream of stage i, at the start of control period t; computed by running the unsteady flow model using boundary conditions $\underline{Q}_i$ and $\underline{Q}_{i+1}$.

$\bar{h}^*_{i+1,t}(\underline{Q}_{i+1})$ = the optimal head upstream of stage i+1, computed and stored from previous dynamic programming calculations for stage i+1.

Notice that if lead time L=0, we are simply optimizing flows $Q_i(\tau)$ over the current period τ, and not attempting to forecast inflows over future time periods. With L=0, the current measurable inflows are simply extrapolated through the current period τ.

Equation 33 guarantees that for control gates, the head differential across the gate and flow through the gate are of the correct sign. The overflows $O_i(t)$ and street flooding $\bar{O}_i(t)$ are also computed using the unsteady flow model. The hydraulic simulation model can be run for the current stage i only since upstream and downstream boundary conditions in stage i are completely defined.

Notice from Figure III-7 that a distinction is made between an uncontrolled inflow hydrograph vector $\underline{R}_i$ and a controlled inflow hydrograph vector $\underline{Q}_i$ representing flow under a gate. Therefore, initially $\underline{Q}_i = \underline{0}$, and the $\underline{R}_i$ serve as the given upstream boundary conditions.

For flow under a submerged gate, the height of the gate opening that will produce flow $Q_i(t)$ is

$$a_{it} = \frac{|Q_i(t)|}{Kb\sqrt{2g}\,|h_{it}(\underline{Q}_i,\underline{Q}_{i+1}) - h^*_{i+1,t}(\underline{Q}_{i+1})|^{1/2}}, \text{ for all } t \tag{34}$$

which is constrained to lie between $a_{\min,it}$ and $a_{\max,it}$ in order to keep the gate opening submerged. For (34), K = head loss coefficient for the gate and b = gate width. Notice also in (34) that $Q_i(t)$ is one component of the vector $\underline{Q}_i$.

It should be noted that the discharges $Q_i(t)$ represent flow rates at the beginning of time period t. The control interval Δt will generally be some multiple of the time step for the hydraulic simulation model. Therefore, we linearly interpolate the $Q_i(t)$ in order to provide values at smaller time intervals for the simulation model. Also, feasibility checks (equation 33) are made for each of the hydraulic model time steps to ensure that infeasible flow conditions do not occur during the longer control interval (e.g., which might arise if a flow reversal occurs during the control interval). It is assumed that any gates are appropriately adjusted (using equation 34) during the control time interval to produce the desired $Q_i(t)$, given the head differentials that develop during the control interval. The overflow rates $O_i(t)$ and $\bar{O}_i(t)$ represent average rates over the control interval.

E.2 Solution Procedure

The dimensionality of the dynamic programming (DP) problem (i.e., L+1 state variables per stage), along with the imbedded solution of the full unsteady flow model, make it computationally infeasible to solve this problem by standard dynamic programming. Incremental DP (Hall et al., 1969), discrete differential DP (Heidari et al., 1971) or successive approximations (Larson, 1968) may be used to deal with the dimensionality problem. Computational experience indicates that for L greater than one, even these techniques are too time consuming, due to the presence of the unsteady flow model.

An alternative approach for this problem is to approximate the vector $\underset{\sim}{Q}_i$ using orthogonal polynomials. That is, let

$$Q_i(t) = Q_i^{(1)}(t) + \sum_{j=0}^{R} \alpha_{ij} T_j(t), \text{ for } t = \tau, \ldots, \tau+L \tag{35}$$

where $Q_i^{(l)}(t)$ is a current known trajectory for the controlled hydrograph at some current iteration l, $T_j(t)$ are orthogonal polynomials of order $j = 0, \ldots, R$; $R \leq L$, and the α_{ij} are coefficients to be determined. In effect, then, our state variables now become the α_{ij} coefficients, and the optimal return function $F_i(\underline{Q}_i)$ can be written as $F_i(\alpha_{i0}, \alpha_{i1}, \ldots, \alpha_{iR})$, since the flow vector $\underline{Q}_i$ can be approximated by equation 35 and the specified α_{ij} coefficients. With selected values for these coefficients, equation 15 will be used to generate various perturbations around current trajectory $\underline{Q}_i^{(l)} = (Q_i^{(l)}(\tau), \ldots, Q_i^{(l)}(\tau+L))$.

Though this is still an R+1 dimensional problem, suppose we take advantage of the orthogonality property associated with the polynomials. That is, for curve fitting problems, the optimal values of the parameters α_{ij} are independent of the order R (Graupe, 1976). That is, we could optimize over each of the α_{ij} one at a time, rather than all at once, and still end up with the best fit. This independent property would not be completely valid for our problem since we are not fitting a curve to a priori known data. However, it would be reasonable to assume that orthogonal polynomials would more closely approximate this property for our problem than nonorthogonal polynomials.

An algorithm is immediately suggested by the above discussion. Suppose we initially use the zeroth order term only. That is, all other $\alpha_{ij}(j=1, \ldots, R)$ in (35) are temporarily set to zero. The DP problem finds $F_i(\alpha_{i0})$ and determines the optimum α_{i0}^*, for $i=1, \ldots, N$. Since the zeroth order term is a constant, we are simply shifting the current trajectory up and down by a constant amount. All flows $Q_i(t)$ are computed using (35), which only needs specification of the α_{ij}. We then repeat this procedure, but this time finding $F_i(\alpha_{i0}^*, \alpha_{i1})$ with the previously found α^*_{i0} held constant and determining the optimum α^*_{i1}. The α_{i1} are associated with the 1st order term, which is linear. Therefore, the current trajectory is now being tilted over various slopes. We then hold α^*_{i0} and α^*_{i1} constant and allow α_{i2} to vary, and so on through order R. As the order increases, the trajectory is perturbed by a quadratic, cubic, and so on, which gives an increasingly varied perturbation of the trajectory. Since the α_{ij} are not necessarily independent, we may need to update the initial trajectory, return to the zeroth order again, and repeat the process. Therefore, the essence of this algorithm is that we proceed through orders sequentially, much like in a successive approximations procedure.

For this study, Chebyshev orthogonal polynomials have been selected for their attractive properties of giving more uniform fitting errors over a given range of data than other kinds of orthogonal polynomials (Graupe, 1976). It is convenient to use the following transformation of $\tilde{t}=t-\tau$ (i.e., $\tilde{t}$ represents the number of future time periods from current real time interval τ):

$$\xi = \cos[(2\tilde{t}+1)\pi/2(L+1)], \text{ for } \tilde{t} = 0,1,\ldots,L \tag{36}$$

which guarantees that $\xi \varepsilon [-1,1]$. This transformation simply normalizes the time domain. The Chebyshev polynomials are generated by

$$T_0(\xi) = 1 \ ; \ T_1(\xi) = \xi \tag{37}$$

$$T_{j+1}(\xi) = 2\xi T_j(\xi) - T_{j-1}(\xi) \tag{38}$$

$$(\text{for } j=1,\ldots,R-1)$$

and satisfy the orthogonality property

$$\sum_{\xi} T_j(\xi)T_k(\xi) = 0, \text{ for all } k \neq j \tag{39}$$

In more formal terms, the algorithm proceeds as follows:

1. Start at order r = 0. Assume that an initial release schedule $Q_i^{(0)}(t)$ for all i,t is given. Initially set all coefficients $\alpha_{ij}^{(0)} = 0$ for all orders j=0,...,R. Set the cycle counter l=0. One cycle occurs when we pass through all the orders once.

2. Given current coefficients $\alpha_{ij}^{(l+1)}$ for orders j=0,...,r-1 (not defined at r=0)

$$Q_i(t) = Q_i^{(l)}(t) + \sum_{j=0}^{r-1} \alpha_{ij}^{(l+1)} T_j(\xi) + \alpha_{ir} T_r(\xi) \tag{40}$$

That is, we are at current order r, and have found optimal $\alpha_{ij}^{*} = \alpha_{ij}^{(l+1)}$ for the previous orders j=0,...,r-1.

3. We now solve the following one-dimensional DP problem (for all stages)

$$F_i(\alpha_{ir}) = \min_{\alpha_{i+1,r}} \left[\sum_{t=\tau}^{\tau+L} [\omega_{it} O_i(t) + \overline{PO}_i(t)] + F_{i+1}(\alpha_{i+1,r}) \right] \tag{41}$$

$$(i = 1,\dots,N-1)$$

$$F_N(\alpha_{Nr}) = \sum_{t=\tau}^{\tau+L} [\omega_{Nt} O_N(t) + \overline{PO}_N(t)] \tag{42}$$

subject to: (32), (33), (34), and (40).

It should be noted that if any $Q_i(t)$ generated from (40) violates a bound on $Q_i(t)$ (i.e., equation 32), then it is simply set to that bound. In addition, and of great importance to the performance of this method, is the need to set $Q_i(t) = 0$ if the current head differential across a gate is too small to feasibly allow an appreciable flow. If this is the case, $Q_i(t)$ is pinned to zero and the gate closed in spite of what is generated by the orthogonal polynomials, until further orders and cycles result in larger head differentials.

4. Set $\alpha_{ir}^{(l+1)} = \alpha_{ir}^{*}$ $(i = 1,\ldots,N)$ which are the optimal coefficients values found from Step 3. Does $r = R$? That is, are we at the highest order yet?

NO: Then $r \leftarrow r+1$ (i.e., replace current order r with r+1);

GO TO STEP 2

YES: Is $|F_l(\alpha_{lr}^{(l+1)}) - F_l(\alpha_{lr}^{(l)})| < \varepsilon$,

where ε is a desired convergence tolerance? That is, are optimal values from the previous cycle close to results of the current cycle?

NO: Then we should start a new cycle:

$$\text{Let } Q_i^{(l+1)}(t) = Q_i^{(l)}(t) + \sum_{j=0}^{R} \alpha_{ij}^{(l+1)} T_j(\xi)$$

This represents the new trajectory we will be perturbing around. $l \leftarrow l+1$ (new cycle) and reset the order counter r back to zero. Now GO TO STEP 2 and we are ready for another cycle.

YES: STOP

In solving the dynamic programming problem of (41) and (42), we select the desired number of discretization intervals M for the α_{ir}. For convenience, an odd number of intervals (≥ 3) is selected so we can define a variable $\bar{\alpha}_{ir} \varepsilon [-1,1]$ and discretize it into M values separated by uniform intervals, making sure that one of the discrete $\bar{\alpha}_{ir} = 0$. This insures that a value on the current trajectory can always be selected if no improvement can be found from a perturbation of it. We then define an appropriate discretization interval ΔQ_i for flow under the gates, and let

$$\alpha_{ir} = \Delta Q_i \cdot \bar{\alpha}_{ir}, \quad i = 1,\ldots,N \tag{43}$$

be defined for each of the M values of $\bar{\alpha}_{ir}$. Thus, the user need only select M and ΔQ_i, and avoids the problem of trying to estimate what an appropriate interval $\Delta\alpha_{ir}$ should be.

The performance of this orthogonal dynamic programming algorithm is documented in Labadie et al. (1980). For a simplified two-dimensional problem, the orthogonal dynamic programming (ODP) found the optimum in about one-half the execution time of incremental dynamic programming. No comparisons were possible for higher dimensional problems because solution by the latter method quickly become computationally infeasible. A new approach called objective-space dynamic programming holds promise as a more powerful replacement for the ODP approach, and is discussed in Chapter V of this report.

F. CASE STUDIES

F.1 North Shore Outfalls Consolidation Project

The SWCP has been applied to the Marina Branch of the North Shore Outfalls Consolidation Project (NSOC) which is near completion in San Francisco. These results are reported in Labadie et al. (1980), Morrow and Labadie (1980), and Labadie et al. (1981), and are only summarized here.

It should be noted that regulatory standards have been altered considerably since our research first began. The Regional Water Quality Control Board has now established detailed requirements that affect system planning and operations. These include:

1. Stipulation of specific allowable annual frequencies of overflows at all outfall locations; these vary according to perceived sensitivity of the receiving water environment to pollution load.

2. Requirement that all storage capacity in the system must be utilized before an overflow is allowed.

3. Maintenance of maximum pumping and treatment plant rates prior to an unavoidable spill or overflow.

These researchers are convinced that it is only with real-time optimal automatic control that these criteria can be met with maximum assurance. Though the NSOC study was conducted by CSU prior to stipulation of these new standards, it is believed that it is still a valid demonstration.

The NSOC is a shoreline interceptor/storage project which includes (a) construction of 3.5 miles of shoreline tunnel for storage and conveyance (total capacity around 3.2 million ft^3), (b) consolidation of many existing sewage outfalls into one or two selected points, and (c) additional wet weather treatment capacity. The locations of the North Point Treatment Plant and North Shore Pump Station are shown in Figure III-8. The Marina Branch of the project runs along the shoreline to the west of the Treatment Plant.

About a 4500 ft portion of this branch running from Baker St. to Laguna St. has been set up for the SWCP with the inclusion of two controllable gates in the system, as shown in Figure III-7. These gates were originally planned for the system, but have been abandoned in the current construction due to budgetary constraints. It is hoped that inclusion of gates or other control structures will be considered as add-on projects in the future. Our research has shown that they can be extremely valuable in making maximum use of tunnel storage, as well as controlling the timing and location of overflows when they are unavoidable.

The Marina Branch of the NSOC is divided into three dynamic programming stages (Figure III-7). Each stage is a separate unsteady flow routing problem requiring the specification of both upstream and

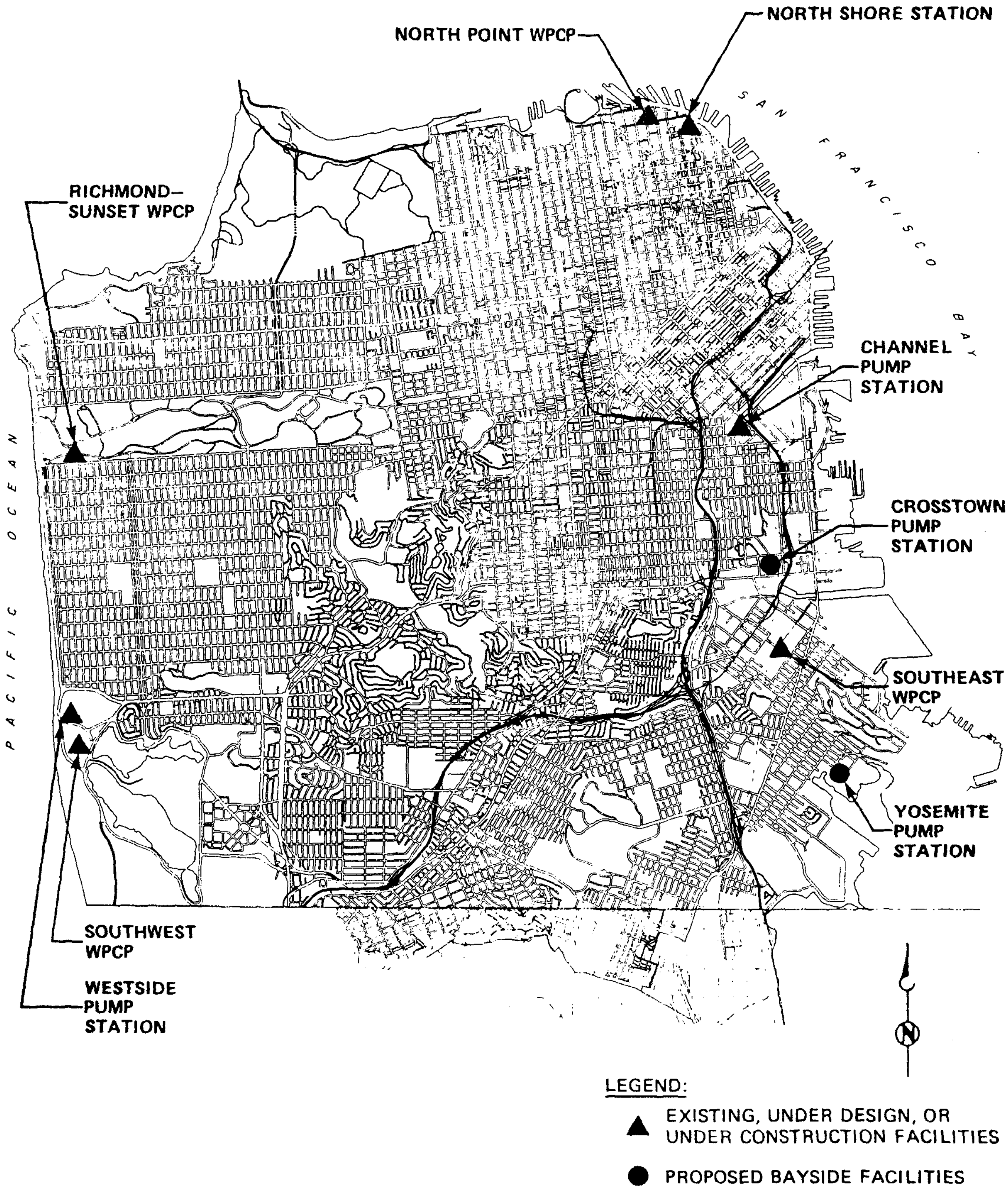


Figure III-8. Location map of major pumping and treatment facilities, City and County of San Francisco (Caldwell-Gonzales-Kennedy-Tudor, 1981).

downstream flows for all time periods (bounary conditions). The flows between each stage are optimally selected by the DP algorithm for each 15 minute control interval, and linearly interpolated for 3 minute routing time steps. Gate openings are a dependent variable calculated at each 15 minute control interval according to the gate equation.

Tests were conducted with the SWCP to ascertain the effect of forecasting and forecast error on the performance of the optimal gate control strategies. "Reactive" policies were also examined as a measure of the absence of any forecasting on the overflow rate. The reactive case simply assumes the current measured inflow rate is constant throughout each 15 minute control interval. Strategies are computed and implemented under this assumption. Actual inflows update the flow rates at the end of each control interval and the process is repeated until the end of the storm. The initial trial policy as a starting point for the DP algorithm produced street flooding with an average weighted overflow rate of 16.4 cms (cubic meters per second). The best-case reactive control policy (no forecasting) produced a weighted rate of 13.0 cms.

"Adaptive" control policies which include forecasting were also tested with the SWCP. The adaptive mode of operation involves a determination of optimal policies over the specified forecast lead time. The optimal policy is executed on the prototype simulation model for the current control interval with the actual data.

The results of adaptive operation of the SWCP for policies derived under forecast error are presented in Table III-1. Initial gate flow trajectories are the same as the reactive strategy experiments so that the results can be compared. Two forecast lead times are presented. The one-control-interval lead time (L=1) test runs are two-dimensional DP problems. The three-control-interval (L=3) forecast lead time test

Table III-1. Effect of Forecasting on Adaptive Operation of the Stormwater Control Package

	Average Overflow Rates in Cubic Meters/Second	
Initial Forecast Error	Forecast Lead Time L=1	L=3
0%	0.85 (30 cfs)	0.85 (30 cfs)[2]
+20%	10.6 (375 cfs)	0.9 (32 cfs)[1]
+70%	16.4 (580 cfs)	10.7 (378 cfs)[3]
-20%	0.74 (26 cfs)[4]	3.5 (123 cfs)
-70%	16.4 (580 cfs) streetflooding[3]	16.4 (580 cfs) streetflooding[3]

[1] two iterations gave better results than three iterations. This is because of the simulated real time nature of this problem and the manner in which the initial best guess trajectories of gate flows are updated at each control interval to match the current optimum trajectory.

[2] maximum polynomial order was always one less than the dimension.

[3] two iterations

[4] this result is suspect, due to control gate linearization errors.

runs are four-dimensional DP problems using three polynomial orders and three "iterations," or reductions of corridor width.

The weighted average overflow rate generally is reduced as the forecast lead time increases and as the forecast error decreases. This demonstrates the effectiveness of forecasting in reducing the average rate of weighted overflow. Positive error represents overestimated inflows; negative error represents underestimated inflows. The forecast model adaptively updates its parameters as the real storm event progresses, which decreases the forecast error for subsequent control intervals. The 0% forecast error is the "perfect foreknowledge" case; that is, assuming the forecasting model is able to exactly predict the incoming storm hydrograph. This was accomplished by providing the real storm to the forecast model in the off-line historical parameter generation step.

All test cases, with the exception of the 20% underestimated inflow case, performed as well or better with the increased forecast lead time. The particular case for L = 1 and the -20% forecast error has several control gate linearization errors in excess of 20% in the prototype routing model. These linearization errors were corrected by the interrupted double sweep solution, but the result remains suspect. The cases for -70% error both result in street flooding, again due in part to gate linearization errors which result since the control algorithm is "surprised" by much larger flows than were forecasted, thereby requiring the gates to be moved rapidly.

The test results indicate that incorporating forecasted inflows into the control policy determination generally increases the accuracy of decisions for the gate settings. The exception is for cases of high forecast error where a rapid change in gate setting (due to large underestimated inflows) is required to avoid an overflo. For moderate levels of forecast error, adaptive policies which utilize forecasts, even for only one control interval ahead, prove superior to reactive policies. For the 0% forecast error case, the longer head time did not improve the overflow rate. Little temporal overflow redistribution is being accomplished by the control model since the objective function for this test case had no time-factor for overflows (time weighting factors were all set to 1.0).

F.2 Bayside Facilities Planning Project

The NSOC case study clearly demonstrates the value of the SWCP for real-time control, but is limited to gate control strategies. As a further demonstration of the SWCP, it was desired to determine if other control elements such as pumping facilities could be optimized by the SWCP. In its current form, the SWCP assumes all control structures are gates. Since gate settings are not directly optimized in the SWCP but actually determined after the model has optimized flows and heads at the control point, it was decided to determine if the SWCP could be "fooled" into controlling pumping through use of a "dummy" gate equation.

For demonstration purposes, the SWCP was applied to a proposed project for pumping excess flows in the NSOC system to the Channel Outfalls Consolidation (COC) Project, since the latter has considerably more storage capacity. The configuration is shown in Figure III-9. The key questions are: when should the wet weather pumps be turned on during a storm event and how should they be controlled in order to avoid overflows. Since turning them on involves considerable expenditure of energy, they should only be used if absolutely needed. With the imperfections in storm inflow forecasting, there is a chance that they would

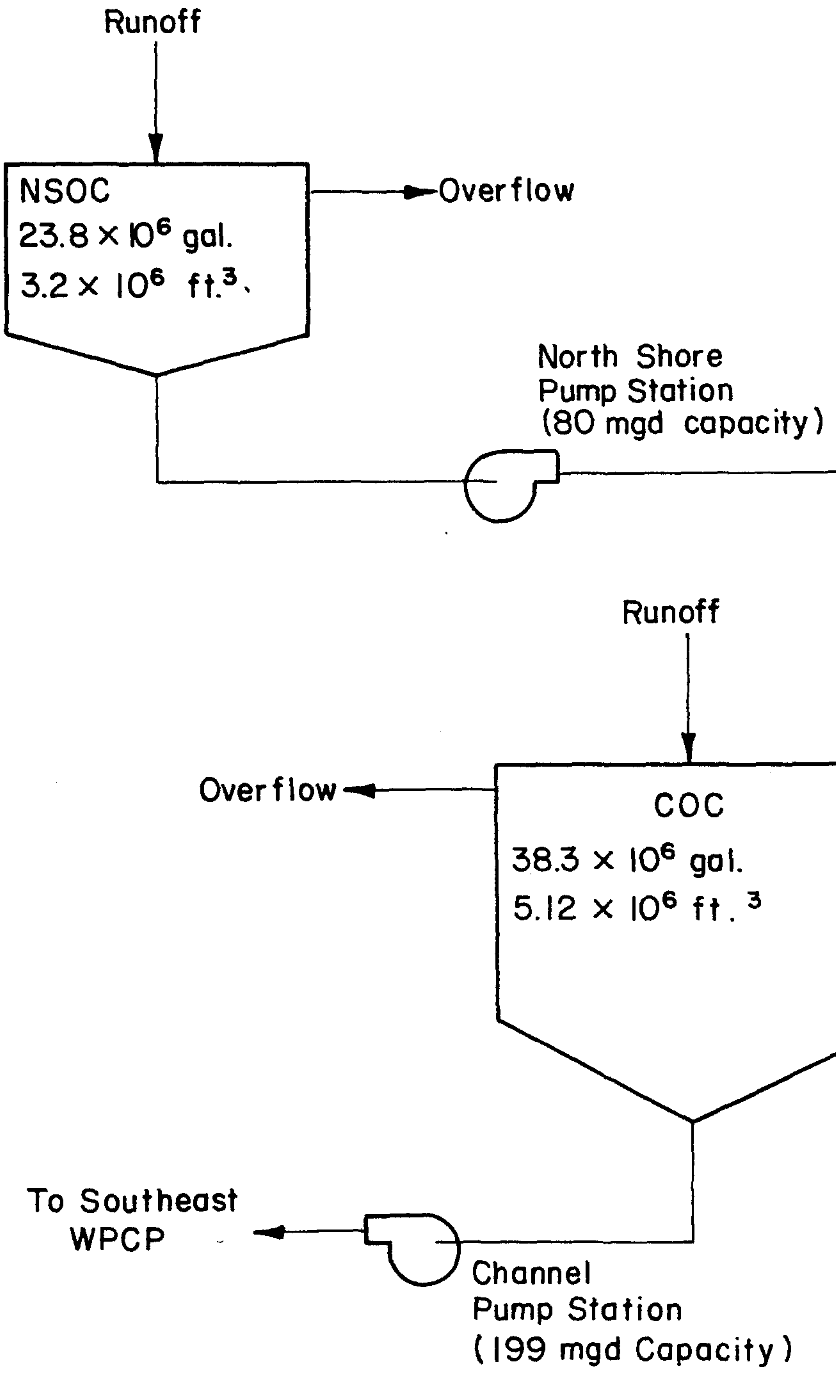


Figure III-9. Proposed transfer of excess storm flows from the North Shore to the channel section of San Francisco.

be turned on during an apparent overflow producing event, only to discover later that it was unnecessary since the storm did not develop as forecasted. On the other hand, there is the danger of delaying too long, and turning the pumps on too late, when an earlier startup would have prevented overflows.

A report by Caldwell-Gonzales-Kennedy-Tudor (CGKT, 1981) presents a supervisory control scheme for the transfer of excess flows from NSOC to COC. A simplified model called SFMAC was applied which essentially uses storage routing only with no dynamic considerations. The CGKT (1981) report shows results of application of SFMAC for what are called reactive, limited predictive and full predictive control. Limited predictive uses a 15 month lead time, whereas full predictive assumes the storm can be perfectly predicted over its entire duration. The results are shown in Figure III-10. The logic for these schemes are described in detail by CKGT (1981) and are not repeated here. They are not based on use of optimization, but on numerous simulation runs by City and County staff and their consultants.

As a demonstration, it was decided to again set up the NSOC project on the SWCP, but without gates, along with the COC project, and simulate the transfer of flow between them by a "dummy gate." The SWCP setup for the system is shown in Table III-2. Note that there are two DP stages in this case; Stage 1 represents NSOC and Stage 2 is the COC. The initial trajectories for flow under the "dummy" gate, which is actually a pump station, were set to the flow transfers computed by the CKGT study, as shown in Figure III-10b and III-10e for the perfect predictive control case. The upper and lower limits on flows were then constrained exactly to these values, which effectively restricts the SWCP to passing exactly that flow. The gate setting that the SWCP

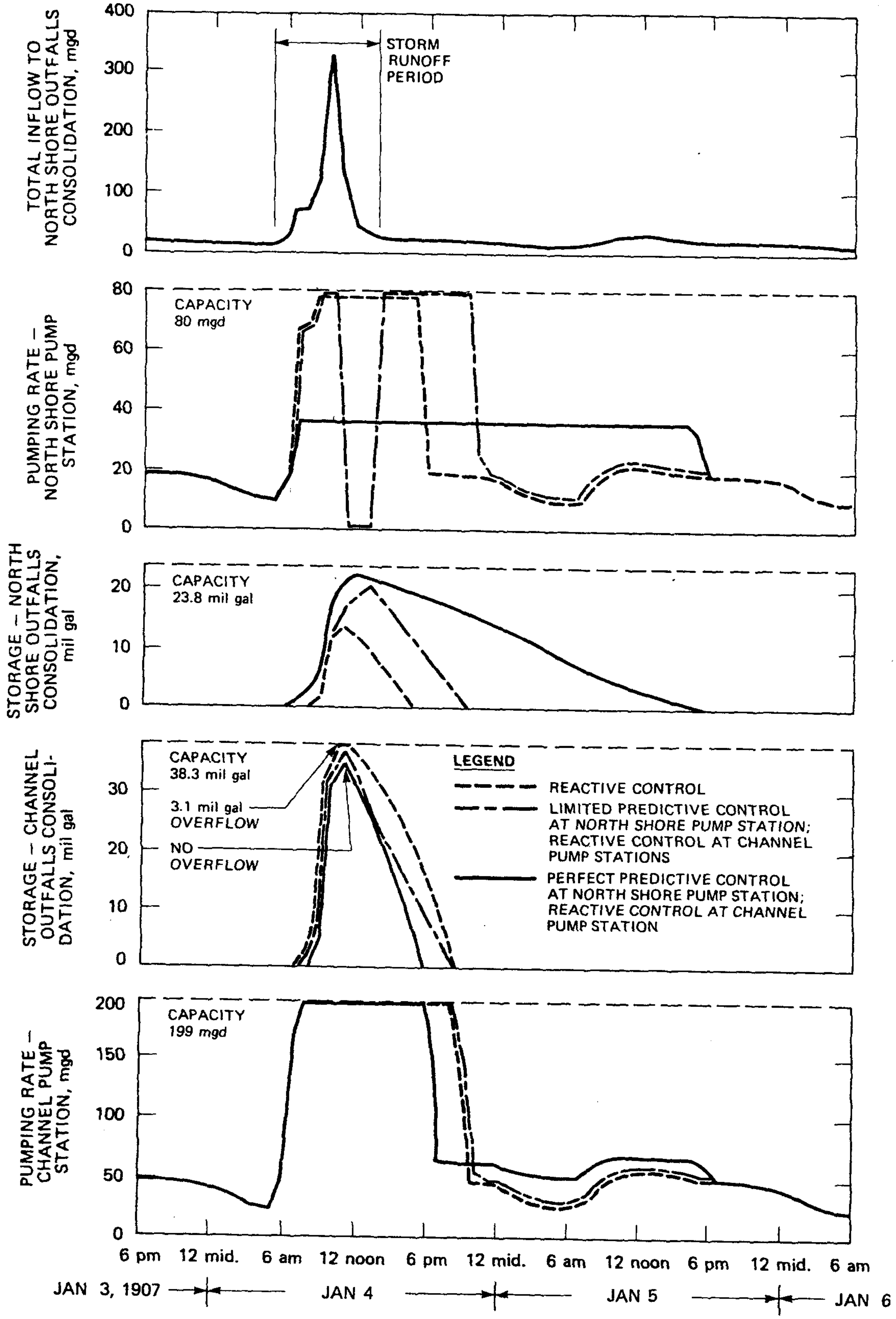


Figure III-10. Comparative operation of CGKT control strategies (CGKT, 1981).

Table III-2. Geometric Configuration of the NSOC (Stage 1) and COC (Stage 2) projects.

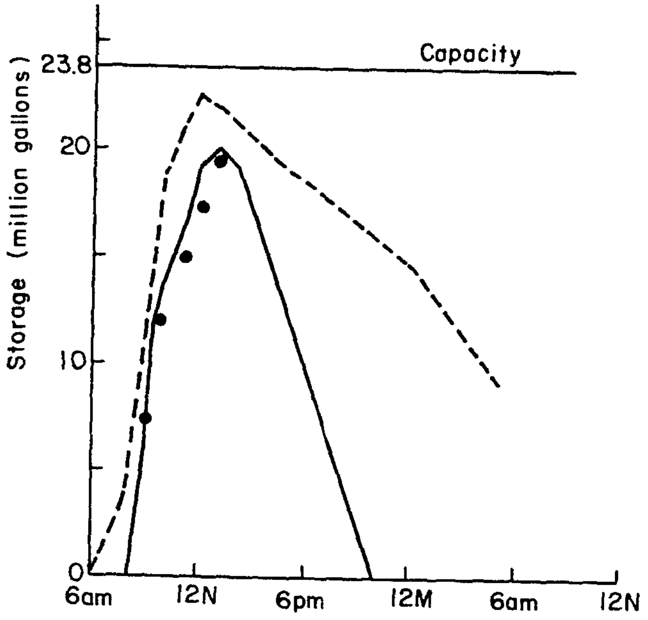
```
CHANNEL GEOMETRY DATA FOR STAGE NUMBER  2:

ECTION      SHAPE        WIDTH      HEIGHT      RANGE       OVERFLOW      BED        INITIAL   WEIGHTING
                                                           SILL HEIGHT  ELEVATION     DEPTH      FACTOR    QO   SO*1000

   1       RECTANGLE     20.00      30.44      6720.00      32.44      -32.440       15.00      100.   1288.  2.000
   2       RECTANGLE     20.00      30.46      6730.00      30.45      -32.460       15.00       90.   1288.  2.000
   3       RECTANGLE     20.00      30.48      6740.00      32.48      -32.480       15.00      100.   1288.  2.000
   4       RECTANGLE     20.00      31.48      7240.00      33.48      -33.480       15.00      100.   1288.  2.000
   5       RECTANGLE     20.00      33.48      8240.00      35.48      -35.480       15.00      100.   1288.  2.000
   6       RECTANGLE     20.00      35.48      9240.00      37.48      -37.480       15.00      100.   1288.  2.000
   7       RECTANGLE     20.00      37.48     10240.00      39.48      -39.480       15.00      100.   1288.  2.000
   8       RECTANGLE     20.00      39.48     11240.00      41.48      -41.480       15.00      100.   1288.  2.000
   9       RECTANGLE     20.00      41.48     12240.00      43.48      -43.480       15.00      100.   1288.  2.000
  10       RECTANGLE     20.00      43.48     13240.00      45.48      -45.480       15.00      100.   1288.  2.000
  11       RECTANGLE     20.00      44.28     13640.00      46.28      -46.280       15.00      100.   1288.  2.000
   0.0000

CHANNEL GEOMETRY DATA FOR STAGE NUMBER  1:

ECTION      SHAPE        WIDTH      HEIGHT      RANGE       OVERFLOW      BED        INITIAL   WEIGHTING
                                                           SILL HEIGHT  ELEVATION     DEPTH      FACTOR    QO   SO*1000

   1       RECTANGLE     20.00      17.00         0.00      14.00      -19.000        6.00        1.   1288.  2.000
   2       RECTANGLE     20.00      17.02        10.00      19.02      -19.020        6.00      100.   1288.  2.000
   3       RECTANGLE     20.00      18.42       710.00      20.42      -20.420        6.00      100.   1288.  2.000
   4       RECTANGLE     20.00      19.82      1410.00      21.82      -21.820        6.00      100.   1288.  2.000
   5       RECTANGLE     20.00      21.22      2110.00      23.22      -23.220       10.00      100.   1288.  2.000
   6       RECTANGLE     20.00      22.62      2810.00      24.62      -24.620       10.00      100.   1288.  2.000
   7       RECTANGLE     20.00      24.02      3510.00      26.02      -26.020       10.00      100.   1288.  2.000
   8       RECTANGLE     20.00      25.42      4210.00      27.42      -27.420       10.00      100.   1288.  2.000
   9       RECTANGLE     20.00      26.82      4910.00      28.82      -28.820       10.00      100.   1288.  2.000
  10       RECTANGLE     20.00      28.22      5610.00      30.22      -30.220       13.00      100.   1288.  2.000
  11       RECTANGLE     20.00      29.62      6310.00      31.62      -31.620       13.00      100.   1288.  2.000
  12       RECTANGLE     20.00      30.42      6710.00      32.42      -32.420       13.00      100.   1288.  2.000
  13       RECTANGLE     20.00      30.44      6720.00      32.44      -32.440       15.00      100.   1288.  2.000
  14       RECTANGLE     20.00      30.46      6730.00      32.46      -32.460       15.00      100.   1288.  2.000
   0.0000
```

computes is of course meaningless since there actually is no gate there.

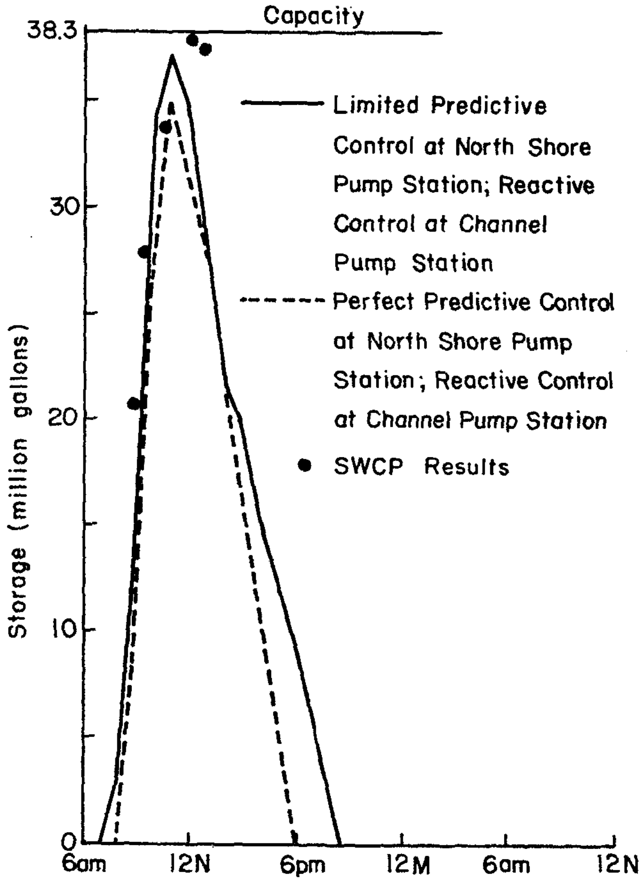
If the SWCP can simulate a controlled pumping operation in this way, then we should be able to match the storage curves for the perfect predictive case in Figures III-10c and III-10d. Figure III-11 gives the results, which show that the SWCP effectively matches the storage curves. The SWCP predicts a slightly higher storage in the COC, but the match is quite good with no overflows occurring. We only carried the computations to the peak storage levels since this was felt to be sufficient to demonstrate the validity of our assertion.

This confirms that the SWCP can use a "dummy" gate to simulate a controlled pumping operation. However, this demonstration did not exploit the optimizing capability of the SWCP since we were using pumping rules developed by trial and error simulation. To test the optimizing capability of the SWCP, we increased the hydrograph of Figure III-10a by 50% and ran the SWCP with no restrictions on pumping except for the 80 mgd maximum capacity limit.

For the L = 3 case, application of the SWCP resulted in a pump control policy that reduced overflows to zero from an initial reactive policy that would have produced significant overflows.



(a) North Shore Outfalls Consolidation Project (NSOC)



(b) Channel Outfalls Consolidation Project (COC)

Figure III-11. Comparison of SWCP and CKGT results for controlled pumping of flows from NSOC to COC.

CHAPTER IV

OPTIMAL LAYOUT AND SIZING OF STORM SEWER SYSTEMS

A. PROBLEM STATEMENT

Considerable research has been conducted on the development of algorithms which for a given horizontal layout of a storm sewer system (i.e., location of manholes and specification of pipe linkages) can determine the least cost depth of manholes, pipe sizes, and pipe slopes. Examples would include the work of Holland (1966), Zepp and Leary (1969), Dajani and Hasit (1974), Froise et al. (1975), and Robinson and Labadie (1981). Linear programming, dynamic programming, nonlinear programming, separable programming, and mixed integer programming have all been employed for solving this problem. Of all of the methods, dynamic programming appears to be the most popular because of its ability to find global optimal solutions in the presence of complex, highly nonlinear pipe and manhole cost functions, as well as the ability to include nonlinear sewer flow routing.

There have also been a number of papers describing techniques for optimal horizontal layout of sewer systems in order to find least cost spanning trees for a proposed sewer network. This would include the work of Liebman (1967), Barlow (1972), Lowsley (1973), and Mandl (1981). Linear programming and network flow theory have been the most commonly used methods for this problem.

Work which has attempted to integrate these two problems together (i.e. coordinated solution of the least cost horizontal layout and vertical sizing problems) has been much more limited. The exception would be the work at the University of Illinois at Urban-Champaign conducted by Mays and Wenzel (1976), Wenzel et al. (1979), and Wenzel (1980). The methodology they employed is a variant of dynamic programming called discrete differential dynamic programming (DDDP). They also included realistic sewer routing and risk considerations in the model development, as well as sizing of detention storage in the sewer system. The primary disadvantages of the DDDP approach include dimensionality difficulties that intensify as the size and complexity of the sewer network increase, and the need to specify apriori a unique solution 'stage' for each manhole in the network. This latter requirement is particularly difficult for large networks.

In order to overcome these weaknesses, a solution procedure is proposed which effectively solves the dimensionality problem and does not require apriori specification of manhole 'stages' for horizontal layout. The network involves tentatively solving separate vertical sizing and horizontal layout optimization problems, with eventual convergence to a local optimal solution of the combined problem. Dynamic programming is employed for the vertical sizing problem and network flow theory is used for the horizontal layout problem. They are linked together by a search procedure founded in nonlinear programming theory similar to an algorithm developed by Frank and Wolfe (see Canon and Callum, 1968). In effect then, the proposed algorithm combines linear programming, nonlinear programming, and dynamic programming together in such a way as to accentuate their advantages in solving certain aspects of the overall problem. For this current work, we have ignored routing, risk considerations, and inclusion of detention storage. Future work will attempt to include these elements.

B. PROBLEM FORMULATION

The vertical sizing problem for a given layout is described in Appendix C of this report in a paper by Robinson and Labadie (1981). The horizontal layout problem is formulated as follows:

$$\min_{\underline{q} \geq \underline{0}} \sum_{i=1}^{N} \sum_{j=1}^{N} c^*_{ij}(\underline{q}) q_{ij} \tag{1}$$

$$\sum_{i=1}^{N} q_{ji} - \sum_{i=1}^{N} q_{ij} = q_j \quad (j=1,\ldots,N) \tag{2}$$

where

q_{ij} = directed flow in link (i,j) as determined by the horizontal layout problem ($\underline{q}$ is the NxN vector representing flows in all links).

q_j = design storm inflow to node or manhole j for given storm return period.

N = total number of manholes or nodes

$c^*_{ij}(\underline{q})$ = nonlinear cost function computed by solving the least cost vertical sizing problem for a given layout and connectivity of the network, as defined by vector $\underline{q}$, and associated minimal costs $\underline{c}^*(\underline{q})$. Costs include manhole and pipe costs, as well as right-of-way, excavation, and placement.

The flows q_{ij} themselves can define the connectivety of the network, where if $q_{ij} = 0$ for link (i,j), then it is not connected; it is also assumed that a spanning tree-like structure with no loops in the system is maintained. A spanning tree for a network is simply defined such that all nodes in the network must have exactly one exiting link or pipe connection.

This optimization problem implies that for each selected layout or spanning tree as defined by vector $\underline{q}$, the vertical sizing problem is solved to compute the minimal associated costs $c^*_{ij}(\underline{q})$ per unit discharge in each pipe link (i,j) in the spanning tree. The horizontal problem is to find the least cost layout $\underline{q}$.

C. FRANK-WOLFE ALGORITHM

The coordinating algorithm proposed for this problem is the Frank-Wolfe method. Consider the following problem, using general mathematical programming terminology:

$$[A] \qquad \min_{\underline{x} \geq \underline{0}} \; f(\underline{x}) \tag{3}$$

subject to:

$$A\underline{x} \geq \underline{b} \tag{4}$$

where f(.) is a real valued, differentiable function. Now define Problem B as:

$$[B] \qquad \min_{\underline{\tilde{x}} \geq \underline{0}} \; \nabla f(\underline{\tilde{x}}^k)\underline{x} \tag{5}$$

subject to:

$$A\underline{x} \geq \underline{b} \tag{6}$$

for some given $\underline{\tilde{x}}^k$ at current iteration k.

Solution of Problem B, which is essentially a linear programming problem if $\underline{\tilde{x}}^k$ is given, yields an optimal $\underline{x}*$ which is defined as $\underline{\tilde{x}}^k$, or

$$\underline{x}^k = \underline{x}^* \tag{7}$$

By definition of an optimum

$$\nabla f(\underline{\tilde{x}}^k)\underline{x} \geq \nabla f(\underline{\tilde{x}}^k)\underline{x}^k \tag{8}$$

for all $\underline{x}$ satisfying:

$$A\underline{x} \geq \underline{b} \tag{9}$$

$$\underline{x} \geq \underline{0} \tag{10}$$

or

$$\nabla f(\tilde{\underline{x}}^k)[\underline{x}^k - \tilde{\underline{x}}^k] \leq 0 \tag{11}$$

Notice that direction vector

$$\underline{d}^k = [\underline{x}^k - \tilde{\underline{x}}^k] \tag{12}$$

is by definition a feasible direction since the directional derivative is ≤ 0 and the constraint set is a convex polyhedron.

Now let

$$\tilde{\underline{x}} = \tilde{\underline{x}}^k + \alpha\, \underline{d}^k \tag{13}$$

and solve

$$\min_{\alpha} \; f(\tilde{\underline{x}}) \tag{14}$$

This is a simple one-dimensional search problem over scalar step size α which can be solved by the Golden Section method or other techniques. We obtain α^* and let

$$\tilde{\underline{x}}^{k+1} = \tilde{\underline{x}}^k + \alpha^*\, \underline{d}^k \tag{16}$$

We now replace iteration counter k with k+1 (i.e. k ← k+1) and solve Problem B again. It can be shown that this procedure will converge to a Kuhn-Tucker point:

proof

There are three possibilities:

[1] $\underline{x}^* = \underline{x}^k = \tilde{\underline{x}}^k$

Then

$$\nabla f(\tilde{\underline{x}}^k)\underline{x} \geq \nabla f(\tilde{\underline{x}}^k)\underline{x}^k = \nabla f(\tilde{\underline{x}}^k)\tilde{\underline{x}}^k$$

or

$$\nabla f(\tilde{\underline{x}}^k)[\underline{x} - \tilde{\underline{x}}^k] \geq 0$$

for all feasible $\underline{x}$. This is by definition a Kuhn-Tucker point because it indicates the objective function cannot locally improve in any feasible direction. [Note: It is possible for the Kuhn-Tucker conditions to be satisfied at saddle points, which are not true local optima.]

[2] $\underline{x}^* = \underline{x}^k \neq \tilde{\underline{x}}^k$, but

$$\nabla f(\tilde{\underline{x}}^k)[\underline{x}^k - \tilde{\underline{x}}^k] = 0$$

since

$$\nabla f(\tilde{\underline{x}}^k)\underline{x} \geq \nabla f(\tilde{\underline{x}}^k)\underline{x}^k \geq \nabla f(\tilde{\underline{x}}^k)\tilde{\underline{x}}^k$$

for all feasible $\underline{x}$, then this is also a Kuhn-Tucker point.

[3] $\underline{x}^k \neq \tilde{\underline{x}}^k$ and

$$\nabla f(\tilde{\underline{x}}^k)[\underline{x}^k - \tilde{\underline{x}}^k] < 0$$

This means we should continue to iterate because a better point can be found.

Note that possibility [3], and hence convergence of the algorithm, actually does not depend on minimizing $f(\tilde{\underline{x}}^k + \alpha\underline{d}^k)$ w.r.t. α, but simply making sure that we find an α such that

$$f(\tilde{\underline{x}}^k + \alpha\underline{d}^k) < f(\tilde{\underline{x}}^k)$$

We will use the Frank-Wolfe method as the basis of our coordinating solution procedure, but with some modification.

D. SOLUTION PROCEDURE

For our problem, Problem A is:

$$\min_{\underline{q}>0} \sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij}(\underline{q})q_{ij} \tag{17}$$

$$\sum_{i=1}^{N} q_{ij} - \sum_{i=1}^{N} q_{ji} + q_j = 0 \quad \text{for } j=1,\ldots,N \tag{18}$$

Note that it should be possible to change the equal constraints in (18) above to $\geq$ constraints in order to conform with the general Problem A formulation of equation 4. The final solution should have all 'tight' constraints since it would never be optimal to size a pipe larger than it needs to be. Problem B is:

$$\min_{\underline{q} \geq \underline{0}} \sum_{i=1}^{N} \sum_{j=1}^{N} \{ \sum_{l=1}^{N} \sum_{m=1}^{N} [\frac{\partial c^*_{lm}(\tilde{\underline{q}}^k)}{\partial q_{ij}} \tilde{q}^k_{lm} + c^*_{lm}(\tilde{\underline{q}}^k) \frac{\partial \tilde{q}_{lm}}{\partial q_{ij}}]\} q_{ij}$$

subject to the same constraints, where $\tilde{\underline{q}}^k$ is some given set of flows (and hence layout) at iteration k. Again, we assume that the positive components of the vector $\tilde{\underline{q}}^k$ form a spanning tree in the network. In solving this problem, we need to realize that cost function $c^*_{ij}(\cdot)$ is not just an ordinary function. It is actually minimal in the sense that the horizontal layout defined by flow vector $\tilde{\underline{q}}^k$ has been optimized in the vertical such that $c^*_{ij}(\cdot)$ are unit costs derived from the least cost sizing of pipes and specification of slopes for the given layout.

We now make the following critical assumption. Since the cost term in Problem B above is defined for a given layout, we assume that

$$\frac{\partial c^*_{lm}(\tilde{\underline{q}}^k)}{\partial q_{ij}} = 0 \text{ for all } (l,m) \text{ and } (i,j) \tag{20}$$

$$\frac{\partial \tilde{q}^k_{lm}}{\partial q_{ij}} = 1 \text{ for } (l,m) = (i,j) \tag{21}$$

$$\frac{\partial \tilde{q}^k_{lm}}{\partial q_{ij}} = 0 \text{ for all } (l,m) \neq (i,j) \tag{22}$$

This is justified in that it would be impossible to differentially change the flow in any link (i,j) for a given layout $\tilde{q}^k$ without creating an infeasibility and violation of mass balance in the network. This leaves Problem B in the form:

$$\min_{\underline{q} \geq \underline{0}} \sum_{i=1}^{N} \sum_{j=1}^{N} c^*_{ij}(\tilde{\underline{q}}^k) q_{ij} \tag{23}$$

subject to the same constraints. This problem can be easily solved by network flow theory. We let the optimal solution $\underline{q}^*$ be defined as $\underline{q}^k$, which gives a new system layout. Then

$$\tilde{\underline{q}}^{k+1} = \tilde{\underline{q}}^k + \alpha[\underline{q}^k - \tilde{\underline{q}}^k] \tag{24}$$

where $[\underline{q}^k - \tilde{\underline{q}}^k] = \underline{d}^k$ = feasible direction.

An α is selected such that

$$\sum_{i=1}^{N} \sum_{j=1}^{N} c^*_{ij}(\tilde{\underline{q}}^{k+1}) \tilde{q}^{k+1}_{ij} < \sum_{i=1}^{N} \sum_{j=1}^{N} c^*_{ij}(\tilde{\underline{q}}^k) \tilde{q}^k_{ij} \tag{25}$$

Strictly following the Frank-Wolfe procedure, we would compute $c^*_{ij}(\tilde{\underline{q}}^{k+1})$ and solve Problem B again. However, to compute $c^*_{ij}(\tilde{\underline{q}}^{k+1})$, we would have to solve the vertical sizing poblem again for the new flows $\tilde{\underline{q}}^{k+1}$. Unfortunately, there is no guarantee that the new flows as calculated by equation 24 will form a spanning tree. It can be shown, however, that flows generated by solving Problem B will always form a spanning tree.

This is evident when considering that if we have N nodes, then we also have N constraints. Assuming storm inflow is input to each node, then each node must have at least one exiting link. In linear programming, if there are N constraints, then at most N variables (i.e. the link flows q_{ij}) can be positive, which are called basic variables. Therefore, the basic variables indeed constitute a spanning tree as previously derived.

Because of the need to maintain flows that constitute a spanning tree, and to avoid redoing the vertical sizing problem in the same iteration, we will use a first order approximation for the function $c^*_{ij}(\tilde{q}^{k+1}_{ij})$. That is, we let

$$c^*_{ij}(\tilde{\underline{q}}^{k+1}) = c^*_{ij}(\tilde{\underline{q}}^{k}) + \alpha[c^*_{ij}(\underline{q}^{k}) - c^*_{ij}(\tilde{\underline{q}}^{k})] \qquad (26)$$

This departure from the Frank-Wolfe algorithm brings into question whether we can prove convergence to a local optimum. Notice that the new system layout is $\tilde{\underline{q}}^k$ and the previous layout is $\underline{q}^k$. Costs $c^*_{ij}(\underline{q}^k)$ have been calculated by solving the vertical sizing problem using the new layout $\underline{q}^k$. If the new layout suggests an increase in unit cost over the old layout for a particular link, then we change the link cost by step size α in the positive direction; otherwise, we decrease the link cost by the same step size. This algorithm should converge as long as step size α is carefully selected to maintain (25) during intermediate iterations. Computational experience has shown that it is best to begin with very low initial costs for the layout model because once a high unit cost is obtained for a particular link, it may not be possible to ever bring back that link into a layout via solution of Problem B, even though it might eventually be desirable link in combination with other positive flow links. Care must also be taken in selection of step size α.

E. COMPUTATIONAL EXPERIENCE

The dynamic programming model CSUDP/SEWER, developed at Colorado State University is used to optimize the vertical alignment of a storm sewer pipeline design with system cost and hydraulic feasibility as criteria. More detail on this program can be found in Appendix C of this report along with a user manual in Appendix D. Modifications to this package have allowed cost per stage (i.e. manhole plus downstream pipe) to be output, thereby providing the unit costs $c^*_{ij}(\tilde{q})$ by taking the total cost for that stage and dividing by the flow in that stage $\tilde{q}_{ij}$. The network code KILTER which uses the out-of-kilter algorithm was selected to solve Problem B; i.e., the horizontal layout problem. Manual iterations were required for transferring cost information between CSUDP/SEWER and KILTER. Future work will attempt to automate this process.

E.1 CSUDP/SEWER Setup

The vertical storm sewer design problem has been developed for systems where up to three upstream pipes may enter a given manhole drained by only one pipe. Drops in manholes may be fixed by the user, allowed to vary by the program, or disallowed by the user. Other contraints on sewer flow are allowable minimum and maximum flow velocities. Should a calculated pipe velocity fall outside the velocity constraints, the solution is still considered feasible; however, a large penalty is associated with this solution, rendering it less attractive than other feasible solutions found.

The available commercial pipe set by default is 8 to 72 inches in industry standard increments; however, any alternative pipe set can be introduced into the model in the data input. Pipe roughness is specified by the Manning's n coefficient, and the type of flow can be selected to be ASCE standard pipefull discharge or a variable n type solution procedure as outline in ASCE Manual No. 37.

Computational discretization of pipe crown elevations is allowed to be set initially to a coarse value and refined in subsequent calculations to a specified tolerance. Optimizations are performed on all feasible, discrete pipe crown elevations, with Manning's equation used to select the smallest commercially available pipe diameter that will carry the required flow, while maintaining velocity and cover constraints.

1. Manhole Numbering

The correct sequential numbering of manholes is critical to proper use of CSUDP-SEWER. The user must adhere to the following convention or unpredictable results may occur. The user must develop the numbering based on a single trunk (main) line with branches extending from the trunk.

Following this procedure should assure correct numbering:

1. The most downstream manhole must be numbered 1.
2. Manholes are then numbered sequentially, proceeding upstream as far as possible.
3. Branches that have been bypassed in the above serial numbering are then similarly numbered in increasing remoteness from manhole number 1 until all manholes are numbered.
4. No manhole numbers may be omitted or duplicated and the last manhole numbered must be equal to the total number of manholes in the network.

2. Solution of the Pipe Selection Problem

The program begins at the highest numbered manhole. It evaluates all feasible solution configurations, while retaining the optimal family of solutions and inimal accumulated cost for all discrete crown elevations. These are used for the next stage as it proceeds downstream. Calculations proceed from the highest numbered manhole until manhole 1 is reached, with a family of optimal solutions retained in memory. When a branch junction is reached, the optimal solutions for each discrete crown elevation of the incoming pipe is gained in memory. As the algorithm proceedsto another branch, that junction will likely be encountered later. The stored policies will be utilized to find the minimum total cost with respect to all incoming branches at that junction.

A traceback procedure advances upstream and finds the best overall solution from the retained optimal solution set, yielding an optimal solution for the current crown elevation discretization interval. Since this initial interval may be coarser than ultimately desired, the process is now repeated using a finer interval. This continues until the desired order of accuracy is attained. The combination of a coarse initial interval with the power of this interval splicing capability allows the code to quickly converge to the desired level of accuracy in establishing pipe elevations, slopes, and pipe diameters. If certain constraints such as minimum and maximum velocities and pipe cover constraints must be slightly violated in order to find a feasible layout, the program will allow this.

3. Cost Functions

The pipe cost function CPIP is user designed (see Appendix D) and may be represented by 'look up' tables or functional relationships. Functional relationships are of course easier to use if a good fit to tabular cost data can be found. This is of course up to the user. Parameters are passed to the CPIP function from Subroutine OBJECT and for a given stage are as follows:

DX	Downstream pipe crown depth,	feet
DX1	Upstream pipe crown depth,	feet
DIA	Pipe diameter,	inches
PIPTHK	Pipe wall thickness	inches

It is convenient to include the excavation cost associated with the pipe in this routine by calculating an average cover and assumed trench width. Other parameters could be passed from OBJECT; however, modification to the code would be required.

The user supplied manhole cost CMNH function is similarly structured with passed parameters as follows:

DX	Downstream pipe crown depth,	feet
DX1	Upstream pipe crown depth,	feet
DIA	Pipe diameter,	inches

Exceedance of cover limits is permitted in order to fit a pipe into a deep setting. However, the user is informed in the output of this violation by a # symbol. Similarly, a symbol is printed near a drop manhole if one is selected.

The typical manhole cost is substantially less than the comparative pipe cost. Hence, CSUDP/SEWER tends to lay pipes as shallow as possible and, if permitted, produce manhole drops. If the user does not wish to allow drop conditions, then the model seeks the most cost effective solution in balancing excavation, pipe, and manhole costs, under this restriction.

E.2 KILTER Setup

KILTER is a network flow code that uses the Ford-Fulkerson (1962) Out-of-Kilter Algorithm for finding the minimum total cost of flow in a network. The version employed for this work restricts flows to integer quantities, which allows more efficient computation. It is interesting that though integer programming methods are generally less efficient than their continuous variable counterparts, just the opposte is true for the out-of-kilter method.

1. Algorithm Characteristics

- Mass balance of flow must be maintained at each node, and throughout the connected arcs.
- Parallel arcs with individual arc costs are possible.
- The model must begin with flows that satisfy mass balance, but may not satisfy upper and lower bounds on flows. A good choice is to set all flows initially to zero.
- Flow direction and costs in an arc may be positive or negative.
- All arcs must be bounded from below and above.
- Fully circulation Networks are required with no sources, sinks, or side constraints allowed.
- Non-circulation networks can be transformed into circulation networks with the addition of at most one node and N return arcs, where N is the number of sources and sinks in the network.
- Network size is currently defined for a maximum of 100 nodes and 100 connecting arcs. However, array dimensions can be easily increased.

2. Input Requirements

Card No.	Units
1. Title Card [80 character description]	
2. Network Configuration	
* No. Arcs	(real + return)
* No. Nodes	(real + artificial)
* Accuracy tolerance for arc cost	$//Q
3. Connection	
* Initial node number	
* Terminal node number	
4. Flow Boundaries	
* Lower flow bound on arc i	+/- Q
* Upper flow bound on arc i	+/- Q

- Signed magnitude of flow in arc i, usually set initially to zero. +/- Q
- Cost per unit flow in arc i +/- $/Q

3. Application of KILTER to Horizontal Sewer Layout

1. The user must develop a cost estimation of the manhole (node) and pipe (arc) costs. Typically, this is accomplished from an initial horizontal layout selected by engineering judgement. The arcs not included in the initial layout are assigned a zero or low cost so that KILTER can include them as candidates for the new configuration to be found. If the costs are initially set too high, KILTER may never bring them into the solution.

2. Construct N return arcs from the outlet node to all nodes, including itself if it has a surface inflows.

- Arc costs per unit flow = 0
- Lower flow bound = Upper flow bound = Surface inlet discharge
- The system outlet node will be the initial node that joins to each of the real nodes in the flow network by the return arcs.

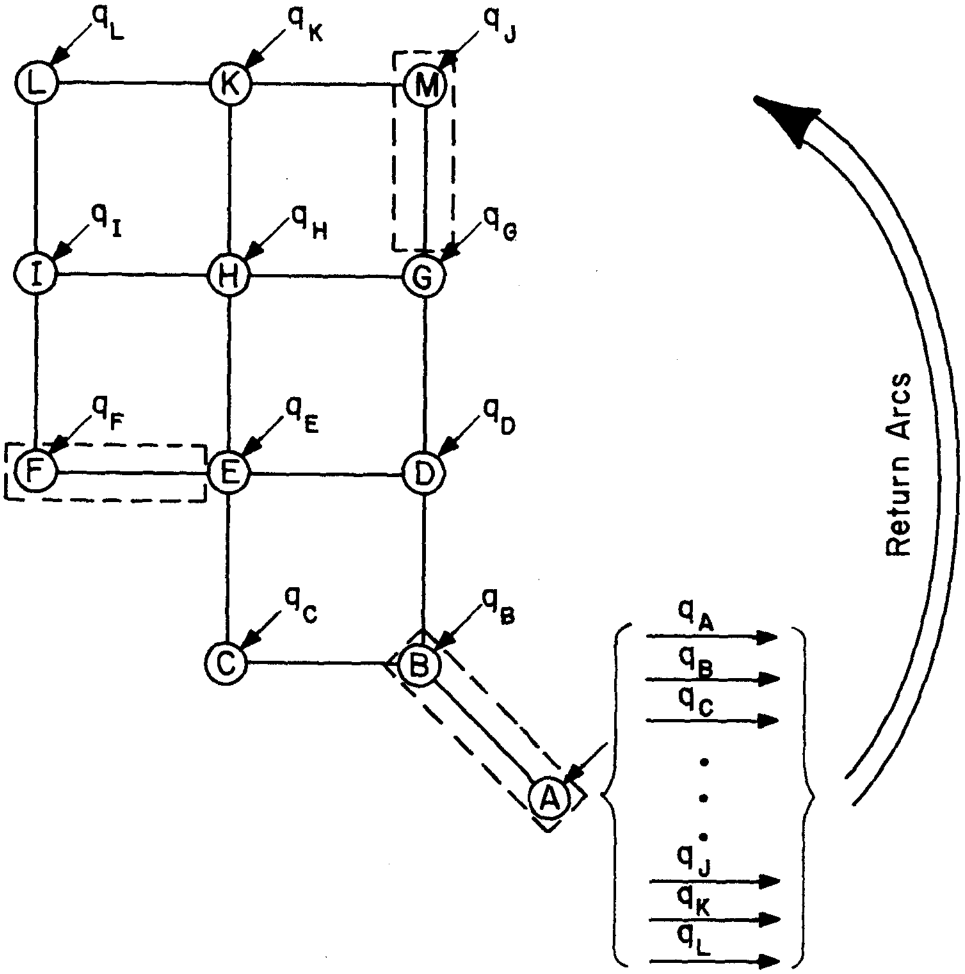
Figure IV-1 demonstrates this arrangement.

E.3 Example Problem

1. Problem Description

The problem we chose to study mainly for demonstration purposes is Example A from ASCE Manual No. 37. This problem is also used in a paper by Wenzel (1980). The problem as described has 12 manholes and is shown on the schematic of Figure IV-2. The schematic diagram indicates all of the potential flow paths. An optimal, least-cost solution was found by Wenzel (1980) using the DDDP approach. It was decided to deliberately start with a different, higher cost layout, and see if the algorithm we devised could converge to the optimal layout.

Each manhole location is identified with a letter which never changes. However, the numbering sequence for the manholes can change,



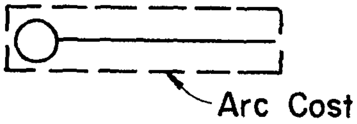


Figure IV-1. Example flow network configuration for KILTER showing return arcs and arc cost definition.

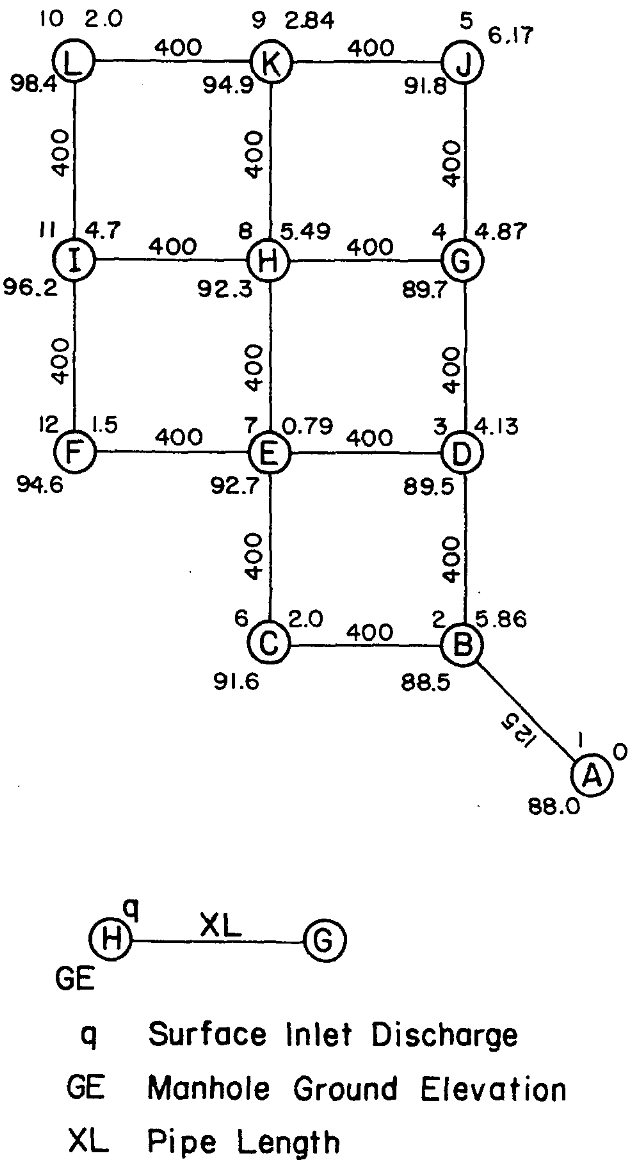


Figure IV-2. Configuration of Example Problem A, ASCE Manual No. 37.

according to the previously defined procedure for manhole numbering. The inflow into the system at each manhole is shown on Figure IV-2 also. This surface inflow is added to the flow(s) in pipe(s) entering that manhole and the total is the flow in the pipe exiting the manhole. Mass balance at each manhole is maintained, and no routing is included. The possibility of detention storage could be easily added if functions were available relating peak flow reduction with detention storage cost.

The cost functions used in CSUDP/SEWER are based on manhole costs and combined pipe and excavation costs from ASCE Manual No. 37. These data are given in tabular form in Table IV-1, from which functional relationships were fitted for case of use in the computations.

2. Solution Method

The method of solving for the least-cost horizontal and vertical design involves iterating back and forth between the two optimization programs CSUDP/SEWER and KILTER. The procedure and comments on this problem are described in the following steps and illustrated in the following worksheets.

1. Choose an initial, feasible horizontal layout. Again, the initial one selected was deliberately chosen to be different from the optimal one in Wenzel (1980) in order to see if the algorithm can converge to it. Both are shown in Figure IV-3a. The optimum layout is given in Worksheet #1, and the starting Layout #2 is given in Worksheet #2. We first choose an initial step size α ($\alpha = 0.5$ in this case), and initially set all arc costs to zero.

2. Using the initial horizontal layout, run CSUDP/SEWER to obtain the optimal vertical alignment and pipe sizes for that layout, and the associated element (manhole and combined pipe and excavation) costs for arcs in that layout. The total cost for the system is given also in the

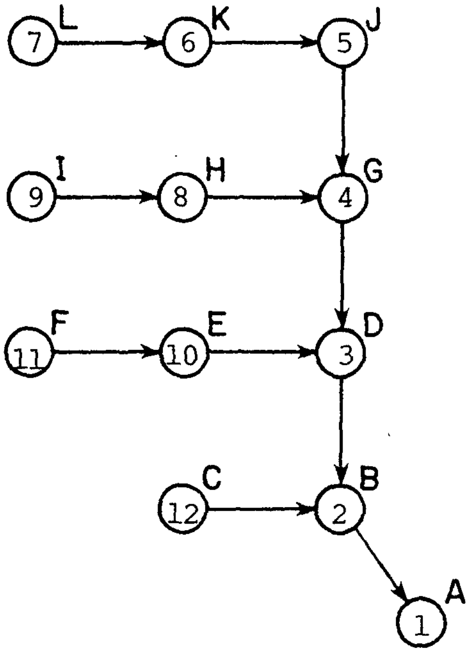
Table IV-1. User supplied cost data for example problem.

```
PIPE COST
        Class 1  Class 2                        Class 1  Class 2
Max Hbar:      4     none                               4     none
                                           Log      Log      Log
P. Dia    Cost     Cost                 P. Dia    Cost     Cost
 inches  $/lin.ft.$/lin.ft.              inches  $/lin.ft.$/lin.ft.
----------------------------            ---------------------------
      12     3.40     3.40                 1.079    0.531    0.531
      15     4.45     4.45                 1.176    0.648    0.648
      18     5.90     5.90                 1.255    0.771    0.771
      21     7.40     7.40                 1.322    0.869    0.869
      24     9.20     9.20                 1.380    0.964    0.964
      27    10.25    11.05                 1.431    1.011    1.043
      30    13.15    14.20                 1.477    1.119    1.152
      36    18.40    19.05                 1.556    1.265    1.280
      42    24.10    25.00                 1.623    1.382    1.398
      48    30.85    32.45                 1.681    1.489    1.511
      54    37.95    39.45                 1.732    1.579    1.596

MANHOLE COST
       0   100.00
all depths
```

Model	Class	r^2	a	b
Linear:	1	0.9717	-9.4173	0.8215
$y = a+bx$	2	0.9744	-10.0616	0.8631
Exponential:	1	0.9781	2.1042	0.0569
$y = a\ e^{bx}$	2	0.9730	2.0867	0.0581
* Power:	1	0.9945	0.0533	1.6313
$y = a\ x^{b}$	2	0.9960	0.0482	1.6707

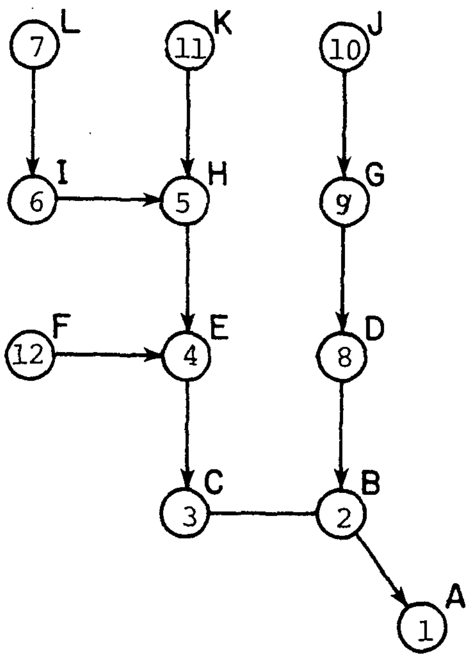
*selected model

CSUDP/SEWER - KILTER Worksheet #1							
	CSUDP/SEWER			Input to KILTER	KILTER OUTPUT		
Master Connect.	"Sewer" Connect.	Arc Cost ($)	Arc Flow (cfs)	Arc Cost per unit flow $/cfs	Arc Cost $	Arc Flow cfs	New Connect.
B-A	2-1	4671	40.35				
C-B	12-2	1433	2.00				
D-B	3-2	6569	32.49				
E-C	10-12	-	-				
E-D	10-3	1506	2.29				
G-D	4-3	5823	26.07				
F-E	11-10	1431	1.50				
H-E	8-10	-	-				
I-F	9-11	-	-				
H-G	8-4	3780	10.19				
J-G	5-4	3538	11.01				
I-H	9-8	2411	4.70				
K-H	6-8	-	-				
L-I	7-9	-	-				
K-J	6-5	2396	4.84				
L-K	7-6	1299	2.00				
TOTAL		$34857					

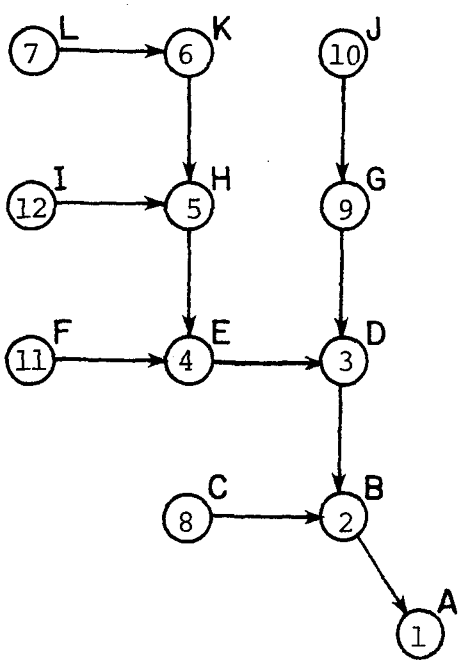


"OPTIMUM" LAYOUT

CSUDP/SEWER - KILTER Worksheet #2							
	CSUDP/SEWER			Input to KILTER	KILTER OUTPUT		
Master Connect.	"Sewer" Connect.	Arc Cost ($)	Arc Flow (cfs)	Arc Cost per unit flow $/cfs	Arc Cost $	Arc Flow cfs	New Connect.
B-A	2-1	4604	40.35	115	4712	41	2-1
C-B	3-2	6073	19.32	515	2062	4	8-2
D-B	8-2	4646	15.17	254	7881	31	3-2
E-C	4-3	5813	17.32	168	0	0	4-8
E-D	4-8	0	0	329	6576	20	4-3
G-D	9-8	4156	11.04	300	1799	6	9-3
F-E	12-4	1053	1.50	828	1656	2	11-4
H-E	5-4	4163	15.03	139	1803	13	5-4
I-F	6-12	0	0	0	0	0	12-11
H-G	5-9	0	0	185	0	0	5-9
J-G	10-9	2304	6.17	347	347	1	10-9
I-H	6-5	3266	6.70	500	1501	3	12-5
K-H	11-5	1769	2.84	311	1246	4	6-5
L-I	7-6	1356	2.00	339	0	0	7-12
K-J	11-10	0	0	247	0	0	6-10
L-K	7-11	0	0	325	649	2	7-6
TOTAL		39203					

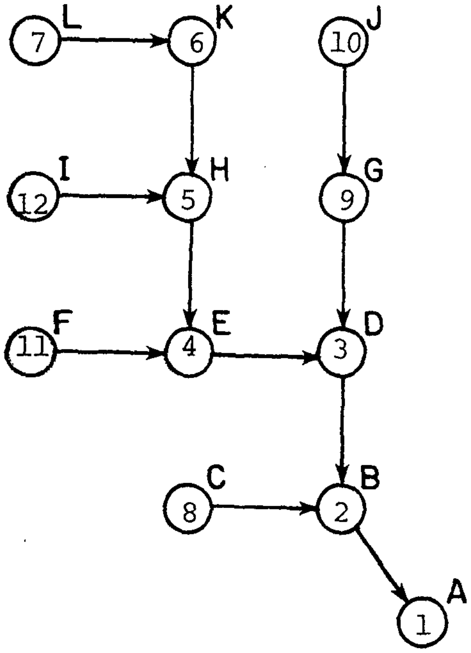


LAYOUT 1

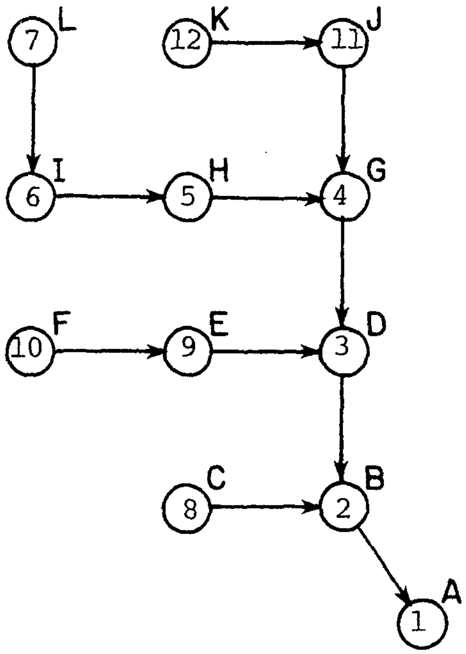


LAYOUT 2

CSUDP/SEWER - KILTER Worksheet #3							
	CSUDP/SEWER			Input to KILTER	KILTER OUTPUT		
Master Connect.	"Sewer" Connect.	Arc Cost ($)	Arc Flow (cfs)	Arc Cost per unit flow $/cfs	Arc Cost $	Arc Flow cfs	New Connect.
B-A	2-1	5415	40.35	124.57	5107	41	2-1
C-B	8-2	1110	2.00	535.28	2676	5	8-2
D-B	3-2	6800	32.49	231.77	6953	30	3-2
E-C	4-8	0	0	23.90	0	0	9-8
E-D	4-3	5626	17.32	326.83	2288	7	9-3
G-D	9-3	4183	11.04	337.41	6449	19	4-3
F-E	11-4	1048	1.50	763.60	1527	2	10-9
H-E	5-4	4118	15.03	206.69	0	0	5-9
I-F	12-11	0	0	0	0	0	6-10
H-G	5-9	0	0	92.74	927	10	5-4
J-G	10-9	2304	6.17	360.38	1442	4	11-4
I-H	12-5	2358	4.70	501.03	2004	4	6-5
K-H	6-5	2410	4.84	404.69	0	0	12-5
L-I	7-12	0	0	169.50	339	2	7-6
K-J	6-10	0	0	123.76	371	3	12-11
L-K	7-6	1443	2.00	523.24	0	0	7-12
TOTAL		36815					



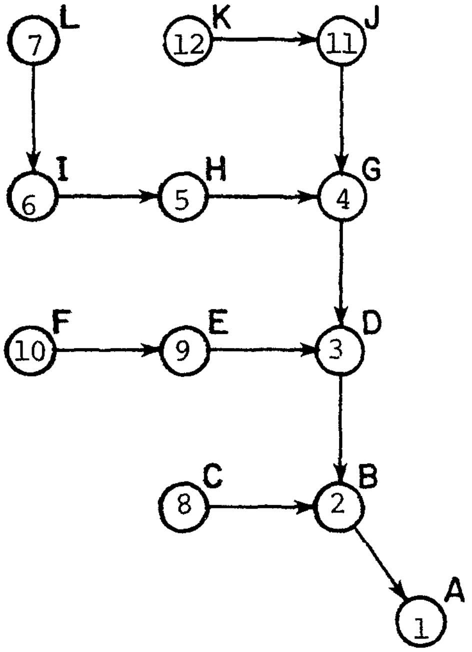
LAYOUT 2



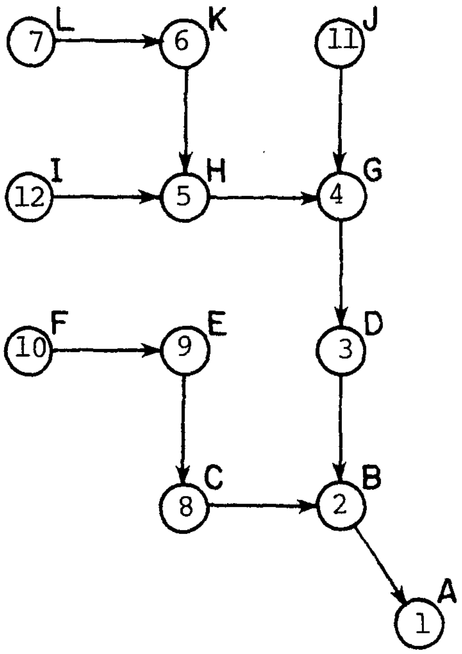
LAYOUT 3

CSUDP/SEWER - KILTER Worksheet # 4

	CSUDP/SEWER			Input to KILTER	KILTER OUTPUT		
Master Connect.	"Sewer" Connect.	Arc Cost ($)	Arc Flow (cfs)	Arc Cost per unit flow $/cfs	Arc Cost $	Arc Flow cfs	New Connect.
B-A	2-1	4671	40.35	120	4927	41	2-1
C-B	8-2	1433	2.00	626	6884	11	8-2
D-B	3-2	6569	32.49	217	5208	24	3-2
E-C	9-8	0	0	42	252	6	9-8
E-D	9-3	1506	2.29	492	0	0	9-3
G-D	4-3	5819	26.07	281	5626	20	4-3
F-E	10-9	1431	1.50	859	859	1	10-9
H-E	5-9	0	0	103	0	0	5-9
I-F	6-10	0	0	0	0	0	12-10
H-G	5-4	3811	12.19	203	2635	13	5-4
J-G	11-4	3858	9.01	394	789	2	11-4
I-H	6-5	3234	6.70	492	1476	3	12-5
K-H	12-5	0	0	202	809	4	6-5
L-I	7-6	1356	2.00	424	0	0	7-12
K-J	12-11	1794	2.84	378	0	0	6-11
L-K	7-12	0	0	262	523	2	7-6
TOTAL		35482					



LAYOUT 3



LAYOUT 4

Worksheets. The arc cost includes the upstream manhole and downstream pipe and excavation costs, except in the arc which includes manhole 1. In this arc cost, the downstream manhole cost (manhole 1) is included also (see Figure IV-1).

3. Find a new updated arc cost per unit flow for every possible arc in the system using equation 26. The arc cost per unit flow is calculated by dividing the total arc cost, calculated from CSUDP/SEWER output, by the arc flow.

4. Run KILTER to obtain a new horizontal layout.

5. Number the manholes in the new layout so that they are suitable for input to CSUDP/SEWER. This information is now transferred to the start of the next worksheet. Now run CSUDP/SEWER under the new layout. Find new unit costs for arcs in this layout and a new total system cost. If the new total cost is less than the old one, then proceed to the next step. Otherwise, reduce α and return to Step 3.

6. Update the arc cost per unit flow to be input to KILTER using equation 26. If an arc is assigned a high cost initially it may no longer be considered a possible flow path by KILTER. By updating the unit costs each time, eventually a reasonable cost should eventually be approached for each arc. The cost of the same arc may change for different layouts, so an exact cost is not expected.

7. With the updated arc costs, run KILTER again to obtain a new horizontal layout, and repeat as before.

8. STOP when $\alpha \rightarrow 0$, or some desired user tolerance.

Notice from these worksheets that the total system cost is steadily decreasing with each iteration and new layout, using step size $\alpha = 0.5$. For the final layout (Layout #4) in Worksheet #4, the total cost slightly increased. Therefore, with a fixed α, we would select Layout

#3 as optimum. Notice that is quite close to the original layout in configuration and total cost.

Further refinement is possible by returning to Layout #4, reducing α, and repeating. This was not done for this demonstration, however. Further work is required in experimenting with various search procedures governing unit cost changes and proper selection of step size α. Further work is also needed in automating the interaction between CSUDP/SEWER and KILTER. In particular, an automated numbering system is needed as each new layout is generated, so that the user does not have to do it each time. The deviation from the Frank-Morte algorithm that was necessary for this algorithm should also be functionized in future work, including the assumptons required in setting up Problem B; particularly equations 20-22. There may be ways that these assumptions can be relaxed, thereby providing a stronger theoretical basis for the methodology.

CHAPTER V

CONCLUSIONS AND RECOMMENDATIONS

A. OVERVIEW

An attempt has been made in this report to show a strong correspondence between cost-effectiveness or productivity in the urban water services and the introduction of computer-aided decision support systems into planning, design, and operational functions. Though attractive hardware abounds for these purposes as a result of the so-called microcomputer revolution, there is a severe lag in computer software availability that can aid urban water managers in finding cost-effective solutions to complex design and operational problems. This might help explain why urban water managers have been slow to implement this technology, even though expectations seemed high in the early 1970's.

Stormwater and combined sewer control is perhaps the most complex area that urban water managers deal with. Computerized decision support capability is vital if these problems are going to be properly solved. The Stormwater Control Package (SWCP) presented herein has been designed to be an effective tool for introducing automation into urban stormwater regulation. It combines state-of-the-art methods in storm forecasting, unsteady hydraulic sewer routing, and dynamic optimization into an integrated framework. We envision three uses of the package:

1. Operational planning for existing systems for introducing automation to regulate existing or planned control facilities in real-time. This mode involves simulated real-time experiments to ascertain the interaction between forecasting and control, and pinpoint

appropriate levels of sophistication in control.

2. Adapt the SWCP for actual real-time control through on-line processing and interaction with field monitoring and control equipment.

3. Introduce the SWCP into planning phases in order to determine if it is possible to reduce capital investment in new drainage, storage, and treatment facilities through the introduction of automation.

Our research has been limited to the first problem area, and has afforded a measure of confidence in the Package through use of the San Francisco system as a case study. Future emphasis must be placed on extending this work to other cities and expanding into the next two problem catagories.

Previous work by Labadie et al. (1977) has stressed that computer control of storm and combined sewer systems should be utilized in an integrated, city-wide basis. Decomposition algorithms using distributed computer processing have been developed in this earlier work for accomplishing this. Even though this previous research was applied to the concept of distributed detention storage over a city, versus the large shoreline interceptors as currently being constructed for San Francisco, it is believed that these algorithms are still valid in a general sense. Future research should concentrate on integrating the SWCP into a decomposition framework for city-wide control.

In addition to the SWCP as an operational tool, we have developed a model called CSUDP/SEWER for cost-effective design of new sewer and storm drainage system, or optimal expansion of existing systems. The ultimate goal is to combine the two packages together whereby the screening model can provide initial layouts and sizings that can be further refined and improved by application of the SWCP.

CSUDP/SEWER is applicable to vertical sizing of systems and selection of pipe sizes manhole depths, and slopes. Extensive experience has been gained with the model on a variety of sewer networks. Results compare well with publicized results of other methods. Horizontal layout which establishes how manholes should be interconnected, must be performed by trial and error if the CSUDP/SEWER package is used alone. In order to overcome this disadvantage, an algorithm is presented for linking a network flow algorithm called KILTER with CSUDP/SEWER as a means of solving both the vertical and horizontal layout problems. This latter work is more preliminary in nature, and more research is needed to fully test and refine the procedure.

An issue that has not been addressed in this research is the stochastic nature of the inflows and consideration of risk in real-time control decisions. The real-time stormwater control problem involves estimating the current and future state (i.e., flows and heads throughout the system) as governed by stochastic, uncontrollable storm flow inputs and controllable gates, valves, regulators, orifices, pumps, etc. A number of elegant results are available for linear systems with Gaussian error under quadratic criteria (Bertsekas, 1976). Even when it is possible to linearize the system state model, we are confronted with criteria that are discontinuous and nonconvex. For our problem, pumping costs which vary nonlinearly with flow rate and pumping lift create such a nonconvexity.

This nonconvexity inhibits applying the methods of modern optimal control theory, particularly with the inclusion of tightly confining state-space and control constraints (Tabuk and Kuo, 1971). It seems clear that dynamic programming is an attractive approach under these circumstances, if a satisfactory way of dealing with the curse of dimensionality (Bellman, 1957) can be found.

In addition to these difficulties, there are further annoyances in the real-time management of water resource systems due to noise-corrupted measurements of the system state. Storm and combined sewers represent a hostile environment for sensors and gages due to suspended materials, and large measurement errors can occur.

The Kalman filter has been around for two decades, but has only recently found its way into research related to water resources management (Kalman, 1960). Chen (1974) combined optimal control theory with the Kalman filter for solving a simplified stormwater control problem, but the method suffers from the previously mentioned disadvantages of optimal control theory and the maximum principal. In effect, the Kalman filter is a recipe for combining two independent estimates of the state of a system into one minimum variance estimate. It also produces valuable second-order statistical information that can be used for risk analysis. One estimate of the system state comes from the state prediction model itself, which of course is only an approximation; the other comes from noise-corrupted direct measurements of the system state at discrete points in time and space.

It is possible to explicitly combine dynamic programming and the Kalman filter together in a real-time decision framework. This results, however, in a high dimensional information vector composed of first and second-order statistical information on the random state vector. The second order information must be included in order to analyze the conditional risk of failure associated with control decisions. For most practical problems, it is impossible to evaluate the dynamic programming optimal value (or optimal return, or 'cost-to-go') function for all possible discrete combinations of the information vector. It is shown here that for a certain class of problems, it may not be necessary to do this.

B. OPTIMAL STORMWATER CONTROL UNDER RISK

B.1 Objective Function

The objective is to minimize the expected value of discounted costs over stages i=1,...,N, where state $\underline{x}_{i+1}$ is a random vector and decision $\underline{u}_i$ is completely controllable. The costs may be real costs, such as associated with energy for pumping, or pseudo costs representing locational priorities for untreated overflows when overflows cannot be avoided.

$$\min E \sum_{i=1}^{N} [f_i(\underline{u}_i, \underline{x}_i, x_{i+1})] \tag{1}$$

Again, state vector $\underline{x}_i$ represents flow rates and heads at discrete locations throughout the system, generally corresponding to the sewer routing model used. The $\underline{u}_i$ are various control decisions related to gate openings, regulator settings, pumping rates, etc.

B.2 Constraints

State equation: It is assumed that the state of the system can be reasonably governed by a linear or linearized set of discrete difference equations. For real-time applications, these equations can be updated and relinearized around current nominal trajectories as new information dictates.

$$\underline{x}_{i+1} = A_i\underline{x}_i + B_i\underline{u}_i + \underline{c}_i + \underline{r}_i, \quad i=1,\ldots,N \tag{2}$$

where $\underline{x}_i \varepsilon R^n, \underline{u}_i \varepsilon R^m, \underline{c}_i \varepsilon R^n, \underline{r}_i \varepsilon R^n$, with $\underline{r}_i$ as an independently distributed Gaussian error term or model noise:

$$\text{(i)} E\{\underline{r}_i\} = \underline{0}, \quad i=1,\ldots,N$$

$$(ii)E\{\underline{r}_i\underline{r}_j\} = Q\delta_{ij} \quad i,j=1,\ldots,N$$

where δ_{ij} is the Kronecker delta and Q is a known covariance matrix. Matrices A_i and B_i are also known or can be updated at each time step, and are (nxn) and (nxm), respectively. Constant $\underline{c}_i$ would represent boundary conditions such as forecasted inflow hydrographs, state-discharge relations, etc. The errors $\underline{r}_i$ would be primarily associated with the inflow forecasts, though not entirely. Equation 2 could represent an explicit form, fully dynamic routing model, with boundary conditions included in (2). An implicit model could be represented as

$$A_i(\underline{x}_i)\underline{x}_i + B_i(\underline{x}_i)\underline{u}_i + \underline{c}_i + \underline{r}_i - D_i(\underline{x}_i)\underline{x}_{i+1} = \underline{0}$$

or

$$\underline{x}_{i+1} = D^{-1}A\underline{x}_i + D^{-1}B\underline{u}_i + D^{-1}\underline{c}_i + D^{-1}\underline{r}_i$$

The error terms are now correlated, but Gelb (1974) shows that this formulation can be converted into an equivalent one with an uncorrelated structure. Note that implicit numerical formulations are rarely solved by explicitly taking the inverse of matrix D, but this is a convenient way of representing the solution.

<u>Initial state</u>: The initial state of the system is assumed to be Gaussion with known mean $\tilde{\underline{x}}_1$ and error covariance matrix P_1.

$$(i)E\{\underline{x}_1\} = \tilde{\underline{x}}_1$$

$$(ii)E\{\underline{x}_1 - \tilde{\underline{x}}_1)(\underline{x}_1 - \tilde{\underline{x}})\}' = P_1$$

and is independent of $\underline{r}_i$ for all i.

<u>Observation model</u>: In addition to the state prediction model (2), direct, though noisy, observations $\underline{z}_{i+1}\varepsilon R^p$ can be taken of some or all of the system states $\underline{x}_{i+1}$ at the end of stage i, after decision $\underline{u}_i$ is made:

$$\underline{z}_{i+1} = C\underline{x}_{i+1} + \underline{v}_{i+1}, \quad i=1,\ldots,N \tag{3}$$

where matrix C is (pxn). If, for example, all the states were directly monitored, then C would be the identify matrix. It is assumed that the error term is also independently distributed Gaussian with known covariance matrix R:

$$\text{(i)}E\{\underline{v}_i\} = \underline{0}$$

$$\text{(ii)}E\{\underline{v}_i\underline{v}_j'\} = R\delta_{ij}$$

and is independent of $\underline{x}_1$ and $\underline{r}_i$ for all i.

Control and state-space constraints: The decision vector $\underline{u}_i$ is assumed contained in a closed and bounded set:

$$\underline{u}_i \varepsilon U_i \tag{4}$$

We would like to confine $\underline{x}_i$ to a closed and bounded set X, but cannot guarantee this since $\underline{x}_i$ is a random vector. This constraint must therefore be stated probabilistically:

$$\text{Prob}\{\underline{x}_{i+1} \varepsilon X_{i+1} | \underline{u}_j, z_j, \quad j=1,\ldots,i\} \geq (1 - \alpha_i), \quad i=1,\ldots,N \tag{5}$$

where α_i is a desired risk level. Notice that this constraint must be stated conditionally. It is much more difficult to evaluate the unconditional probability that $\underline{x}_{i+1} \varepsilon X_{i+1}$. It is assumed here that the problem as formulated is both controllable and observable.

C. KALMAN FILTER

C.1 Observation Model

Assuming that decision $\underline{u}_i$ is made at the beginning of stage i, prior to observations taken during stage i, it can be proved by maximum likelihood or least-squares methods (see Åstrom (1970)) that the minimum variance estimate of the state of the system after observations are taken for stage i is:

$$\tilde{\underline{x}}_{i+1|i+1} = \tilde{\underline{x}}_{i+1} + G_i[\underline{z}_{i+1} - C\tilde{\underline{x}}_{i+1}] \qquad (6)$$

where the state estimate from the model is:

$$\tilde{\underline{x}}_{i+1} = A_i\tilde{\underline{x}}_{i|i} + B_i\underline{u}_i + \underline{c}_i \text{ with } \underline{x}_{1|1} \triangleq \tilde{\underline{x}}_1 \qquad (7)$$

and the error covariance matrix from the model is:

$$P_{i+1} = A_i P_{i|i} A_i' + Q$$
$$(i=1,\ldots,N)$$

The matrix G_i is called the Kalman gain:

$$G_i = P_{i+1}C' \; [R + CP_{i+1}C']^{-1} \qquad (9)$$

The best (i.e., minimum variance) estimate of the state error covariance matrix after observations are taken is:

$$P_{i+1|i+1} = P_{i+1} - G_i C P_{i+1} \qquad (10)$$

the subscript (i+1|i+1) means that the estimate is conditioned on observations taken during stage i. The subscript (i+1) signifies that these estimates are obtained from the system state model prior to taking observations.

Notice that Equation 6 is suggestive of a gradient-type algorithm. Estimates are adjusted according to whether observations are above or below the estimates generated by the system model. The gain matrix tends to decrease as the elements of R increase. This means that we have less confidence in the observations so they have less of an effect on the estimates generated by (6). It can be seen from Equations 8 and 9 that this same effect is produced if the elements of Q are small in comparison with those of R. If, on the other hand, Q is large in relation to R, then the gain factor will also tend to be larger since this indicates greater trust in the measurements than the model.

The great advantage of the Kalman filter is its recursive structure; which suggests an obvious linkage with dynamic programming. Young (1974) has shown that the Kalman filter is related to recursive least-squares estimation of the parameters of linear models. It is not necessary to store the entire past history of the process, but only keep track of 1st and 2nd order statistical information from the previous stage. Since this information is sufficient to describe a random Gaussian process, we are in effect generating a sequence of conditional probability distributions.

C.2 Forecast Model

Notice from Equation 8 that without benefit of the observations, the error covariances tend to increase, reflecting the increasing uncertainty of projecting further into the future. In the forecast mode, without benefit of observations, the estimates of the future mean and covariances are recursively generated by:

$$\underline{x}_{i+1} = A_i\underline{x}_i + B_i\underline{u}_i \tag{11}$$

$$P_{i+1} = [A_iP_iA_i' + Q] - G_iC[A_iP_iA_i' + Q] \tag{12}$$

where G_i is defined by Equation 9.

We now have the means of generating extended forecasts of the system state and covariance using Equations 11 and 12. When actual observations become available, the states can be updated using Equation 6.

C.3 Estimating Noise Covariances

The noise covariance matrices Q and R must be estimated, along with the initial state error covariance matrix P_1. This is largely a subjective matter. As pointed out by Mehra (1978), one advantage of the Kalman filter is that:

> '...the forecaster can use his judgement regarding the relative accuracy of the model values vs. observations to select appropriate values for noise covariance matrices Q and R. He can then examine the actual operation of the filter and adjust values on-line if the situation changes at a later time.'

The matrix P_1 can often be initialized as a diagonal matrix with large elements (e.g., 10^4 to 10^6). As observations are obtained in real-time, these covariances should reduce, reflecting the decreasing uncertainty as to the system state as additional information is obtained. If these covariances are too small, the model will tend to weight model estimates too heavily over subsequent data that may indicate different different estimates. On the other hand, if the covariances are set too high, convergence may be extremely slow. The great flexibility in selecting noise and error covariances is therefore both an advantage and disadvantage. Often a large amount of experimentation is needed.

It is assumed here that a forecasting model would be developed separately for predicting storm inflows to the sewer system, which would be included in term $\underline{c}_i$ in equation (2). This would imply that all rainfall data would have to be preprocessed by a rainfall-direct runoff watershed model, and then this processed data used to identify a time series model such as AR or ARMA. In work with the City of San

Francisco, watershed modeling has shown to have a high degree of accuracy, based on data collection from individual storm events (Kibler and Roesner, 1975). Labadie et al. (1981) have used an autoaggressive transfer function model for forecasting rainfall directly, but it is believed that direct forecasting of watershed runoff might prove to be more accurate due to the damping effect a watershed system normally has on rainfall input. This would not be true for urban areas with a high percentage of perviousness.

D. DYNAMIC PROGRAMMING FORMULATION

With the assumption of independent Gaussian noise, the sufficient statistics for the random state vector $\underline{x}_i$ are $(\tilde{\underline{x}}_i, P_i)$. We will continue to refer to $(\tilde{\underline{x}}_i, P_i)$, as the system state, in a probabilistic sense. The following backward dynamic programming problem can now be formulated for all discrete combinations of $(\tilde{\underline{x}}, P_i)$, evaluate:

$$F_i(\tilde{\underline{x}}, P_i) = \text{minimum expected discounted cost totaled over stages i through N, given that state } \tilde{\underline{x}}_i \text{ is Gaussian with mean } \tilde{\underline{x}}_i \text{ error covariance matrix } P_i.$$

$$= \underset{u_i \varepsilon U_i \underline{r}_i, \underline{v}_{i+1}, \underline{x}_i}{\min} E[f_i(\underline{u}_i, \underline{x}_i, \underline{x}_{i+1}) + F_{i+1}(\tilde{\underline{x}}_{i+1|i+1}, P_{i+1|i+1})] \tag{13}$$

where

$$\underline{x}_{i+1} = A_i \underline{x}_i + B_i \underline{u}_i + \underline{c}_i + \underline{r}_i ; \; \underline{x}_i \sim N(\tilde{\underline{x}}_i, P_i) \tag{14}$$

$$\tilde{\underline{x}}_{i+1|i+1} = [A_i \tilde{\underline{x}}_i + B_i \underline{u}_i] + G_i [\underline{v}_{i+1} + C(\underline{u}_i)[\underline{x}_{i+1} - \tilde{\underline{x}}_{i+1}]] \tag{15}$$

$$G_i = P_{i+1} C' [R + C P_{i+1} C']^{-1} \tag{16}$$

$$P_{i+1} = A_i P_i A_i' + Q \tag{17}$$

$$P_{i+1|i+1} = P_{i+1} - G_i C P_{i+1} \tag{18}$$

$$\text{Prob}\{\underline{x}_{i+1} \varepsilon X_{i+1} | \tilde{\underline{x}}_i, P_i, \underline{u}_i\} \geq (1-\alpha_i) \tag{19}$$

The dynamic programming recursion is initialized by defining:

$$F_{N+1}(\cdot) \triangleq 0 \tag{19}$$

The terms on the right hand side of Equation 13 are conditioned on $\tilde{\underline{x}}_i$ and P_i. There are several difficulties with this formulation. First, evaluating the expected values on the right-hand side of Equation 13 with respect to vectors $\tilde{\underline{x}}_i, \underline{v}_i$ and $\underline{v}_{i+1}$ is a virtually impossible task for general nonlinear functions f_i it would have to be linear or quadratic to do this. Equation 19 would be difficult to evaluate in this formulation since $\underline{x}_{i+1}$ is multivariate Gaussian with statistics conditioned on $\tilde{\underline{x}}_i, p_i$, and $\underline{u}_i$. It is a foregone conclusion that unless the state-space is severely limited, or $f_i(.)$ is approximated as a quadratic, this problem is impossible to solve directly. This is true not only because of the enormous number of values of $F(.)$ that would have to be computed and stored in core memory as a function of all discrete combinations of $(\tilde{\underline{x}}, P_i)$ (which, assuming P_i is a symmetric matrix, would have a dimension of $(n^2 + 3n)/2$), but also because of the prodigious amount of computer time involved in evaluating the expected value of the terms in brackets. Clearly, alternative approaches must be sought.

E. ALTERNATIVE FORMULATION

E.1 Deterministic Case

An alternative formulation to this problem will now be developed which has considerable computational advantage over the previous one.

With this formulation, it is possible to define a set of sufficient conditions for optimality of the decisions obtained from this procedure, but not, in general, both necessary and sufficient conditions. In order to explain the essential basis of the method, we will first use a deterministic formulation; i.e., assume all covariances R, Q, and P equal zero.

We will assume that the objective function $f_i(.)$ is a positive mapping. Define L_i as a lower bound on the total costs accumulated over stages 1,2,...,i, under a particular decision policy $\underline{u}_j^*$, j=1,...,i. Therefore

$$L_i \leq \sum_{j=1}^{i} f_j(\underline{x}_i,\underline{u}_i,\underline{x}_{j+1})$$

By this definition

$$L_i \leq L_{i-1} + f_i \leq F_{i-1} + f_i$$

where F_{i-1} are accumulated costs through stage i-1. Assume that for various discrete lower bounds L_{i-1}, we have stored unique optimal states $x_i^*(L_{i-1})$ from previous stage computations. That is, we are assuming that for a specified lower bound L_{i-1} and given initial state $\underline{x}_i$, there is a unique optimal policy $\underline{u}_1^*,\underline{u}_2^*,\ldots,u_{i-1}^*$ and resulting state $\underline{x}_i^*$ that minimizes total costs over stages 1,...,(i-1), subject to a specified lower bound on costs L_{i-1}. Assume that we have stored for a discrete range of lower bounds L_{i-1} the actual accumulated costs $F_{i-1}(L_{i-1})$.

Therefore, for stage i, and for several discrete lower bounds L_i:

$$\underset{L_{i-1},\underline{u}_i\varepsilon U_i}{\text{minimize}} \quad F_{i-1}(L_{i-1}) + f_i(\underline{x}_i^*(L_{i-1}),\underline{u}_i,\underline{x}_{i+1})$$

subject to:

$$L_i \leq F_{i-1}(L_{i-1}) + f_i(\underline{x}_i^*(L_{i-1}), \underline{u}_i, \underline{x}_{i+1})$$

$$\underline{x}_{i+1} = A_i\underline{x}_i^*(L_{i-1}) + B_i\underline{u}_i + \underline{c}_i$$

This minimization could be performed by any number of static optimization methods. In fact, for one problem, dynamic programming could be applied sequentially over spatial stages (i.e., sewer sections), as done previously by Labadie et al. (1980). For each discrete lower bound, L_i, we store the optimal $L_{i-1}^*(L_i)$ and $\underline{u}_i^*(L_i)$, as well as the corresponding optimal $\underline{x}_{i+1}^*(L_i)$. Again, we assume uniqueness. Without this assumption, there would be several other policies just as good as the selected one which would have to be discarded due to the combinational problems in carrying them along. Even though up to stage i, any of these policies would be equally optimum, a nonunique policy that we arbitrarily decided to discard at this stage might turn out to be better in the long run. If, however, there is only one unique policy corresponding to the lower bound L_i, then a decision to discard policies is not required. We then proceed to stage i+1, and so on.

Upon reaching stage N, solve for a range of discrete L_N:

$$\underset{L_{N-1}, \underline{u}_N \varepsilon U_N}{\text{minimize}} \quad F_{N-1}(L_{N-1}) + f_N(\underline{x}_N^*(L_{N-1}), \underline{u}_N, \underline{x}_{N+1}$$

subject to

$$L_N \leq F_{N-1}(L_{N-1}) + F_n(\underline{x}_N^*(L_{N-1}), \underline{u}_N, \underline{x}_{N+1})$$

$$\underline{x}_{N+1} = A_N \underline{x}^*_N(L_{N-1}) + B_N \underline{u}_N + \underline{c}_N \ \varepsilon X_{N+1}$$

The minimum total cost is bounded from below by the least lower bound. That is, we obtain $F_N(L_N)$ from these computations and then simple solve the following:

$$\underset{L_N}{\text{minimize}} \ F_N(L_N)$$

We can now traceback through the stored optimal $L^*_{i-1}(L_i)$ and $\underline{u}^*_i(L_i)$ to determine the optimal decision policy.

Notice that this is an optimization in objective-space rather than the usual state-space dynamic programming approach. Optimization is performed at each stage over a discrete range of a scalar valued quantity L_i, rather than all discrete combinations of the state vector $\underline{x}_i$. Since we are assuming uniqueness of the policies developed at each stge, we can store the optimal policies $\underline{u}^*_i(L_i)$ as a function of the lower bound L_i rather than storing $\underline{u}^*_i(\underline{x}_i)$ in the usual approach. Since we also have $\underline{x}^*_i(L_i)$, we can obtain a corresponding operating rule $\underline{u}^*_i(\underline{x}_i)$, but it will not necessarily be as complete as with the standard DP approach. The advantage of this procedure is that the modest storage and computational requirements would allow the algorithm to be rerun in real-time for any given initial state in any given stage, rather than having policies for all possible states stored apriori.

E.2 Stochastic Case

With this basic approach, it would be possible to carry along the covariances $P_i(L_{i-1})$ from stage to stage just as the optimal state estimates $\tilde{\underline{x}}_i(L_{i-1})$ would be carried along, again assuming that uniqueness applies. The problems of evaluating the expected value in equation 13 and evaluating the probabilistic constraint (equation 19) still apply.

Certain simplifications could be possibly made without unduly jeopardizing the integrity of the formulation:

1. Replace $\underline{v}_{i+1}$ in Equation 13 with its expectation; namely zero. This might be justifiable if measurement error is considered to be an order of magnitude less than model error, as reflected in selection of model covariance matrices Q and R. This is probably valid for our problem.
2. Replace x_i on the right-hand side of Equation 13 with its expectation, $\tilde{x}_i$.

E.3 Risk Analysis

It is recommended that the risk constraint (Equation 19) be indirectly considered through inclusion of a penalty term. It could be based on current covariance estimates and would attempt penalize policies which based on current covariance information, have a higher risk of violating the constraints X_{i+1}, such as related to localized flooding and untreated overflows. The penalty term would have to be adjusted and then simulations performed on the system to determine an actual risk level.

REFERENCES

American Public Works Association, "Feasibility of Computer Control of Wastewater Treatment Plants," 1970.

American Society of Civil Engineers, "Design Construction of Sanitary and Storm Sewers," Manuals and Reports on Engineering Practice No. 37, 1970.

Astrom, K. J., Introduction to Stochastic Control Theory, Academic Press, New York, 1970.

Barlow, J. F., "Cost Optimization of Pipe Sewerage Systems," Proceedings, Institution of Civil Engineers (London), Vol. 53, Part 2, pp. 57-64, June 1972.

Bellman, R. E., Dynamic Programming, Princeton University Press, Princeton, New Jersey, 1957.

Bertsekas, D. P., Dynamic Programming and Stochastic Control, Academic Press, 1976.

Book, D. E., J. W. Labadie and D. M. Morrow, "Dynamic vs. Kinematic Routing in Modeling Urban Storm Drainage," Proceedings of the Second International Conference on Urban Storm Drainage, B. C. Yen, editor, University of Illinois at Urbana-Champaign, Urbana, Illinois, pp. 154-163, June 14-19, 1981.

Box, G.E.P. and G. M. Jenkins, "Time Series Analysis," Forecasting and Control, Holden-Day, San Francisco, 1976.

Brown and Caldwell, Inc., "Municipality of Metropolitan Seattle's Computerized Stormwater Control System," Project Highlights, No. 5, 1978.

Caldwell, Gonzales, Kennedy, Tudor Consulting Engineers, "Bayside Facilities Plan, Citywide Control System Report," San Francisco Clean Water Program, City and County of San Francisco, California, February 1981.

Canon, M. D. and C. D. Cullum, "A Tight Upper Bound on the Rate of Convergence of the Frank-Wolfe Algorithm," IBM Watson Research Center, Yorktown Heights, New York, 1967.

Chen, Y. H., "Water Routing in Rivers," Chapter 8, Analysis of River Systems and Watersheds Shortcourse," Colorado State University, Fort Collins, Colorado, June 1979.

Dajani, J. S. and R. S. Gemmell, "Economics of Wastewater Collection Networks," Research Report No. 43, Water Resources Center, University of Illinois at Urbana-Champaign, Illinois, 1971.

Dedinsky, H., J. Grinker, R. Meagher and J. Schlintz, "What is Needed to Successfully Automate a Wastewater Treatment Plant," Central States Water Pollution Control Association, May 1982.

Fontane, D. G., J. W. Labadie, and B. Loftis, "Optimal Control of Reservoir Discharge Quality Through Selective Withdrawal," Water Resources Research, Vol. 17, No. 6, pp. 1594-1604, December 1981.

Ford, L. R. and D. R. Fulkerson, Flows in Networks, Princeton University Press, Princeton, New Jersey, 1962.

Froise, S., "Least Cost Control Strategies in Urban Drainage Design--A Dynamic Programming Approach," Report No. 46, Charles W. Harris Hyd. Lab., University of Washington, November 1975.

Gelb, A., Applied Optimal Estimation, the MIT Press, Cambridge, Mass., 197 .

Graupe, D., "Identification of Systems," Van Nostrand Reinhold, New York, 1976.

Grigg, N. S., "Productivity Improvements in Urban Water Systems," ASCE Water Planning and Management Specialty Conference, Lincoln, Nebraska, May 1982.

Hall, W. A., et al., "Optimum Firm Power Output from a Two Reservoir System by Incremental Dynamic Programming," Cont. No. 130, University of California, Los Angeles, California, Water Res. Center.

Heidari, M., et al., "Discrete Differential Dynamic Programming Approach to Water Resources Systems Optimization," Water Res. Research, Vol. 7, No. 2, April 1971, pp. 273-282.

Holland, M. G., "Computer Models of Wastewater Collection Systems," Water Resources Group, Harvard University, Cambridge, Mass., 1966.

Kalman, R. E. and R. S. Bucy, "New Results in Linar Filtering and Prediction Problems," Journal of Basic Engineering, March 1960, pp. 35-46.

Kazanowski, A. D., "A Standardized Approach to Cost-Effectiveness Evaluation," Cost-Effectiveness, J. M. English, ed., John Wiley, 1968.

Kibler, D. F., J. R. Monser, and L. A. Roesner, "San Francisco Stormwater User's Manual and Program Documentation," Water Res. Engineers, Walnut Creek, California, 1975, pp. 40,47.

Labadie, J. W., "Real-Time Forecasting and Control Decisions for Urban Runoff," 1982 Spring Meeting, American Geophysical Union, Philadelphia, Pennsylvania, May 31-June 4, 1982.

Labadie, J. W., Grigg, N. S., and B. H. Bradford, "Automated Control of Large-Scale Sewer Systems," Journal of the Environmental Engineering Division, ASCE, Vol. 101, No. EE1, pp. 27-39, February 1975.

Labadie, J. W., Lazaro, R. C., and D. M. Morrow, "Worth of Short-Term Rainfall Forecasting for Combined Sewer Overflow Control," Water Resources Research, Vol. 17, No. 6, pp. 1594-1604, October 1981.

Labadie, J. W., Morrow, D. M. and Y. H. Chen, "Optimal Control of Unsteady Combined Sewer Flow," Journal of the Water Resources Planning and Management Division, ASCE, Vol. 106, No. WR1, pp. 205-223, March 1980.

Labadie, J. W., Morrow, D. M. and Lazaro, R. C., "Urban Stormwater Control Package for Automated Real-Time Systems," Report, prepared for U.S. Dept. of Interior, Office of Water Research and Technology, Project C 6174, Colorado State University, Fort Collins, Colorado, December 1978.

Larson, R. E., State Increment Dynamic Programming, American Elsevier, 1968.

Liebman, J. C., "A Heuristic Aid for the Design of Sewer Networks," Jour. San. Eng. Div., ASCE, Vol. 93, No. SA4, pp. 81-90, August 1967.

Liggett, J. A. and J. A. Cunge, "Numerical Methods of Solution of the Unsteady Flow Equations," Chapter 4, Vol. 1, Unsteady Flow in Open Channels, Water Resources Publications, Fort Collins, Colorado, 1975.

Lowsley, Jr., I. H., "An Implicit Enumeration Algorithm for Optimal Sewer Layout," Ph.D. Thesis, Johns Hopkins University, Baltimore, Maryland, 1973.

Mandl, C. E., "A Survey of Mathematical Optimization Models and Algorithms for Designing and Extending Irrigation and Wastewater Networks," Water Resources Research, Vol. 17, No. 4, pp. 769-775, August 1981.

Mays, L. W. and H. G. Wenzel, Jr., "A Serial DDDP Approach for Optimal Design of Multi-level Branching Storm Sewer Systems," Water Resources Research, Vol. 12, No. 5, October 1976.

McPherson, M. B., "Feasibility of the Metropolitan Water Intelligence System Concept," Technical Memorandum #15, ASCE Urban Water Resources Research Program, December 1971.

Medearis, K., Computer and Control Equipment, MWIS Project, Colorado State University, December 1971.

Mehra, R. K., "Practical Aspects of Designing Kalman Filters," in Applications of Kalman Filter to Hydrology, Hydraulics, and Water Resources, C.-L. Chiu, Ed., Stochastic Hydraulics Program, Department of Civil Engineering, University of Pittsburg, Pennsylvania, 1978.

Morrow, D. M. and J. W. Labadie, "Urban Stormwater Control Package for Automated Real-Time Systems," AGU Symposium on Urban Hydrometeorology, Toronto, Ontario, Canada, May 26-27, 1980.

Pirnie, M., "Wastewater Program Overview for the City and County of San Francisco," Malcolm Pirnie and Associates, January 1980.

Poertner, H., "Existing Automation, Control and Intelligence Systems of Metropolitan Water Facilities," MWIS Technical Report #1, Colorado State University, 1972.

Ponce, V. M., Li, R. M. and D. B. Simons, "Applicability of Kinematic and Diffusion Models," Journal of the Hydraulics Division, ASCE, 104 (HY3):353-360, March 1978.

Robinson, D. K. and J. W. Labadie, "Optimal Design of Urban Stormwater Drainage Systems," Proceedings of the 1981 International Symposium on Urban Hydrology, Hydraulics, and Sediment Control, D. Wood, editor, University of Kentucky, Lexington, Kentucky, pp. 145-156, July 27-30, 1981.

Tabuk, D. and B. Kuo, Optimal Control by Mathematical Programming, Prentice-Hall, 1971.

Trotta, P. D., "On-Line Adaptive Control for Combined Sewer Systems," Ph.D. Dissertation, Colorado State University, Spring 1976.

Trotta, P. D., J. W. Labadie, and N. S. Grigg, "Automatic Control Strategies for Urban Stormwater," Journal of the Hydraulics Division, ASCE, Vol. 103, No. HY12, pp. 1443-1459, December 1977.

U.S. Army Corps of Engineers, "Tainter Gates in Open Channels--Discharge Coefficients Hydraulic Design Criteria," U.S. Army Engineers Waterways Experiment Station, Vicksburg, Mississippi, 1971.

U.S. Environmental Protection Agency, "Stormwater Management," 1974 Report by Metcalf and Eddy.

Wenzel, H. G., "Application of a Least Cost Design Model for Stormwater System Design," Presented at the 1980 ASCE Annual Convention at Hollywood, Florida, October 1980.

Wenzel, H. G., Jr., B. C. Yen, and W. H. Tang, "Advanced Methodology for Storm Sewer Design--Phase II," University of Illinois Water Resources Center Research Report No. 140, June 1979.

Young, P., "Recursive Approaches to Time Series Analysis," Institute of Mathematics and Its Applications--Bulletin, Vol. 10, Nos. 4 & 5, 1974.

Zepp, P. L. and A. Leary, "A Computer Program for Sewer Design and Cost Estimation," Regional Planning Couom NTIS, Springfield, Virginia), 1969.

APPENDIX A

WORKS OF SHORT-TERM RAINFALL FORECASTING FOR COMBINED SEWER OVERFLOW CONTROL

WATER RESOURCES RESEARCH, VOL. 17, NO. 5, PAGES 1489–1497, OCTOBER 1981

Worth of Short-Term Rainfall Forecasting for Combined Sewer Overflow Control

JOHN W. LABADIE AND ROGELIO C. LAZARO

Department of Civil Engineering, Colorado State University, Fort Collins, Colorado 80523

DENNIS M. MORROW

Department of Civil Engineering, University of Missouri at Columbia, Columbia, Missouri 65211

Real-time, short-term rainfall forecasting is gaining recognition as a valuable input to more effective performance of a variety of urban water management activities, including controlling the incidence of untreated combined sewer overflows. An important question is, What levels of forecast error can be tolerated before it is better to abandon adaptive control policies utilizing forecast information in favor of simple reactive control methods? Experiments with an autoregressive-transfer function model for short-term forecasting are presented, utilizing the San Francisco North Shore Outfalls Consolidation Project as a case study. A split data technique is used to gain insight into expected forecast errors for selected overflow-producing storms varying from high intensity–low duration to low intensity–high duration. These results are then compared with the performance of the planned system, utilizing automatically controlled gates in a large shoreline tunnel, for various levels of forecast error. The results of a limited number of simulation runs indicate that expected forecast model errors are generally lower than the error threshold above which reactive policies become more attractive.

INTRODUCTION

Forecasting an occurring storm event is important for real-time operational control of combined sewer flows as a means of obtaining the best possible performance from existing or planned facilities. Forecasting models allow anticipation of sewer inflows and therefore provide additional lead time to effect control strategies such as starting wet weather pumps and treatment plants and manipulating controllable gates or weirs. It may be possible to reduce the magnitude and frequency of untreated wet weather overflows to receiving waters through use of forecast information. In situations where this is not possible, there may still be substantial benefits in reducing the adverse impacts of untreated overflows on receiving waters by altering the temporal and spatial distribution of total overflows (e.g., capturing the first flush or taking advantage of tidal fluctuations).

There are two basic approaches to real-time control: reactive (or myopic) control and adaptive (or anticipatory) control. The reactive approach involves use of a priori derived operating rules. The current period control decisions are selected by monitoring only current telemetered rainfall, sewer flow, and storage. For the adaptive approach it is argued that control decisions in the current real-time period should be based on future anticipated inflows as well as on current conditions so that available storage, treatment, and flow capacities can be utilized in the best way. The latter approach requires more sophisticated on-line computer capability, both in hardware and software. There are, of course, various gradations between these extremes, such as deriving several operating rules for various anticipated storm characteristics.

These real-time control concepts apply not only to urban sewer systems but also to a wide variety of complex water resource systems such as multipurpose, multireservoir systems. Among the first to recognize the discrepancy between these two control approaches were *Jamieson and Wilkinson* [1972, p. 915], who stated:

> If the forecast is based solely on telemetered values of rainfall, the implicit assumption is that there will be no subsequent rainfall from the time of forecasting; this assumption must be the worst one possible in the middle of a severe storm. Clearly, some other assumption is desirable, but in the absence of quantitative rainfall forecasts, it is not obvious what it should be.

In arguing the pros and cons of each approach, the basic question as to which is better is inextricably tied to forecast accuracy. If real-time forecasts are totally unreliable, with little hope for improvement even if more sophisticated technology is applied, then it is probably safer to use reactive policies. Or it may be possible to improve forecast accuracy dramatically through, for example, extensive radar facilities, but the costs involved might be prohibitive. On the other hand, it might be demonstrated that forecasts are reliable enough such that appreciable improvement of adaptive policies over reactive policies is possible. The question is, what is enough? More specifically:

1. What level of forecast error can be tolerated in adaptive control policies before it becomes better to simply use reactive methods?
2. Assuming that it is better to use forecasts, what lead time is most appropriate? (I.e., how far in the future should we attempt to anticipate?)

An initial attempt to answer these questions is presented herein. The San Francisco Master Plan for Wastewater Management is used as a case study, so the results are specific to that area. However, they may provide an indication of the value or worth of real-time forecasting and automatic control in an adaptive mode.

The following sections describe the forecasting model and experiments to provide an indication of what kinds of forecast errors can be expected for the San Francisco area. An attempt is then made to tie these results to some concurrent work on measuring the performance of a simulated real-time automatic control system as a function of degree of forecast error. For the latter the forecasting model was not actually used. Rather, as a controlled experiment, inflow hydrographs with a

Paper number 1W1020.
0043-1397/81/001W-1020$01.00

range of error deviations from the 'real' event were input to the control model in order to determine what effect these errors would have on the performance of the resulting control strategies. Reactive and adaptive control policies were compared in order to determine whether expected forecast errors found in experiments with the forecast model were compatible with levels of error that could be tolerated in an adaptive control mode; that is, error levels above which it would be better to use simple reactive policies.

SHORT-TERM FORECASTING

National Weather Service (NWS) and National Oceanographic and Atmospheric Administration (NOAA) forecasting programs have primarily been macro in nature, both temporally and spatially, and therefore not amenable to real-time storm forecasting on the localized basis needed for stormwater control. Under urging by the American Meteorological Society, NOAA has undertaken to improve and expand local forecasting systems such as the Local Flash Flood Warning Systems (LFFWS) by integrating automated ground measurements with radar and satellite data. In addition, according to *McPherson* [1980], the new Prototype Regional Observing and Forecasting Service (PROFS) under development by NOAA in Boulder, Colorado, portends a greater emphasis on meeting short-term forecast needs of metropolitan areas. The goal is to apply this forecasting technology to a wide variety of urban needs, including stormwater control. *Beran and Little* [1978, p. 19] state that the forecasting system associated with PROFS will be designed to utilize

> ... the total available data set to prepare short term (0–3 hr.) extrapolations of the nowcasts, and mid-term (3–12 hr.) forecasts of local weather. The latter would be prepared using physical, numerical, and statistical forecasting techniques designed for each specific region, taking into account the local topography and other surface features. The former would involve a major research effort to learn how best to extrapolate the nowcast data, using physical and statistical methods.

The physically based models referred to here attempt to describe the dynamics of atmospheric processes related to raincell activity within a large mesoscale area (LMSA). Examples include the work of *Amorocho and Wu* [1977], *Gupta and Waymire* [1979], and *Colton* [1976]. These models are particularly valuable for synthetic generation of rainfall and for obtaining more accurate areal estimates of rainfall intensity from point observations. At present it appears that they are too large and unwieldy for real-time operational control situations requiring successive forecasts over short time intervals. They may eventually become feasible for real-time use, however, with the computer power and sophisticated communication system envisioned for the PROFS system.

At the other extreme are statistically based black box type models which attempt to discover correlative patterns in spatially distributed telemetered climatic data and extrapolate based on these patterns. *Phanartzis* [1979] has developed a simple duoseriocorrelation model for the city of San Francisco called RAFORT (RAinfall FOrecast in Real Time). The model is premised on the concept that for a storm moving in a reasonably consistent direction the real-time information from a distant or outlying raingage and a local raingage (where a forecast is desired) are cross-correlated. Short-term forecasting is therefore possible as long as the direction of storm movement allows advance warning at the distant gage. The forecast time increment is one hour.

Nguyen et al. [1978] attempted to apply an earlier version of the forecasting concept developed by *Phanartzis* [1979] to the Montreal area. Since a radar-based storm tracking capability is envisioned for this area, the model was designed to consider radial to point correlations rather than point to point correlations as in the San Francisco work. It was assumed that radar could effectively track incoming storms to a 50-mile (80-km) radius. This would provide about a one hour lead time for forecasting rainfall at Dorval airport on Montreal Island if a satisfactory correlation could be found between depths at a 50-mile radius and depths at Dorval.

The city of Seattle [*Leiser*, 1974] is planning to utilize a real-time storm forecasting model for the automated portion of their combined sewer system that would group storm events according to the following parameters: (1) storm direction, (2) precipitation amount (light, medium, heavy), and (3) duration (short, medium, long).

A priori rule curves for regulators in the system would be developed for each storm classification. Ongoing storms would be classified (and reclassified) in real time from radar and raingage data. For this reason this approach should be categorized as a combined reactive-adaptive approach. As of this writing, no published results are available that we are aware of.

The Illinois State Water Survey [*Changnon and Semonin*, 1978] has been conducting a comprehensive real-time storm forecasting effort in the Chicago area called the Chicago Hydrometeorological Area Project (CHAP). This work has concentrated on establishing radar-raingage statistical relationships and improving areal estimates of rainfall, but storm tracking procedures similar to the work in Montreal have been experimented with.

A more sophisticated storm tracking procedure using the discrete Kalman filter has been presented by *Johnson and Bras* [1980]. Rather than relying on historically based regression relations as in the Montreal effort, this model assumes each event to be unique and makes forecasts based on an elaborate statistical analysis of the ongoing event. Error covariance estimates of forecasted rainfall are produced along with the forecasts; these are valuable for risk analysis. Also, unlike Box-Jenkins type time series analysis [*Box and Jenkins*, 1976], a distinction is made between observation error and model error. This can be either an advantage or a disadvantage, depending on the situation, since it requires more information for estimating appropriate covariances. It appears that this model is best suited to broad, relatively homogeneous frontal storms with little localized orographic influence.

Perhaps better suited to more spatially nonhomogeneous storm conditions is a model developed by *Trotta* [1976] and reported by *Trotta et al.* [1977] called FORCST which relies on an autoregressive structure for predictions. The model allows the analyst to weight historical storm patterns against the actual ongoing event in a convenient fashion. For example, early in the storm there may be more reliance on historical patterns since little is yet known of the current event. This is called the base line forecasting model since its parameters are estimated from available historical data. As the storm progresses, the weighting factors can be altered to reflect increasing confidence in information on the storm. Experience in how to adjust the weighting factors in real-time could be gained by performing a large number of simulations with historical events. This was considered to be beyond the scope of our present study. The model is discussed in more detail in the following section.

AUTOREGRESSIVE-TRANSFER FUNCTION MODEL

The model developed by *Trotta* [1976] is written as follows:

$$\hat{y}_{i,t+1} = \sum_{\tau=0}^{p-1} \hat{a}_{i\tau}(t) R_{i,t-\tau} + \sum_{j \in J(i)} \sum_{\tau=0}^{p-1} \hat{b}_{j\tau}(t) I_{j,t-\tau} \qquad \text{locations} \quad i = 1, \cdots, n \qquad (1)$$

where

$\hat{y}_{i,t+1}$	rainfall forecast (depth or intensity) at location i for lead time of one period from current real-time period t;
$R_{i,t-\tau}(\tau = 0, \cdots, p-1)$	previous rainfall measurements at location i;
$I_{j,t-\tau}(\tau = 0, \cdots, p-1)$	other measurable inputs, such as adjacent and outlying raingage depth, meteorological data such as wind speed and direction, or radar data;
$\hat{a}_{i\tau}(t)$, $\hat{b}_{j\tau}(t)$	current (period t) estimates of model parameters. The parameters are shown as functions of t to convey the idea that they are nonstationary and are updated as new information becomes available;
$J(i)$	set of other measurable inputs of consequence for forecast location i;
p	model order.

Graupe [1976] would define this as an autoregressive-transfer fucntion model, since the second term includes measurable inputs other than the time series being forecast. In obtaining minimum variance estimates of the parameters, based on historical as well as ongoing storm data, weighting factors ω_t are included in the least squares criteria which allow consideration of historical storm patterns against the perceived pattern for the current event. Suspect historical data can obviously be given a lower weight. Let the column vector $[\mathbf{x}_t]_i$ represent all measurable inputs ($R_{i,t-\tau}$ and $I_{j,t-\tau}$) on the right-hand side of (1) for real-time period t and vector $[\hat{\mathbf{a}}(t)]_i$ as all parameters ($\hat{a}_{i\tau}$ and $\hat{b}_{j\tau}$) on the right-hand side of (1). Dropping the subscript i for convenience, (1) can now be simply written as

$$\hat{y}_{i,t+1} = \hat{\mathbf{a}}(t)^T \mathbf{x}_t \qquad \text{for each location} \quad i = 1, \cdots, n \qquad (2)$$

As shown by *Graupe* [1976] and several others, a sequential regression algorithm can be used to update estimates of the model parameters that requires certain information from the previous period only. As actual rainfall observations $R_{i,t+1}$ become available during period $t + 1$, we can update the parameters as follows for each location i:

$$\hat{\mathbf{a}}(t+1) = \hat{\mathbf{a}}(t) + P_{t+1}\omega_t \mathbf{x}_t (R_{i,t+1} - \hat{y}_{i,t+1}) \qquad (3)$$

where the matrix P_{t+1} is recursively computed by

$$P_{t+1} = P_t - [(P_t \omega_t \mathbf{x}_t \mathbf{x}_t^T P_t)/(1 + \omega_t \mathbf{x}_t^T P_t \mathbf{x}_t)] \qquad (4)$$

for each location $i = 1, \cdots, n$. If, for example, we assume that the other measurable inputs correspond to raingages only and that forecasts are desired at each raingage location, then P_t is an $n \times n$ matrix which represents estimated parameter error covariances.

If we desire forecasts for a lead time k greater than one, (1) is simply applied recursively using current parameter estimates and forecasted values of the inputs. According to *Box and Jenkins* [1976], estimates of forecast error covariances for any lead time can be obtained based on the 'weights' calculated from the parameter estimates of the autoregressive and transfer function portions of the model, as well as the estimated variance of the residuals.

CASE STUDY

The city of San Francisco has a dense automated raingage network (i.e., approximately one gage per 900 acres) that has been operational for over eight years. *Phanartzis* [1979] reports that the majority of winter storm fronts (over 90%) tend to come from a northwesterly direction, so that there is a strong correlation with rainfall measurements northwest of the city in Marin County (Figure 1). *Amorocho and Wu* [1977] have observed that these storms have a banded structure with a number of shortlived cells of high rainfall activity moving roughly parallel along the front.

The North Shore area of San Francisco was pinpointed for study since intensive work has been conducted by the city in developing a real-time automatic control system (RTACS) for storm runoff in this area. An integrated, city-wide system is the ultimate goal [*San Francisco Department of Public Works*, 1978]. Hourly storm records for the period October 9, 1972 to March 13, 1975 inclusive were taken from the rain gages at Novato (gage 10) and Tomales (gage 11), both located in Marin County north of San Francisco, as well as the gage at the Federal Office Building in downtown San Francisco (see Figure 1). Later studies used additional records from gages in the North Shore area of San Francisco (gages 14, 26, 28, and 29). Figure 2 gives the locations of these gages.

Most of the historical data record (29 identifiable storm events) was used for the base line forecasting model. The storms were simply linked together to form a continuous time series for the base line identification. Model order p was found by an iterative underfitting-overfitting procedure. Order $p = 2$ gave the lowest residual error variance, which is consistent with cross correlation results between the local and outlying raingages reported by *Phanartzis* [1979]. Four additional storm events in the record were utilized as simulated real-time events for testing and analyzing forecast capability. As each of these events were realized in the real-time simulations, the forecast model parameters were adjusted accordingly. Storms 32 and 33 (occurring on March 6 and 13, 1975, respectively) are categorized as low duration–high intensity storms. Storms 30 and 31 of February 12 and 18 are considered to be high duration–low intensity. Both types are considered to be potentially overflow producing storms.

As an illustration, Figure 3 shows the parameter trace for a base line identification ($p = 2$), using the 15 storms designated as low duration–high intensity, plus the trace for storm 31. In this case the ω_t for all storms were set at 1.0, so the parameters change very little during storm 31. Error residuals are shown in Figure 4.

FORECAST RESULTS

The cumulative plots of the actual storm and the corresponding forecast values for lead time of one hour are shown in Figures 5–8. A base line identification using all 29 storms provided initial parameter estimates for these forecasts, with $\omega_t = 1.0$. After a forecast with lead time equal to one hour is made, the actual measurement is used for the next forecast. It is evident that storm 30 (the longest duration storm) was over-

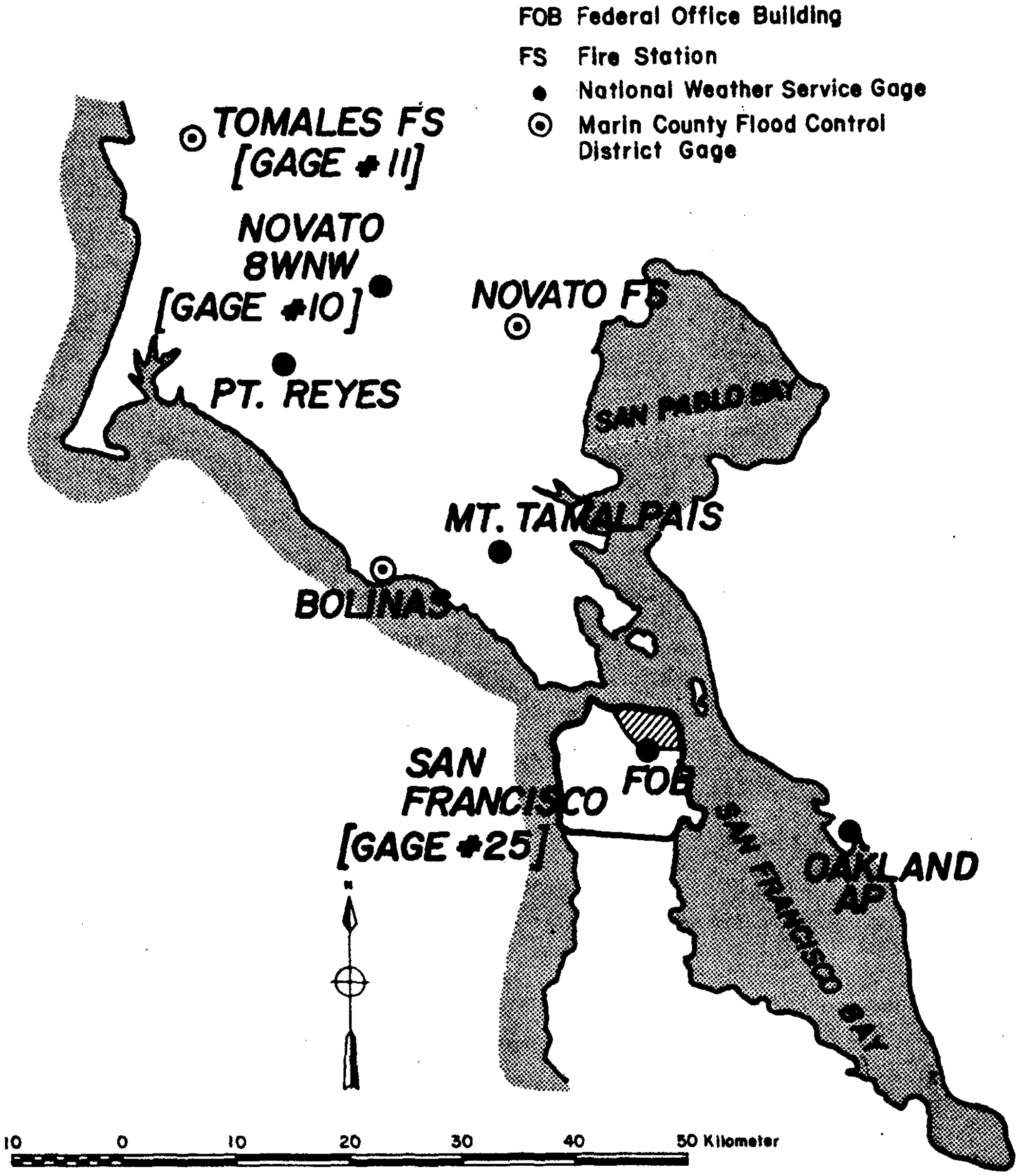


Fig. 1. Locations of outlying raingages.

forecasted, storms 33 and 31 were under-forecasted, and storm 32 was under-forecasted early but over-forecasted later.

A cross-correlation analysis was conducted for comparing actual values with the forecasted values. *Granger and Newbold* [1977] suggest that the best forecast will be the one most correlated with the actual values, provided that their means are close and the standard deviation of the forecast is equal to the product of the degree of the duoseriocorrelation ρ between the forecasted and actual values, and the standard deviation of the actual values.

The autocorrelation for the forecasts and the corresponding forecast errors, as well as the cross correlation between them, are shown in Tables 1 and 2. The results for storms 32 and 33 are probably questionable due to the small number of sampled events. The results for storms 30 and 31 are based on a larger number of samples and therefore give a better indication of whether most of the information has been utilized. Note that the 95% confidence limits are approximated by $2\sigma \simeq 1.96/\sqrt{N}$ where N is the number of samples.

Forecasts for lead times up to four hours in the future were generated for storms 31 and 33. The average absolute forecast errors in percent (AAFE) are tabulated in Tables 3 and 4 and are determined from

$$\text{AAFE} = \frac{100}{(NL)(K)} \sum_{i=1}^{NL} \sum_{k=1}^{K} \frac{|R_{i,t+k} - \hat{y}_{i,t+k}|}{R_{i,t+k}} \qquad (5)$$

where

$R_{i,t+k}$	actual measured rainfall at location i;
t	current real-time period;
k	number of forecast time intervals beyond t;
$\hat{y}_{i,t+k}$	forecasted rainfall for the time period $t + k$;
K	lead time or maximum number of time intervals into the future that the forecast covers;
NL	number of locations considered.

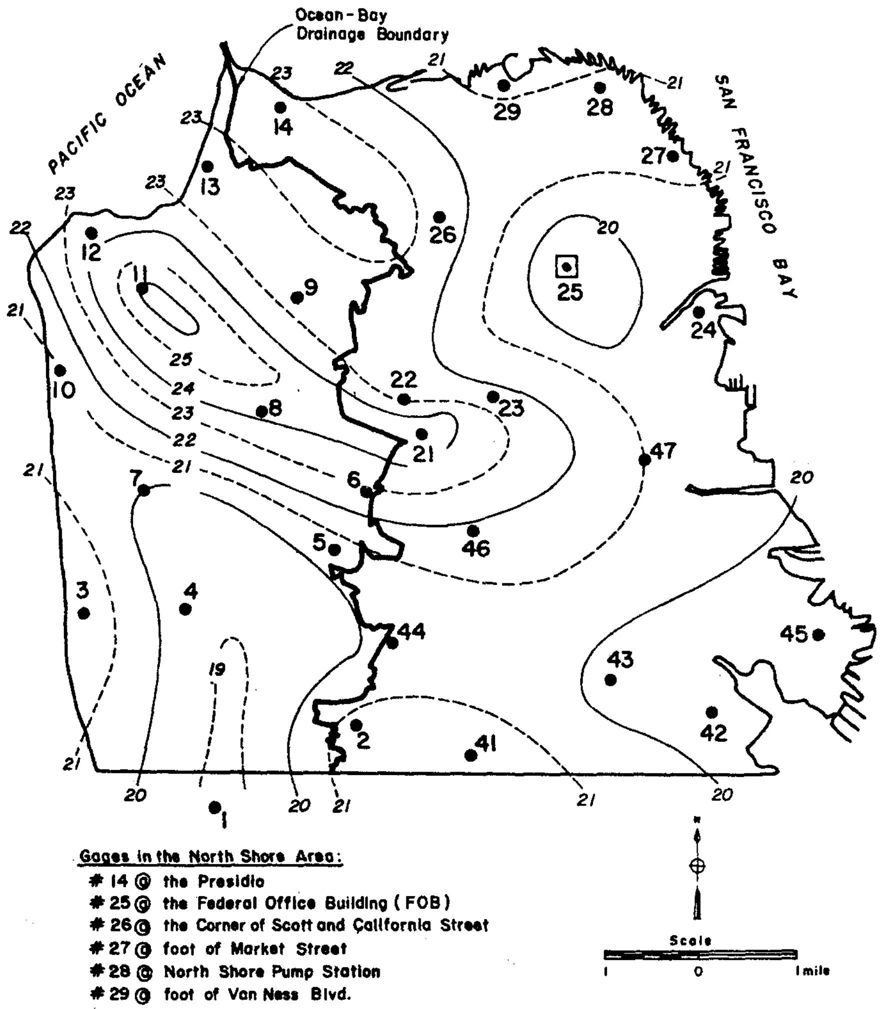


Fig. 2. Isohyetal map of San Francisco showing average annual rainfall in inches, based on records for 1972–1978, and locations of raingages [*Phanartzis*, 1979].

The results show, with some exceptions, an increasing error with increasing lead time. In general the errors in Table 4 tend to decrease as the event unfolds, indicating that improvement in the forecast occurs as more information is obtained. No such conclusion can be drawn from Table 3. Forecasts for periods late in the storms are not included in Tables 3 and 4 since the benefits of such forecasts would probably be marginal.

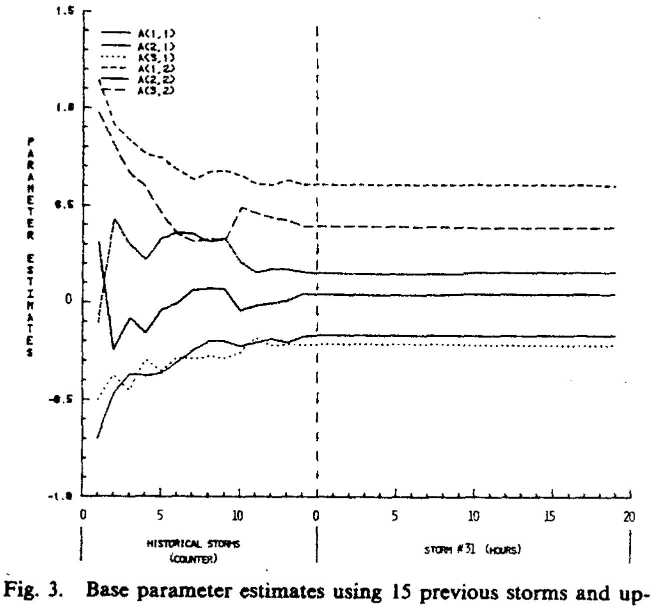


Fig. 3. Base parameter estimates using 15 previous storms and update using storm 31 as a real event.

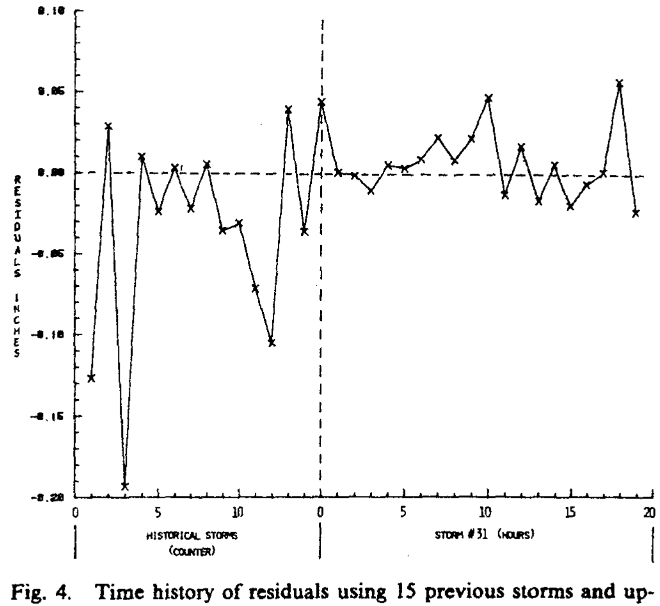


Fig. 4. Time history of residuals using 15 previous storms and update using storm 31 as a real event.

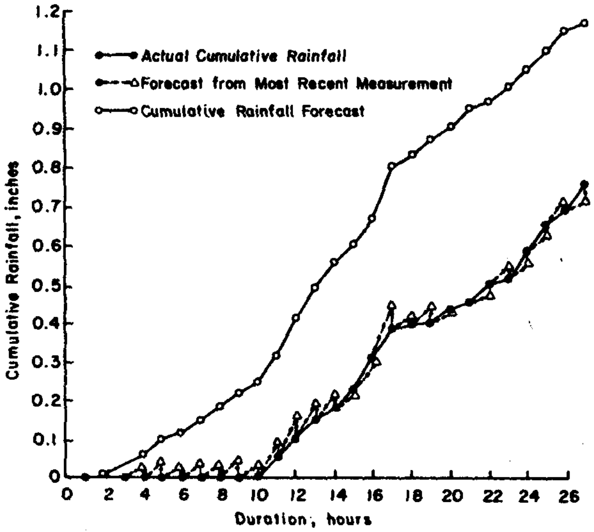


Fig. 5. Cumulative forecasts at gage 25 using remote gages 10 and 11 for storm 30.

Some additional experiments were conducted to determine the effect on forecasting error of (1) being able to catgorize storms based on radar data, (2) using the weighting factors ω_n, and (3) use of additional local gages in the forecasting model. Results are reported by *Labadie et. al.* [1978], but appear to be rather inconclusive. Much more work is needed for determining how these methods can be used to reduce forecast error.

SENSITIVITY OF ADAPTIVE OPERATING STRATEGIES TO FORECAST ERROR

Tables 3 and 4 provide some indication of the magnitudes of forecast errors that could be expected during a real-time event. These results are admittedly only an indication, but it seems reasonable to suggest that the errors could be further reduced through use of radar for storm categorization and tracking, as well as proper use of the storm weighting factors. Next we would now like to determine how sensitive the performance of the system for controlling stormwater in real-time is to the degree of forecast error. These sensitivities can then be compared to the expected forecast error magnitudes given in Tables 3 and 4.

The San Francisco North Shore Outfalls Consolidation Project [*San Francisco Department of Public Works,* 1978] was used as a case study, which is the same area in which the rainfall forecasting experiments were conducted (location indicated as shaded portion of Figure 1). This project involves the planned use of automatically controlled gates in a large shoreline tunnel as a means of minimizing untreated overflows. Details of the project are presented by *Labadie et al.* [1980] and therefore are not repeated here. A dynamic programing technique using orthogonal polynomials was applied in order to deal with dimensionality difficulties associated with the optimal control problem.

Tests were conducted to compare reactive and adaptive control policies and to ascertain the effect of forecast error on performance of the control strategies. A stormwater control package (SWCP) was used for this purpose, which integrates program FORCST with the dynamic programing (DP) algorithm for obtaining optimal stormwater control strategies

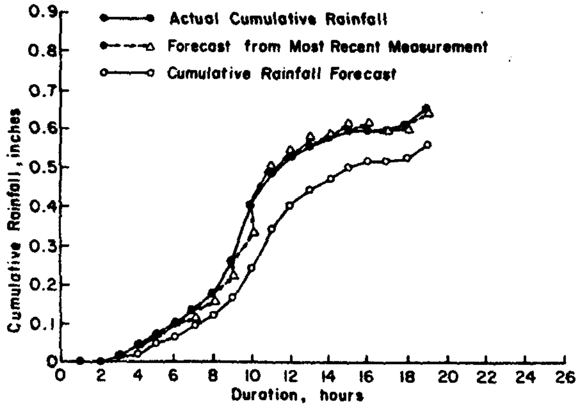


Fig. 6. Cumulative forecasts at gage 25 using remote gages 10 and 11 for storm 31.

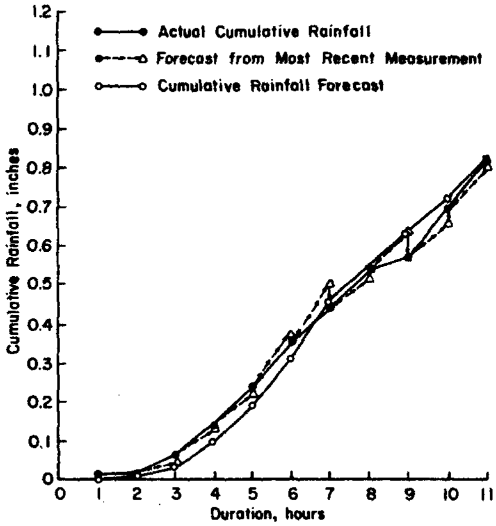


Fig. 7. Cumulative forecasts at gage 25 using remote gages 10 and 11 for storm 32.

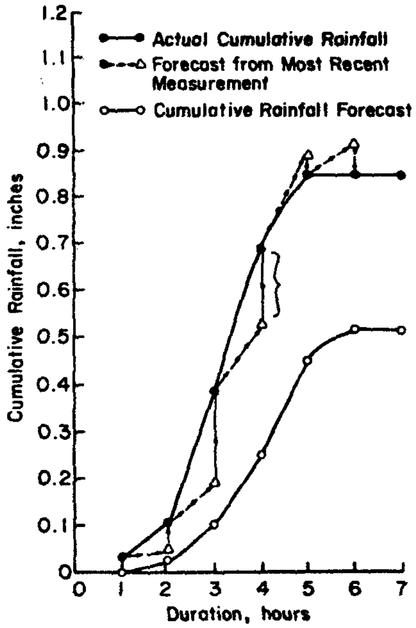


Fig. 8. Cumulative forecasts at gage 25 using remote gages 10 and 11 for storm 33.

TABLE 1. Autocorrelation Functions of the Forecast and Error Series for FORCST

Storm Identification Number	Forecast							Error							MSE*
	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_1	r_2	r_3	r_4	r_5	r_6	r_7	
30	0.389	0.190	0.159	0.236	0.223	−0.083	−0.175	−0.038	0.341	0.107	−0.136	0.361	−0.148	0.246	0.0010
31	0.766	0.429	0.111	−0.134	−0.302			0.283	0.026	−0.045	−0.134	−0.169			0.0005
32	0.792	0.526	0.011					−0.353	0.304	−0.187					0.0011
33	0.430	−0.592						0.422	−0.504						0.0104

* MSE = mean square error = $(1/N)\sum_{t=1}^{N} \varepsilon_t^2$ where: $\varepsilon_t = y_t - \hat{y}_t$, y_t is the actual hourly rainfall depth, and $\hat{y}_t$ is the forecasted hourly rainfall depth.

TABLE 2. Cross Correlation Functions of the Forecast Error Series

Storm Identification Number	Forecast versus Error						Correlation Coefficient Bounds, 95%
	r_0	r_1	r_2	r_3	r_4	r_5	
30	−0.478	−0.001	−0.039	0.202	0.079	0.024	± 0.377
31	0.111	−0.179	−0.381	−0.404			± 0.450
32	−0.446	−0.198					± 0.591
33	0.066						± 0.741

[*Morrow and Labadie*, 1980]. A fully dynamic, unsteady flow routing model is then used to predict flows in the sewer system as a result of the computed strategies, as well as to simulate the prototype system (Figure 9).

For the reactive case a typical overflow-producing storm event was input as the real storm into the SWCP. Since the reactive case uses no forecast information, we let current measured (i.e., as simulated by the input storm event) inflow rate R_{it} at the beginning of period t remain constant over that period when running the SWCP. The reactive case is then run period by period with controls computed under the above restrictions. The actual inflows are then used to update the flow levels in each section of the tunnel resulting from the previous period operation, and the process is repeated for the next period. Fifteen-minute control intervals were used and a 5-year design storm was specified by the staff of the San Francisco Department of Public Works [see *Labadie et al.*, 1978].

For the adaptive case, forecast lead times of $L = 1$ (15 minutes) and $L = 3$ (45 minutes) were run. A range of prespecified forecast error magnitudes were used in order to see what effect these errors would have on the control strategy performance. The results are summarized in Figure 10. Only the results from deliberately overestimated inflows are shown. Some of the underestimated inflow results are suspect due to linearization errors in the unsteady flow model. Mixed forecasts were not considered. The values on the ordinate represent weighted overflows. Weighting factors were attached to overflows at certain locations in order to reflect a preference for avoiding untreated overflows near boat berthing and tourist areas. Again, details on these aspects of the control problem can be found in the work by *Labadie et al.* [1980]. Notice that the longer forecast lead time ($L = 3$) gave better results.

In order to obtain a rough idea of the worth of the forecast information, we can attempt to compare Figure 10 with the results in Tables 3 and 4. The comparison can only be rough since the $L = 3$ case represents a 45-minute forecast lead time in 15-minute increments, whereas the first column of Tables 3 and 4 is for a 1-hour lead time. This means it is assumed that interpolations of the one hour forecasts over 15-minute intervals would have approximately the same average forecast errors as shown in the first column of Tables 3 and 4. In addition, actual storms were used in obtaining the results of Tables 3 and 4 and a hypothetical storm inflow for the control experiments.

Another assumption that could be important is that the errors generated in the watershed runoff modeling were assumed negligible in comparison to the rainfall forecast errors. This allowed use of direct inflows into the shoreline tunnel for the control experiments. We believe this to be a reasonable assumption for the San Francisco case, but it may not be so for other areas. The Laguna Street watershed, located in the North Shore area, has been extensively modeled using the San Francisco stormwater model [*Kibler and Roesner*, 1975]. An

TABLE 3. Average Absolute Forecast Error in Percent for Storm ID31, a High Duration Storm on February 18, 1975

Time, hr	Forecast Lead Times, hr			
	1	2	3	4
1	0.0	0.0	33.3	50.0
2	0.0	10.0	55.6	50.0
3	0.0	16.7	22.2	79.2
4	66.7	33.3	55.6	60.4
5	0.0	25.0	16.7	31.2
6	0.0	12.5	16.7	20.8
7	50.0	50.0	51.8	53.2
8	25.0	40.0	48.3	39.3
9	44.4	50.8	46.4	99.8
10	50.0	37.5	78.3	321.2
11	25.0	82.5	255.0	257.9
12	20.0	160.0	184.0	563.3

TABLE 4. Average Absolute Forecast Error in Percent for Storm ID 33, a Low Duration Storm on March 13, 1975

Time, hr	Forecast Lead Times, hr			
	1	2	3	4
1	100.0	100.0	100.0	100.0
2	75.0	85.6	89.4	90.5
3	70.4	73.9	63.8	72.9
4	51.6	38.3	58.9	69.2
5	25.0	62.5	75.0	75.0

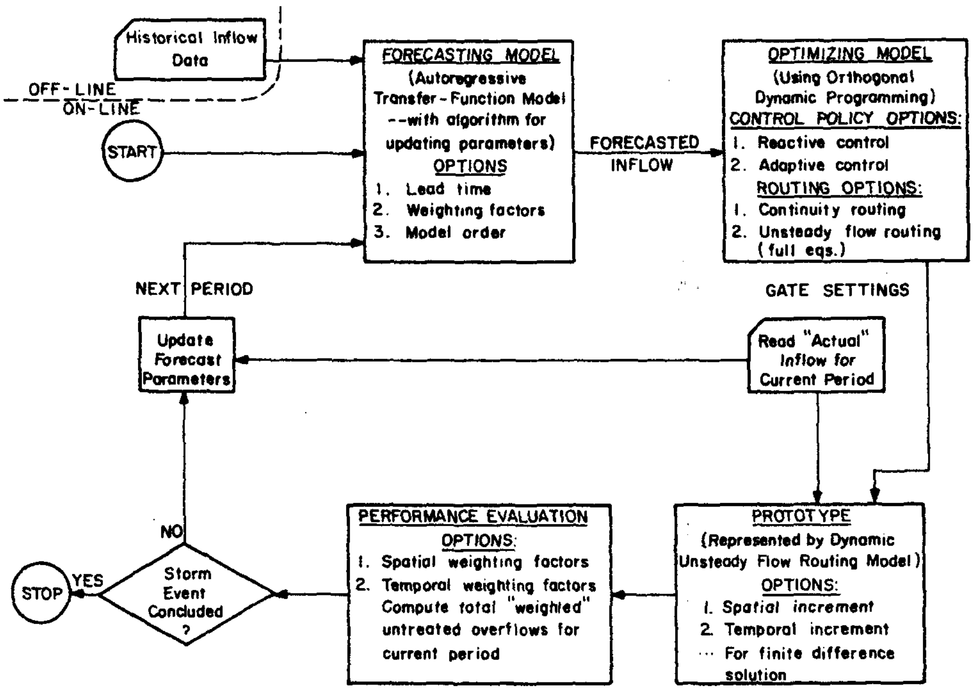


Fig. 9. *Flow chart of the stormwater control package (SWCP) (note: the current package assumes that rainfall data have been input into an accurate watershed model for predicting direct storm inflow into the sewer system).*

excellent fit was attained between computed and observed stormwater runoff.

It should be noted that if the times of concentration of watersheds contributing to interceptors are larger than the desired control interval, then good predictions over a lead time of one or more periods may be obtained with just rainfall measurements and a reliable rainfall runoff model. This particular area of San Francisco is, however, extremely steep, with a time of concentration of less than 10 minutes.

It can be seen from Figure 10 that the $L = 3$ case is better than the reactive case, even for overestimated forecast errors of up to 70%. Except for periods early in the low duration, high intensity storm, the errors in the first column of Tables 3 and 4 are at or under this limit.

Obviously much more work is needed and many of the assumptions we have made need to be relaxed. Our hope is that these preliminary results will serve to stimulate other interested researchers.

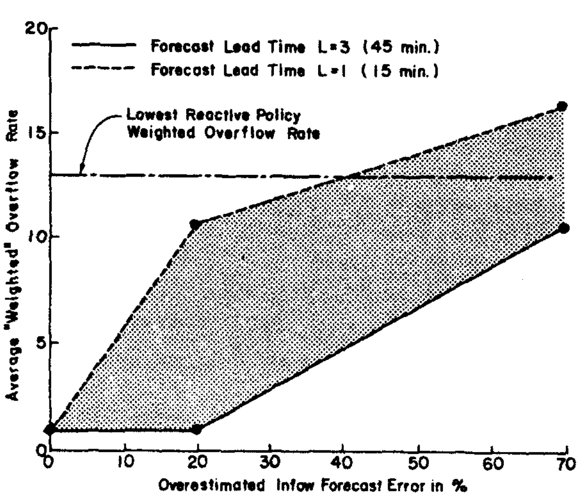


Fig. 10. Comparison of performance of adaptive and reactive control policies.

Acknowledgments. The writers would like to acknowledge the important previous work of Paul Trotta on the forecasting model. We are also grateful to Chris Phanartzis and Gene Handa, consultant and staff member, respectively, of the Clean Water Program, City and County of San Francisco, for their helpful interaction with us. The research leading to this paper was partially supported by the Office of Water Research and Technology, U.S. Department of Interior, as authorized by the Water Research Development Act of 1978. Contents of this paper do not necessarily reflect the views of the Office of Water Research and Technology.

REFERENCES

Amorocho, J., and B. Wu, Mathematical models for the simulation of cyclonic storm sequences and precipitation fields, *J. Hydrol.*, *32*, 329–345, 1977.

Beran, D. W., and C. G. Little, Technology transfer in PROFS, paper presented at the 1978 Technical Exchange Conference, Air Weather Serv., Air Force Acad., Colorado Springs, Colo., Nov. 28 to Dec. 1, 1978.

Box, G. E. P., and G. M. Jenkins, *Time Series Analysis: Forecasting and Control,* Holden-Day, San Francisco, 1976.

Changnon, S. A., Jr., and R. G. Semonin, Chicago area program: A major new atmospheric effort, *Bull. Am. Meteorol. Soc.*, *59*(2), 153–160, 1978.

Colton, D. E., Numerical simulation of the orographically induced precipitation distribution for use in hydrologic analysis, *J. Appl. Meteorol.*, *15*(12), 1241–1251, 1976.

Granger, C. W. J., and P. Newbold, *Forecasting Economic Time Series,* Academic, New York, 1977.

Graupe, D., *Identification of Systems,* Van Nostrand Reinhold, New York, 1976.

Gupta, V. K., and E. C. Waymire, A stochastic kinematic study of subsynoptic space-time rainfall, *Water Resour. Res.*, *15*(3), 637–644, 1979.

Jamieson, D. G., and J. C. Wilkinson, River Dee research program, 3, A short-term control strategy for multipurpose reservoir systems, *Water Resour. Res.*, *8*(4), 911–920, 1972.

Johnson, E. R., and R. L. Bras, Multivariate short-term rainfall prediction, *Water Resour. Res., 16*(1), 173–185, 1980.

Kibler, D. F., and L. A. Roesner, The San Francisco stormwater model for computer simulation of urban runoff quantity and quality in a combined sewer system, report, Water Resour. Eng., Inc., Walnut Creek, Calif., 1975.

Labadie, J. W., Morrow, D. M., and R. C. Lazaro, Urban stormwater control package for automated real-time systems, *Rep. C 6174, Colo. State Univ.*, Fort Collins, Colo., 1978.

Labadie, J. W., D. M. Morrow and Y. H. Chen, Optimal control of unsteady combined sewer flow, *J. Water Resour. Plann. Manage. Div. Am. Soc. Civ. Eng.*, 106(WRI), 205–223, 1980.

Leiser, C. P., Computer management of a combined sewer system, *Proj. 11022 ELK, Program Elem. 1BB034,* Storm and Comb. Sewer Sect., Advan. Waste Treat. Res. Lab., U.S. Environ. Prot. Agency, Cincinnati, Ohio, 1974.

McPherson, M. B., Study of integrated control of combined sewer regulators, report, EPA grant R806702010, Munic. Environ. Res. Lab., Office of Res. and Dev., U.S. Environ. Prot. Agency, Cincinnati, Ohio, 1980.

Morrow, D. M., and J. W. Labadie, Urban stormwater control package for automated real-time systems, paper presented at the Canadian Hydrology Symposium: 80, Natl. Res. Council of Can., Toronto, May 26–27, 1980.

Nguyen, V.-T.-V., M. B. McPherson, and J. Rousselle, *Tech. Memo. 35,* Urban Water Resour. Res. Program, Am. Soc. Civ. Eng., New York, 1978.

Phanartzis, C. A., Rainfall prediction, progress rep., Wastewater Program, City and County of San Francisco, Calif., 1979.

San Francisco Department of Public Works, Demonstrate real-time automatic control in combined sewer systems, *Progress Rep. 3,* Bur. of Sanit. Eng. and Water Resour. Eng., Inc., San Francisco, Calif., 1978.

Trotta, P. D., On-line adaptive control for combined sewer systems, Ph.D. dissertation, Dep. of Civ. Eng., Colo. State Univ., Fort Collins, 1976.

Trotta, P. D., J. W. Labadie, and N. S. Grigg, Automatic control strategies for urban stormwater, *J. Hydraul. Div. 103*(HY12), 1977.

(Received January 30, 1981;
revised June 1, 1981;
accepted June 8, 1981.)

APPENDIX B

USER GUIDE TO THE STORMWATER CONTROL PACKAGE

A. PROCEDURE FILE

A procedure file gets and organizes all useful data files and command files for run production. Included also in the procedure file are initialization statements for the system, as well as space and memory allocation statements for the hardware.

An example procedure file for the SWCP is included in Table 1 for the NOS 2.3 operating system for the Cyber 825/835 main frame computer system. This has been set up for the three stage NSOC problem which was discussed in Chapter III. For other systems, the procedure file will have to be changed. Since the commands are descriptive in the sense that their names suggest the function they perform, the task of writing procedure files for other systems will hopefully not be difficult. A general flow chart for the SWCP is given in Figure 1.

B. ORTHOGONAL DYNAMIC PROGRAMMING DATA (SUBROUTINE DYNPRO)

B.1 Overview

The dynamic programming formulation for the SWCP presents a multidimensional problem which defies solution by standard DP methods. The dimension of the state and decision vectors is one greater than the lead time L (from the forecast model). For example, a lead time of three control intervals results in a four dimensional DP problem.

Multidimensional DP problems can be decomposed by several techniques including incremental DP, discrete differential DP, and successive approxi-

Table 1. Procedure File for Batch Operation

```
/JOB
TC99        ,T  92. DENNIS MORROW
/USER
   CLASS,BG.
ROUTE,OUTPUT,DC=PR,ID=01,UJN=TC99,DEF.
*  %%%%%%%%%%%%%%%%
*  SKIPR,INPUT,1.
*  %%%%%%%%%%%%%%%%
GET,MORROW=LGO1.
  FTN5,ANSI=T,B=DENNIS,L= LIST ,OPT=3,ET=F,LO=  M/R.ET=F.
REWIND,LIST.
COPYSBF,LIST,IST,999.
   ROUTE, IST,DC=PR,ID=01,UJN=TC99.
RETURN,LIST.
   SKIPR,INPUT,1.
*
REWIND,DENNIS.
  COPYL,MORROW,DENNIS,LGO.
  REPLACE,LGO=LGO1.
  RETURN,DENNIS,MORROW.
*
COPYBR,INPUT,TAPE15.   3-STAGE DP INPUT TAPE (STAGES BROKEN AT GATES).
COPYBR,INPUT,TAPE19.   SIMULATION ROUTING MODEL INPUT DATA TAPE (WHOLE SYSTEM)
COPYBR,INPUT,TAPE12.   REAL STORM PARAMETERS ARE INPUT FROM TAPE 12.
REWIND,TAPE12,TAPE15,TAPE19.
LDSET,PRESET=NGINF.MAP=BS.
LGO,INPUT,OUTPUT,INPUT,OUTPUT,TAPE12,TAPE15,TAPE19,TAPE16,*PL=50000.
* TAPE 16 IS FOR OUTPUT FROM THE DP.
RETURN,LGO.
RWF,OUTPUT.
*
*  REPLACE,TAPE12.
*
COPYSBF,TAPE16.
COPYSBF,TAPE15.
COPYSBF,TAPE19.
*
EXIT.
ROUTE,OUTPUT,DC=PR,ID=01,UJN=TC99,DEF.
* SORRY!!!
RWF,OUTPUT.
COPYSBF,TAPE16.
COPYSBF,TAPE15.
COPYSBF,TAPE19.
*OPYSBF,TAPE12.
SKIPR,INPUT,3.
COPYSBF,INPUT,OUTPUT,999.
*
/EOR
```

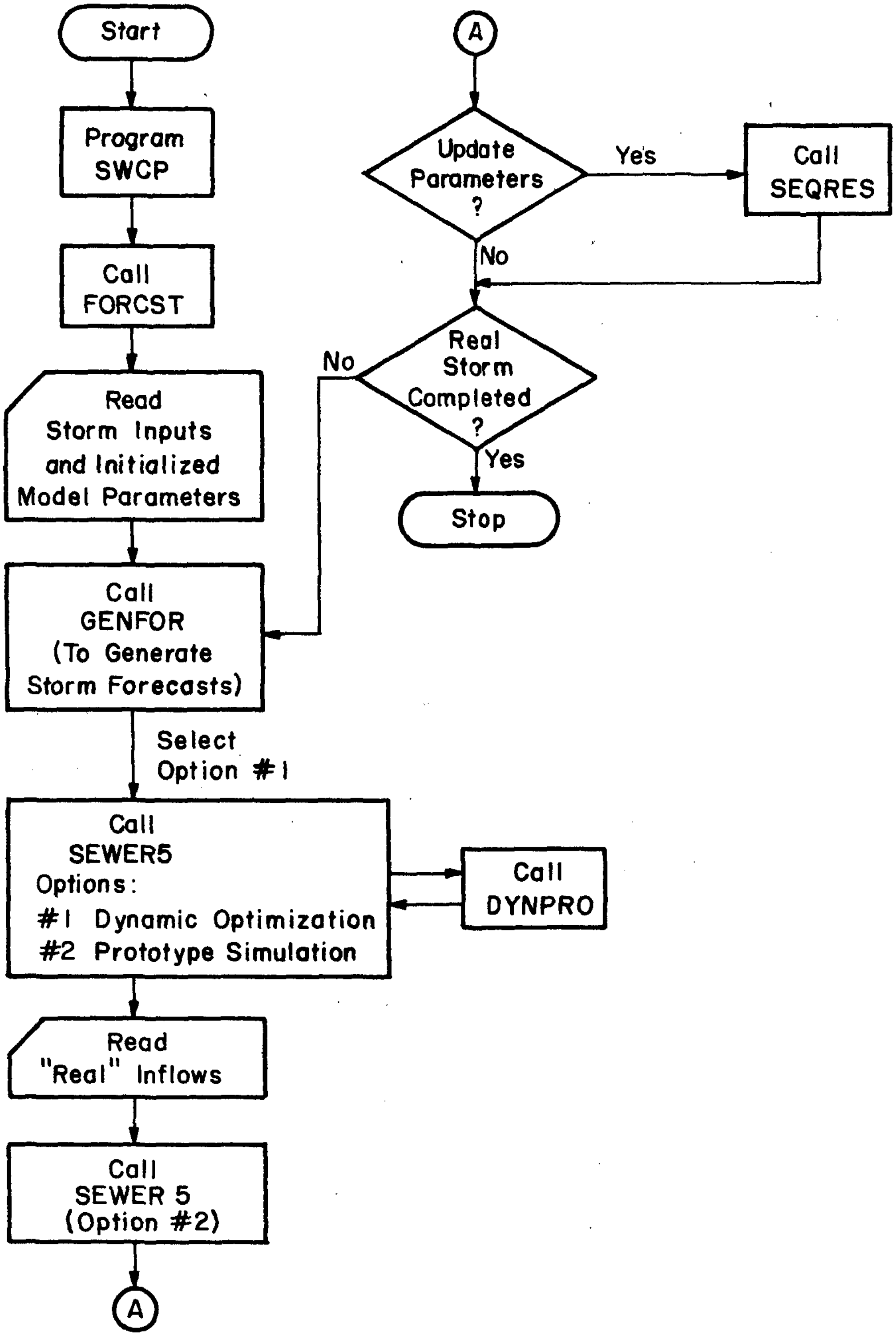


Figure 1. Flow Chart for SWCP

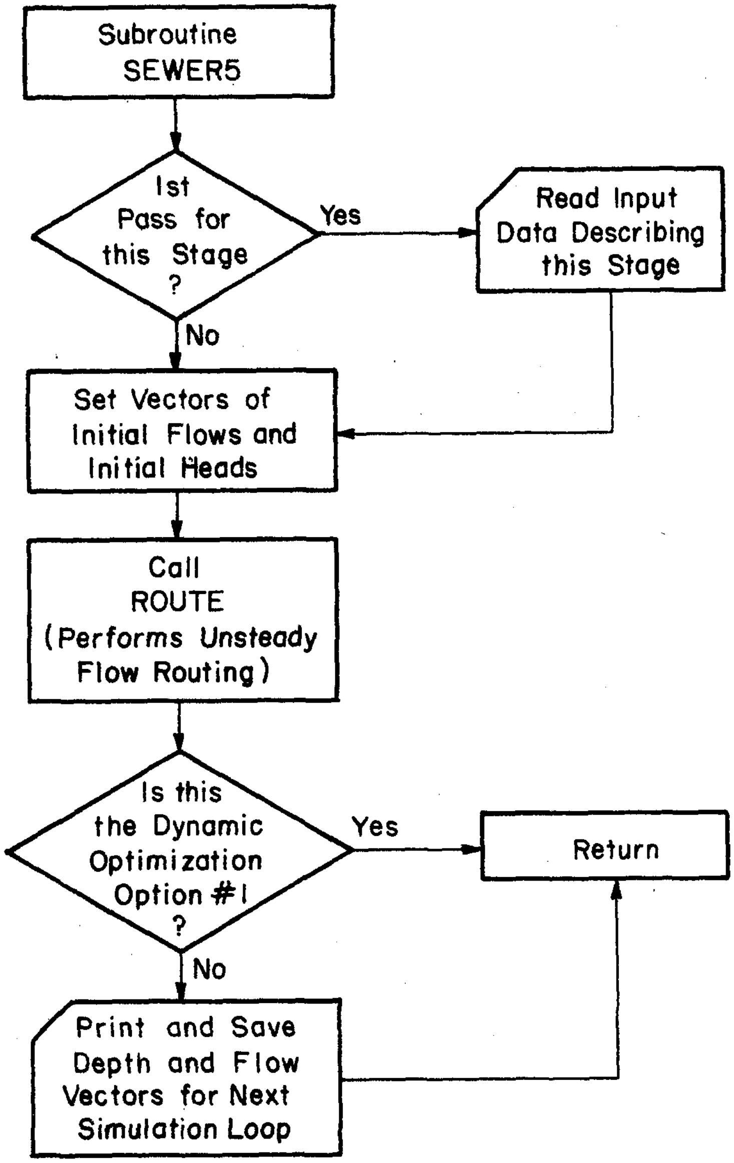


Figure 1. continued

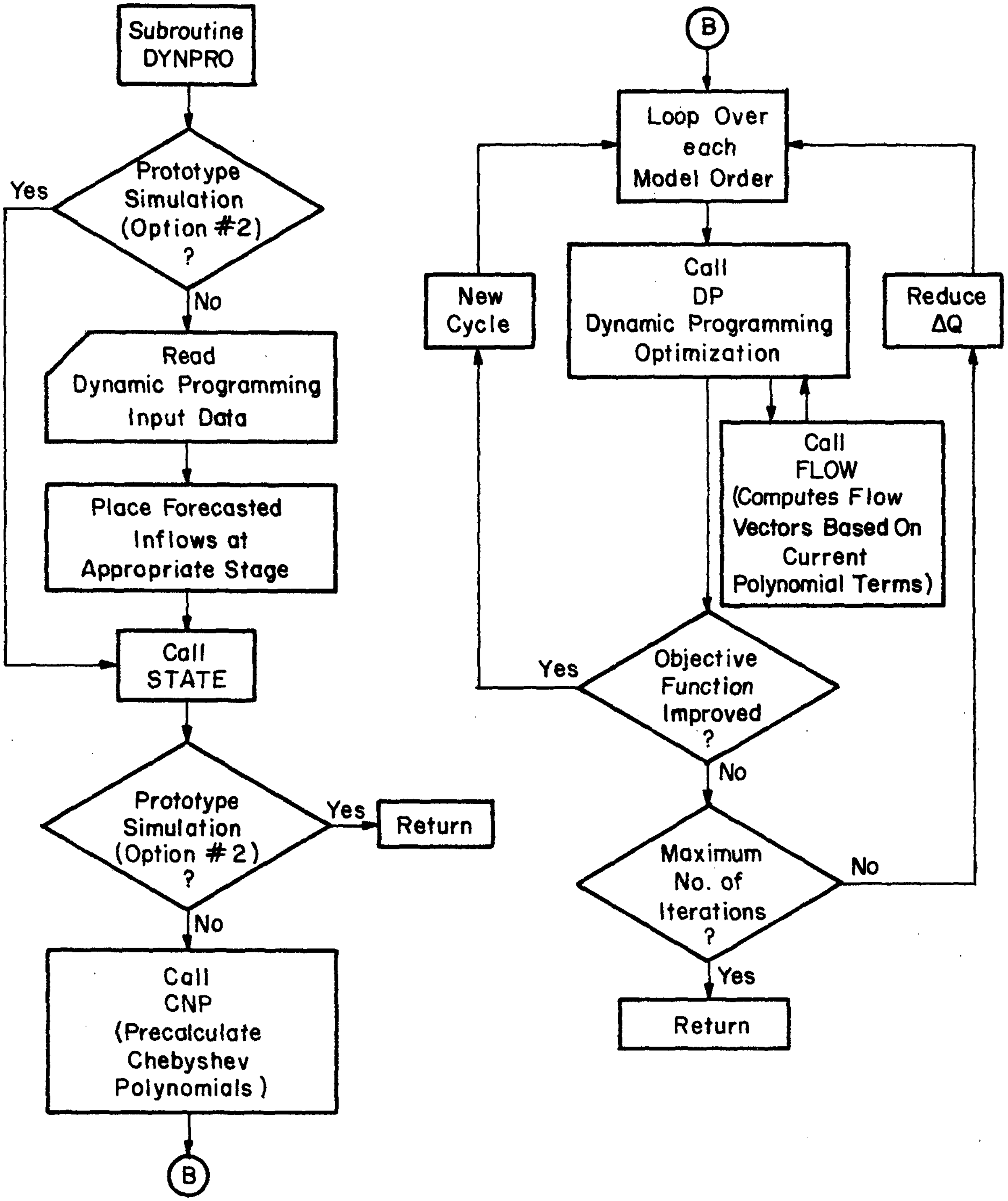


Figure 1. continued

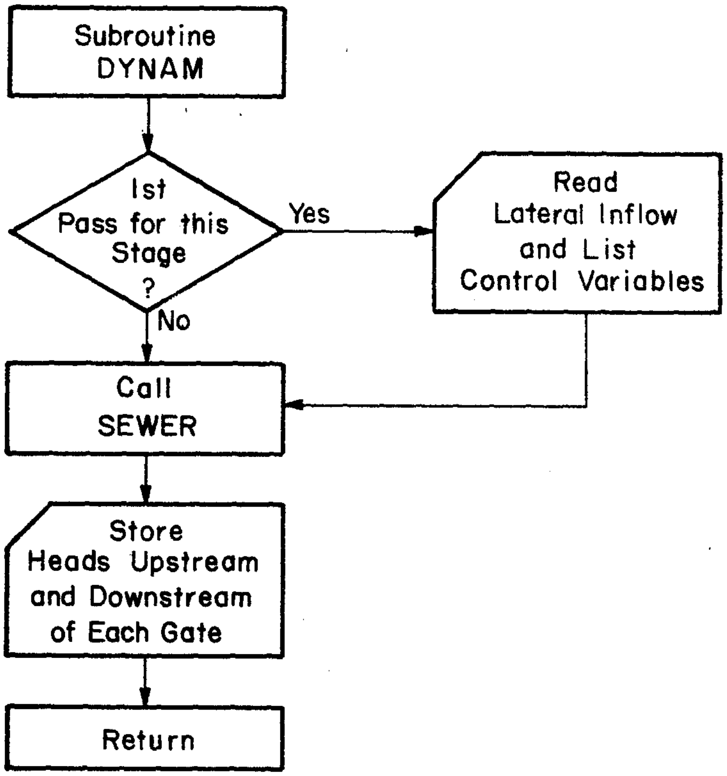


Figure 1. continued

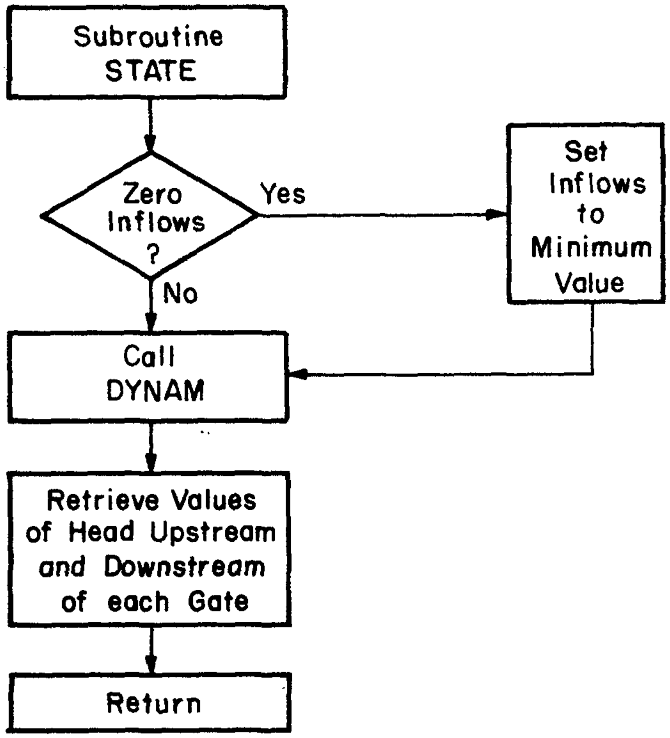


Figure 1. continued

mations. However, the heuristic algorithm presented here uses orthogonal polynomials for decomposition of the L + 1 dimensional problem into a series of one-dimensional problems which can be easily solved by a standard DP.

In this application, sewer section releases or gate flows Q_i are approximated over time with orthogonal polynomials as follows:

$$Q_i(t) = Q_i^{(k)}(t) + \sum_{j=0}^{R} \alpha_{ij} T_j(t) \qquad \text{for} \quad t = \tau, \ldots, \tau + L$$

where

$Q_i^{(k)}$ = current known flow trajectory at some cycle k, sewer section i, (or DP stage i)

R = maximum order of polynomials selected, subject to $R \leq L$ (also called variable IRMAX)

$T_j(t)$ = orthogonal polynomial term of order $j = 0, \ldots, R$ for $R \leq L$

α_{ij} = coefficients of the orthogonal polynomial (which become state variables to be determined in the DP)

Perturbations are generated around the current trajectory $Q_i^{(k)}$ by the products of the coefficients and the terms of the orthogonal polynomials. Further detail can be found in Chapter III.

B.2 <u>Data Input</u> (NSOC example in Table 2)

```
READ(N15,55)IRMAX,M,N,MAXIT,QDELT,EPSI
55 FORMAT(4I10,F10.1,F10.2)
```

IRMAX = maximum order of polynomials, where IRMAX ≤ no. of lead time periods. Larger values of IRMAX increase computer execution time. A positive value for IRMAX gives 'forward' order of polynomials. A negative value of IRMAX gives 'reverse' orders of

Table 2. Staged DP Data and ODP Control Variables (Tape 15)

```
      -2         -3        +3         1        2.0      0.05  IRMAX,M,N,MAXIT,
 -100.      -100.      -100.      THIS IS THE VECTOR "QMIN"
   100.       100.      100.      THIS IS THE VECTOR "QMAX"
  1.          1.        1.        VECTOR OF WEIGHTING FACTORS FOR STAGES.
 +1.         -1.       10.        INITIAL TRAJECTORY FOR TIME 1
 +1.         -1.       10.         INITIAL TRAJECTORY FOR TIME NO. 2
 +1.         -1.       10.          INITIAL TRAJECTORY FOR TIME NO. 3
 +1.         -1.       10.           SAME FOR TIME STEP NUMBER 4.
       1.         1.         1.         1.         1.        1.
       0         00         00         00         00
99        THIS IS THE LIST CONTROL FOR STAGE 3 (FURTHEST DOWNSTREAM)        99
   5    8    +1    0
     1.98      154.7      0.013        3.0       22.7        1.19
      1.0        1.0        1.0        1.0
01     1        17.      14.71      2120.     18.265     -18.265     11.265      100.
      -1.
02     1        17.      14.71      2130.     18.275     -18.275     11.275      100.
      20.
03     1        17.      14.805     2140.     13.315     -18.285     11.285       90.
      60.
04     1        17.      14.805     2150.     18.295     -18.295     11.295      100.
      60.
05     1       17.       15.47     2500.    18.9685   -18.9685      11.97      100.
      60.
06     1       17.0      16.470    3000.0     19.970      -19.97    12.9700     100.0
      60.
07     1       17.0      17.470    3500.0     20.970      -20.97    13.9700     100.0
      60.
08     1       17.0      18.470    4000.0     21.970      -21.97    14.9700     100.0
      60.
09     1       17.0      19.470    4500.0     22.970      -22.97      15.97     100.0
      60.
       1.         1.         1.         1.         1.        1.
       0         00         00         00         00    NO LATERAL INFLOWS.
99       THIS IS THE LIST CONTROL FOR STAGE 2  (MIDDLE STAGE)               99
   5    2    +1    0                                  INCRE,NL,LPOWER,NGATES,ETC.
      3.3        0.0      0.013        3.0        1.7        1.19
      1.0        1.0        1.0        1.0   THESE.ARE THE WEIGHTING FACTORS.
01     1         9.      13.325    1530.0     16.825    -16.825      6.825  100.0000
      +1.                THIS IS THE INITIAL FLOW "QZERO" FOR SECTION  1
02     1        17.      14.3525     1942.    17.8525   -17.8525    7.8525       100.
 -1.
03     1       17.       14.78     2110.0     18.255     -18.255      8.255  100.0000
      -1.
       1.         1.         1.         1.         1.        1.         1.        1.
       0         00         00         00         00    NO LATERAL INFLOWS.
99       THIS IS THE LIST CONTROL FOR STAGE 1 (FURTHEST UPSTREAM)           99
   5    4    +1    0
     1.98        0.0      0.013        3.0        1.8        1.19  CKS,FLOW,RNI,DT,
      1.0        1.0        1.0        1.0   THESE ARE WEIGHTING FACTORS.
1      1         9.0        9.5        0.0        5.0     -13.00       10.0       1.00
     20.
2      1         9.0   9.525000        10.     13.025    -13.025      10.03     100.00
     20.
3      1         9.0     10.750       500.     14.250     -14.25      11.24       100.
     20.
4      1         9.0     12.000     1000.0     15.500     -15.50      12.50      100.0
     20.0
5      1         9.0     13.300     1520.0     16.800     -16.80      13.80      100.0
     20.0
```

polynomials as discussed in Chapter III.

M = number of discretizations for coefficients α, where M must be odd, and integer-valued (either 3, 5, 7, or 9). The coefficients are first discretized by the program into M possible optimal values between -1 and +1, separated by uniform intervals and including zero. The zero coefficient represents zero perturbation of the current trajectory, which is the basis for comparison. If no improvement can be found, the current trajectory (zero coefficient) is retained. The uniformly spaced coefficients are multiplied by the state variable discretization parameter ΔQ (discussed later) for flow under the gates, thus generating the desired trajectory perturbation. Larger values of M increase computer execution time. The sign of the variable M is also significant. A positive value of M indicates 'no trajectory update' for the initial gate flow trajectory. A negative value of M causes trajectory update after each measured storm input.

Example: (M = 5)

+1 - Discretization #1 + ΔQ
+0.5 - Discretization #2 + $(\Delta Q * 0.5)$
0 - Discretization #3 this is the current trajectory
-0.5 - Discretization #4 - $(\Delta Q * 0.5)$
-1 - Discretization #5 - ΔQ

N = number of Dynamic Programming stages (sewer lengths of tunnel system separated by gates) in series. It is best to start numbering stages at the farthest upstream point.

MAXIT = number of times QDELT (ΔQ) is discretized for a 'fine-tuned' solution. After each MAXIT, the state variable discretization parameter ΔQ is halved. Large values of MAXIT increase computer execution time.

QDELT = the state variable discretization parameter (ΔQ) for flow under the gates. This parameter discretizes the state variable Q by the amount ΔQ for a discrete solution state-space.

EPSI = A number of 'sweeps' through the orthogonal polynomials can be made, with the initial gate flow trajectory updated prior to each sweep or cycle. The DP algorithm is repeated for another cycle until insignificant improvement in the objective function is attained. This test for improvement of the objective function is accomplished by the variable EPSI. If the difference in objective function values at any cycle is not significant as compared to EPSI, the cycles will stop. No further trajectory update and sweep through the polynomials will provide significant objective function reduction at this level of discretization ΔQ. Suggested value for EPSI is 0.05.

KMAX = number of time periods considered by the package, calculated by the program as the forecast of lead time plus one (also may be described as the dimension of the DP problem). The need to regulate in-line storage efficiently is formulated as a multidimensonal dynamic programming problem. The dimension of the state and decision vectors is one greater than the storm forecasting lead time (LEAD + 1), or variable value KMAX. Thus, a problem with a storm lead forecast time of three control intervals results in a four dimensional DP problem with KMAX = 4. KMAX is *not* a part of formal data input, but appears in program output.

```
READ(N15,300)(QMIN(I),I=1,N)
READ(N15,300)(QMAX(I),I =1,N)
```

300 FORMAT(1X,F9.2,7F10.2)

QMIN(I) = gate flow (state variable) lower bound, per stage. QMIN gives the largest negative number the gate flow can achieve, or maximum negative outflow or reverse flow from stage I+1 into stage I.

QMAX(I) = gate flow (state variable) upper bound, per stage. QMAX sets the maximum positive value the gate flow can achieve, or maximum outflow from stage I into stage I+1.

READ(N15,300)(W(I),I=1,N)

W(I) = penalty term on exceeding the maximum available storage for stage I. This is the spatial weighting factor or penalty term associated with overflows resulting from the unsteady flow routing model, as imbedded in the DP. The DP is called over forecast lead time L with controlled hydrograph $\underline{Q}_i$ entering at the top of stage i.

READ(N15,300)(QINITL(I,K),I=1,N) for k=1, LEAD+1

QINITL(I,K) = the initial or starting 'guess' for outflow or flow through the gate from stage I of the sewer system at time period K. The initial flow guess can be optionally updated at each iteration of the DP algorithm. Suggested initial setting is 'gates almost closed,' or QINITL = ± 1.0 cfs depending on the direction of flow between stages as dictated by the initial water levels in the sewer.

READ(N15,378)(JFAT(I), I = 1,NL7) for NL7 $\leq$ N, the number of DP stages.

378 Format (10I8)

JFAT(I) = DP stage numbers (up to 10 maximum) which receive inflow from the forecasting model. The stage numbers should be entered in ascending order, e.g. 1, 2, 5, 8, etc.

C. SEWER SYSTEM DATA FOR STAGED DP MODEL AND PROTOTYPE SIMULATION

C.1 Overview

The staged DP formulation of the SWCP breaks the entire length of sewer into stages separated by controllable gates, or other control elements. For each resulting stage, a physical description of the sewer is input and the SWCP attempts to minimize overflows (maximize total storage) by allocating storage throughout the entire system. The unsteady flow routing procedure described in Chapter III is used to accurately route sewer flow within each stage.

The modeling of flow leaving the system at the overflow weirs is accomplished by a user-defined section of sewer with the section length equal to the effective width of the overflow pipe. A weir is assumed to be placed in the rectangular pipe section. Overflows occur when the water level in any sewer section exceeds the sill height of the overflow weir for that section. Overflows are computed by a broad-crested weir formula, which is linearized by the first two terms of a Taylor series approximation. Street flooding occurs at any sewer section when the head (water surface elevation) exceeds ground level elevation.

C.2 Data for Subroutine DYNAM (1)

```
READ(N15,50)(CLOSS(IU),IU=1,5),CLOSS7
50 FORMAT(1X,F9.2,7F10.2)
```

CLOSS(IU) = lateral inflow factor - lateral inflow rate equals the factor CLOSS(IU) multiplied by the upstream input hydrograph values for each time step.

---

(1) The following data are input for each stage of the staged DP beginning with stage N, the farthest downstream stage. Data for the prototype simulation (TAPE19) also follow the format for Subroutine DYNAM.

CLOSS7 = downstream boundary condition flow factor. CLOSS7 multiplied by the input hydrograph values at each time period equals the downstream discharge hydrograph. CLOSS7 can be negative if inflow occurs at the downstream end. This option can be selected if a pump station or a constant outflow condition or an outflow weir are not requested as the downstream boundary condition. Only one of these four options for the downstream boundary condition can be selected.

```
READ(N5,75)L1,L2,L3,L4,L5
75 FORMAT(1X,I9,4I10)
```

L1 - L5 = sewer tunnel section numbers (up to five maximum) where lateral inflows occur. These lateral inflows are multiples, by variable CLOSS, of the upstream inflow hydrograph, and are not the independent inflows (up to ten maximum) which are produced by the forecast model (discussed later).

```
READ(N5,125)LIST
125 FORMAT(I3)
```

LIST = this variable allows printout of all sewer section information at every time step for each DP iteration at any DP stage. It should be set to 1 for de-bug purposes only. Set equal to zero for just summary printout, or set LIST = 99 for just a bit more information than the summary printout.

C.3 Data for Subroutine SEWER

```
READ(N5,340)INCRE,NL,LPOWER,NGATES,IGATE(1),IGATE(2),IGATE(3),IGATE(4),
     IGATE(5),IGATE(6), IGATE(7), IGATE(8), IGATE(9)
340 FORMAT(16I5)
```

INCRE = number of increments to divide each time step. The total simulated time for each time step is INCRE multiplied by the variable DT (on next card). This allows the routing model to have a smaller time step than the forecasting model.

NL = number of sewer section lengths minus one in this sewer (each sewer stage in the DP is handled separately, beginning with the last stage on the downstream end). EXAMPLE: For 16 sections, input a value of 15 for NL.

LPOWER = power to which the overflow criterion is raised, allowing smoothing of the objective function if LPOWER is chosen greater than one.

NGATES = number of gates in the sewer for the prototype simulation model. For the staged DP calculations on the first pass, NGATES is set to zero, and variables IGATE(1) to IGATE(5) are zero or blank. Maximum number of gates allowed in the current version is nine, but this could be increased by modifying the FORTRAN program.

IGATE(1) to IGATE(9) = sewer tunnel section number (up to 9 gates maximum) where gates are located. If no gates or if this is the first pass for the staged DP data, set to zero.

```
READ(N9,350)CKS,FLOW,RNI,DT,ZSILL,TOL
350 FORMAT(8F10.0)
```

CKS = discharge coefficient for overflow weir equation.

FLOW = a constant outflow condition (cfs) if pump stations are not requested at the end of each stage. If no constant outflow, set to zero or leave blank. Leave blank for the staged-DP data.

RNI = Manning's 'n' value for tunnel or pipe roughness.

DT = time increment (minutes) for flow routing simulation.

ZSILL = height of weir at downstream boundary if this is chosen in lieu of pump stations or fixed constant outflows. If no weir at downstream boundary and constant outflow is selected, set ZSILL to a large positive number to preclude weir flow. If ZSILL is set equal to zero, this activates the pump station option for the downstream boundary condition. Pump rating curve is specified in subroutine PUMP as a FORTRAN subroutine.

TOL = maximum allowable Froude number for routing simulation before the model undertakes automatic reduction of the time step. Usually between 1.1 and 1.2, since a minor amount of 'false' supercritical flow can be generated by linearization errors in the routing model.

```
READ(N5,350)(WFGAG(ITIME),ITIME = 1,LEAD)
350 FORMAT(8F10.0)
```

WFGAG(ITIME) = the vector of weighting factors in the objective function for the controllable gates over time. If certain sewer overflow times are to be more critical than others, WFGAG can be adjusted accordingly to influence the time distribution of overflows. This can be used to attempt to capture the 'first flush' from a storm sewer system.

```
READ(N5,450)(ITYPE(I),WIDTH(I),HEIT(I),XL(I),ZS(I),ZZERO(I),DZERO(I),WF(I),
     QZERO(I),I=1,NX)
450 FORMAT(8X,I2,7F10.4),/,F10.4)
```

ITYPE(I) = selector for either circular or rectangular tunnel types.

ITYPE = 0 is circle   ITYPE = 1 is rectangular

WIDTH(I) = width of rectangular tunnel section in feet or diameter in feet of circular tunnel section.

HEIT(I) = height of the rectangular tunnel section in feet with respect to sewer invert; equal to the diameter for a circular section.

XL(I) = range or horizontal distance location for the tunnel sections in feet. Zero is the location of the furthest UPSTREAM section.

ZS(I) = weir sill height in feet for overflow points in the tunnel system. If there is no weir, it is set equal to the difference between ground surface elevation and sewer invert. Any flow depth or head in the sewer greater than ZS will cause an overflow or street flooding if at ground surface.

ZZERO(I) = invert elevation. A negative number implies that ground surface elevation is taken as the zero datum.

DZERO(I) = initial depth (feet) in each tunnel section with respect to channel invert; must be $\geq$ Manning's normal depth to avoid supercritical flow.

WF(I) = weighting factors for objective function for overflows (spatially) at each section (this might be set to 100 if an overflow is critical at any given section; or as low as one if not critical or to encourage overflow at a particular section).

QZERO(I) = initial discharge (cfs) in each tunnel section at the beginning of the simulation. Should be nonzero, and should be a flow $\leq$ critical discharge as calculated by Manning's equation. A negative value for QZERO indicates reverse flow at beginning of simulation. These values appear on the next card-image following WF(I).

D. PROTOTYPE UNSTEADY FLOW ROUTING DATA (2)

The data requirements between TAPE 15 and TAPE 19 differ only by:

1. TAPE 19 data are not separated into stages. Since we are now modeling the system as a whole, only one set of the variables (NL,LPOWER,NGATES,CKS,FLOW,RNI,DT,etc.) is required.
2. variable LIST may be set to '1' in TAPE 19. If you recall, LIST was set to '1' at any stage in TAPE 15 only as a debug device or to view all important calculations of the staged DP.

E. FORECAST DATA (SUBROUTINE FORCST)

The model used for forecasting in this study is the autoregressive-transfer model described in Chapter III. Historical or base line parameters of the forecast model are computed off line and written to Tape 12 (NTAPE) for access and possible updating by the SWCP if the parameter update option is selected. The following Section F shows how the data for parameter identification are input to Program PARAM, which then writes to TAPE12. This section assumes that TAPE12 is already constructed. See Table 4 for the NSOC example of the listing for TAPE12.

The following variables are read into the SWCP from the historical data disk file called NTAPE = TAPE12. These are the values computed by Program PARAM in the off-line historical analysis of storm data.

```
READ(NTAPE,220) NG,NT,NL,IOR,IL,LIST,(((A(LI,LN),P(LI,LJ,LN),LI=1,IL),
     LJ=1,IL),LN=1,NL
220 FORMAT(6I2,8X,3G20.14,/(4G20.14))
```

NG = number of historical storm events which are used to generate

---

(2) We simply repeat Section C.2, but this time for the entire system as one stage; written to TAPE19; see Table 3 for the NSOC example.

Table 3. Prototype Simulation Data for Unsteady Routing (Tape 19)

```
/EOR  SEPARATES THE DATA   NOW FOR THE FULL BLOWN SIMULATION...
     1.692     0.000     0.000      0.00      0.00     -.000 CLOSS(IU),CLOSS7
         9        00        00        00        00 ONE LATERAL INFLOW..
  1            THIS IS THE LIST CONTROL...LIST=1 FOR FULL PRINTOUT.           1
    5   15   +1    2    5    8                 INCRE,NL,LPOWER,NGATES,ETC.
      1.98     10.00     .0150       3.0     111.7      1.19 CKS,FLOW,RNI,DT,
        1.        1.   WEIGHTING FACTORS IN TIME  (FOR ONE TIME STEP ONLY)
 01      1       9.0       9.5       0.0       5.0    -13.00      10.0      1.00
       20.    INITIAL FLOW "QZERO" FOR SECTION 1 (FURTHEST UPSTREAM).
 02      1       9.0  9.525000       10.    13.025   -13.025     10.03    100.00
       20.
 03      1       9.0    10.750      500.    14.25     -14.25     11.24      100.
       20.
 04      1      9.     12.        1000.     15.50     -15.50     12.50     100.
       20.
 05      1      9.     13.3       1520.     16.80     -16.80     13.80     100.
        1.
 06      1      9.     13.325     1530.     16.825   -16.825     6.825     100.
       +1.
 8       1      17.    14.3525     1942.   17.8525  -17.8525    7.8525      100.
  -1.
 07      1       17.     14.78     2110.    18.255   -18.255     8.255      100.
       -1.
 08      1       17.     14.71     2120.    18.265   -18.265    11.265      100.
       -1.
 09      1       17.     14.71     2130.    18.275   -18.275    11.275      100.
       20.
 10      1       17.    14.805     2140.    13.315   -18.285    11.285       90.
       60.
 11      1       17.    14.805     2150.    18.295   -18.295    11.295      100.
       60.
 12      1      17.     15.47     2500.    18.9685  -18.9685     11.97     100.
       60.
 13      1      17.     16.47     3000.      19.97    -19.97     12.97     100.
       60.
 14      1      17.     17.47     3500.     20.97     -20.97     13.97     100.
       60.
 15      1      17.     18.47     4000.     21.97     -21.97     14.97     100.
       60.
 16      1      17.     19.47     4500.     22.97     -22.97      15.97    100.
       60.
```

Table 4. Historical or Base-Line Forecast Parameters

```
/EOR   NOW FOR THE OFF LINE FORECAST DATA....
 5 4 2 1 8 1      -24.962347085151   346484.57896246     14.861140652196
-205535.47017944   20.481482882927   370292.91892880    -12.428771651326
-216577.02881729   121.14667659193   7281.5866184033    -71.168011599913
-7333.5435188104   73.351287167961  -2.5020451048362    -43.351408187216
 2.5020498950780  -24.962347085151  -205535.47017944     14.861140652196
 121924.12626058   20.481482882927  -219658.54091531    -12.428771651326
 128473.94951431   121.14667659193  -4319.3388963597    -71.168011599913
 4350.2079195044   73.351287167961   1.4865344352796    -43.351408187216
-1.4865372812967  -24.962347085151   370292.91892880     14.861140652196
-219658.54091531   20.481482882927   395962.45398208    -12.428771651326
-231592.61354298   121.14667659193   8026.1990378347    -71.168011599913
-7980.2732771740   73.351287167961   2.2129635897789    -43.351408187216
-2.2129678251683  -24.962347085151  -216577.02881729     14.861140652196
 128473.94951431   20.481482882927  -231592.61354298    -12.428771651326
 135455.13006633   121.14667659193  -4696.5523656328    -71.168011599913
 4668.7828036113   73.351287167961  -1.3380803054587    -43.351408187216
 1.3380828666422  -24.962347085151   7281.5866184033     14.861140652196
-4319.3388963597   20.481482882927   8026.1990378347    -12.428771651326
-4696.5523656328   121.14667659193   427.03367791626    -71.168011599913
-314.31933207422   73.351287167961   5.3705215281125    -43.351408187216
-5.3705318658296  -24.962347085151  -7333.5435188104     14.861140652196
 4350.2079195044   20.481482882927  -7980.2732771740    -12.428771651326
 4668.7828036113   121.14667659193  -314.31933207422    -71.168011599913
 248.88147699734   73.351287167961  -3.1173203874854    -43.351408187216
 3.1173263878147  -24.962347085151  -2.5020451048362     14.861140652196
 1.4865344352796   20.481482882927   2.2129635897789    -12.428771651326
-1.3380803054587   121.14667659193   5.3705215281125    -71.168011599913
-3.1173203874854   73.351287167961   3.0977480536524    -43.351408187216
-1.8752098417709  -24.962347085151   2.5020498950780     14.861140652196
-1.4865372812967   20.481482882927  -2.2129678251683    -12.428771651326
 1.3380828666422   121.14667659193  -5.3705318658296    -71.168011599913
 3.1173263878147   73.351287167961  -1.8752098417709    -43.351408187216
 1.1526696128695  -42.236291268121   346484.57896246     25.145049983544
-205535.47017944   34.654669037962   370292.91892880    -21.029481634075
-216577.02881729   204.98017679328   7281.5866184033    -120.41627562690
-7333.5435188104   124.11037788819  -2.5020451048362    -73.350582652766
 2.5020498950780  -42.236291268121  -205535.47017944     25.145049983544
 121924.12626058   34.654669037962  -219658.54091531    -21.029481634075
 128473.94951431   204.98017679328  -4319.3388963597    -120.41627562690
 4350.2079195044   124.11037788819   1.4865344352796    -73.350582652766
-1.4865372812967  -42.236291268121   370292.91892880     25.145049983544
-219658.54091531   34.654669037962   395962.45398208    -21.029481634075
-231592.61354298   204.98017679328   8026.1990378347    -120.41627562690
-7980.2732771740   124.11037788819   2.2129635897789    -73.350582652766
-2.2129678251683  -42.236291268121  -216577.02881729     25.145049983544
 128473.94951431   34.654669037962  -231592.61354298    -21.029481634075
 135455.13006633   204.98017679328  -4696.5523656328    -120.41627562690
 4668.7828036113   124.11037788819  -1.3380803054587    -73.350582652766
 1.3380828666422  -42.236291268121   7281.5866184033     25.145049983544
-4319.3388963597   34.654669037962   8026.1990378347    -21.029481634075
-4696.5523656328   204.98017679328   427.03367791626    -120.41627562690
-314.31933207422   124.11037788819   5.3705215281125    -73.350582652766
-5.3705318658296  -42.236291268121  -7333.5435188104     25.145049983544
 4350.2079195044   34.654669037962  -7980.2732771740    -21.029481634075
 4668.7828036113   204.98017679328  -314.31933207422    -120.41627562690
 248.88147699734   124.11037788819  -3.1173203874854    -73.350582652766
 3.1173263878147  -42.236291268121  -2.5020451048362     25.145049983544
 1.4865344352796   34.654669037962   2.2129635897789    -21.029481634075
-1.3380803054587   204.98017679328   5.3705215281125    -120.41627562690
-3.1173203874854   124.11037788819   3.0977480536524    -73.350582652766
-1.8752098417709  -42.236291268121   2.5020498950780     25.145049983544
-1.4865372812967   34.654669037962  -2.2129678251683    -21.029481634075
 1.3380828666422   204.98017679328  -5.3705318658296    -120.41627562690
 3.1173263878147   124.11037788819  -1.8752098417709    -73.350582652766
 1.1526696128695
```

stochastic base line parameters for the forecasting model.

NT = desired autoregressive order P, or the number of initial time periods that must pass before a forecast can be initiated.

NL = number of locations used in the forecast analysis (number of independent sewer inflow points); currently dimensioned for a maximum of ten inflows.

IOR = model power or parameter n in equation 9 of Chapter III (can vary in integers from 1 to 4; i.e., IOR = 1 represents a linear model).

IL = NL*NT*IOR which used for problem dimensioning.

LIST = 1, for writing results from forecast analysis, as previously used off-line; otherwise, LIST = 0.

A(LI,LN) = the baseline model parameters parameters (LI=1,...,IL) identified by Program PARAM for each location LN=1,...,NL).

P(LI,LJ,LN) = error covariance matrix produced by Program PARAM using equation 19 of Chapter III.

The following variables can be input to the SWCP program in an 'interactive' or 'time sharing' mode, or batch mode if desired, with the remaining data retrieved from disk files (* denotes 'free' format). (3)

```
READ(N5,*)IUD
READ(N5,*)LIST
READ(N5,*)LEAD
```

IUD = variable to either update or not update base-line parameters in real-time. Set IUD=1 to update parameters, otherwise set IUD=0.

LIST = same as previous list, set LIST=1 to get detailed Forecast Output.

---

(3) See Table 5 for example of a CDC Cyber 825 procedure file to exercise this interactive option for the model (this is the NSOC case, illustrating the operation of the SWCP for actual inflows less than what were expected by the forecast model

Table 5. Interactive data input.

```
*
*  CALL DENNIS MORROW--491-7243--FOR ASSISTANCE.
*
RTF.
GET,LGO=LGOSWCP,INPXX=SWCPTAP.  UN=??
COPYBR,INPXX,TAPE15.   3-STAGE DP INPXX TAPE (STAGES BROKEN AT GATES).
COPYBR,INPXX,TAPE19.   SIMULATION ROUTING MODEL INPXX DATA TAPE
COPYBR,INPXX,TAPE12.   REAL STORM PARAMETERS ARE INPXX FROM TAPE 12.
REWIND,TAPE12,TAPE15,TAPE19.
LDSET,PRESET=NGINF.MAP=BS.
LGO,INPUT,OUTPXX,INPUT,OUTPXX,TAPE12,TAPE15,TAPE19,TAPE16,*PL=50000.
* TAPE 16 IS FOR OUTPUT FROM THE DP.
RETURN,LGO,TAPE12,TAPE15,TAPE19,INPXX.
RWF,OUTPUT.
*   TAPE16 IS A LOCAL FILE CONTAINING SWCP DETAILED OUTPUT.
*  ROUTE THIS FILE TO THE PRINTER OF YOUR CHOICE.
DAYFILE,OUTPUT.
*
EXIT.  WE SKIP TO HERE IN CASE OF "DUMP".
DAYFILE,OUTPUT.
*
```

Table 5. (continued).

```
REAL STORM, PARAMETERS ARE INPUT FROM TAPE 12

THE MODEL BASE LINE PARAMETERS ARE BASED UPON    5  STORM SEQUENCES
EACH FORECAST WILL BE BASED UPON THE PREVIOUS 4 TIME PERIODS(MODEL ORDER)
THE NUMBER OF LOCATIONS CONSIDERED IS  2
THE MODEL POWER IS  1

ENTER "1" TO UPDATE PARAMETERS, "0" FOR NO UPDATE.
    1
UPDATE PARAMETERS AFTER EACH REAL TIME PERIOD

ENTER "1" FOR FULL PRINTOUT, "0" FOR PARTIAL PRINT
    1
LIST CONTROL IS  1

ENTER LEAD TIME OR "0" FOR REACTIVE MODE
    4
LEAD TIME IS  4

BASE FLOW IGNORED

SIMULATE THE PASSAGE OF A REAL STORM


          MAXIMUM VARIABLE DIMENSIONS .GE.    8
          STORM LENGTH MUST BE .GE.     5  TIME PERIODS.
ENTER NUMBER OF TIME PERIODS THIS STORM LASTS.
    9
ENTER WEIGHTING FACTOR FOR THIS STORM.
      1.0
ENTER STORM ID WHICH CAN BE ANY POSITIVE NUMBER
 8451
ENTER STORM DATE AS INTEGER MONTH, DAY,  AND YEAR
    9   10   84
STORM DATE  9 10    84  ID     8451
THIS STORM LASTED  9 TIME PERIODS
AND HAS A WEIGHTING FACTOR OF      1.000
ENTER STORM HYDROGRAPH FOR  2 LOCATIONS.
 0.          0.
ENTER STORM HYDROGRAPH FOR  2 LOCATIONS.
 0.          0.
ENTER STORM HYDROGRAPH FOR  2 LOCATIONS.
 0.          0.
ENTER STORM HYDROGRAPH FOR  2 LOCATIONS.
 0.          0.
THE FOLLOWING SEQUENCE CONSISTS OF   4 MEASURED VALUES
THE BALANCE HAS BEEN FORECASTED FROM THESE VALUES
                                            LOCATION
                          1         2
  TIME
    1              0.         0.
    2              0.         0.
    3              0.         0.
    4              0.         0.
                   -FORECAST--FORECAST--
    5              0.         0.
```

Table 5. (continued).

```
    6              0.         0.
    7              0.         0.
    8              0.         0.
ENTER STORM HYDROGRAPH FOR  2 LOCATIONS.
  1.0000      1.0000
THE FOLLOWING SEQUENCE CONSISTS OF   5 MEASURED VALUES
THE BALANCE HAS BEEN FORECASTED FROM THESE VALUES
                                            LOCATION
                          1         2
  TIME
    1              0.         0.
    2              0.         0.
    3              0.         0.
    4              0.         0.
    5               1.0000     1.0000
                   -FORECAST--FORECAST--
    6               30.000     50.760
    7               50.000     84.600
    8               30.000     50.760
    9               9.9999     16.920
ENTER STORM HYDROGRAPH FOR  2 LOCATIONS.
  24.000      40.610
LEAD TIME IS  3

THE FOLLOWING SEQUENCE CONSISTS OF   6 MEASURED VALUES
THE BALANCE HAS BEEN FORECASTED FROM THESE VALUES
                                            LOCATION
                          1         2
  TIME
    1              0.         0.
    2              0.         0.
    3              0.         0.
    4              0.         0.
    5               1.0000     1.0000
    6               24.000     40.610
                   -FORECAST--FORECAST--
    7               49.915     84.456
    8               25.475     43.104
    9               13.198     22.331
    OBJECTIVE VALUE (INCLUDING PENALTIES) 48.7430389
RES. NO. 2     GATE         .0273         .0326         .0403
RES. NO. 3     GATE        -.0170        -.0156        -.0149
    OBJECTIVE VALUE (INCLUDING PENALTIES) 47.6228435
RES. NO. 2     GATE         .0103         .0228         .0466
RES. NO. 3     GATE        -.0064        -.0110        -.0178
ENTER STORM HYDROGRAPH FOR  2 LOCATIONS.
  40.000      67.680
 TRANSFER 2 GATE TRAJS FOR SIMULATION MODEL.
     .010        .023
 TRANSFER 2 GATE TRAJS FOR SIMULATION MODEL.
      .006      .011

SEC.   DEPTHS         FLOWS     ... FROM THE SIMULATION MODEL.
 1      6.542         40.00
 2      6.580         1.383
```

Table 5. (continued).

```
 3      7.801          1.667
 4      9.049          1.919
 5      10.35          2.162
 6      7.150          2.162
 7      8.177          -.6910
 8      8.580         -2.213
 9      12.65         -2.213
10      12.66          62.48
11      12.67          62.20
12      12.68          61.92
13      13.35          52.10
14      14.35          38.07
15      15.35          24.04
16      16.35          10.00
    OBJECTIVE VALUE (INCLUDING PENALTIES) 4.63089679
LEAD TIME IS  2

THE FOLLOWING SEQUENCE CONSISTS OF   7 MEASURED VALUES
THE BALANCE HAS BEEN FORECASTED FROM THESE VALUES
                                                 LOCATION
                         1          2
  TIME
     1              0.         0.
     2              0.         0.
     3              0.         0.
     4              0.         0.
     5               1.0000     1.0000
     6               24.000     40.610
     7               40.000     67.680
                    -FORECAST--FORECAST--
     8               23.416     39.619
     9               7.9348     13.426
TRANSFER   9 INITIAL D.+ Q. FOR THE DP STAGE 3
    8.580    12.649    12.656    12.666    12.676    13.349    14.349
   -2.213    -2.213    62.479    62.199    61.919    52.101    38.071
TRANSFER   3 INITIAL D.+ Q. FOR THE DP STAGE 2
   10.348     7.150     8.177
    2.162     2.162     -.691
TRANSFER   5 INITIAL D.+ Q. FOR THE DP STAGE 1
    6.542     6.580     7.801     9.049    10.348
   40.000     1.383     1.667     1.919     2.162
    OBJECTIVE VALUE (INCLUDING PENALTIES)0.
RES. NO. 2     GATE          .0253          .0540          .2080
RES. NO. 3     GATE         -.0129         -.0211         -.0313
    OBJECTIVE VALUE (INCLUDING PENALTIES)0.
RES. NO. 2     GATE          .0253          .0540          .2080
RES. NO. 3     GATE         -.0129         -.0211         -.0313
ENTER STORM HYDROGRAPH FOR  2 LOCATIONS.
  24.000      40.610
 TRANSFER 2 GATE TRAJS FOR SIMULATION MODEL.
      .025      .054
 TRANSFER 2 GATE TRAJS FOR SIMULATION MODEL.
      .013      .021
```

Table 5. (continued).

```
SEC.   DEPTHS         FLOWS     ... FROM THE SIMULATION MODEL.
 1      6.201         24.00
 2      6.233        -2.498
 3      7.459        -.3938
 4      8.709         1.757
 5      10.01         3.994
 6      8.143         3.994
 7      9.170        -1.571
 8      9.573        -4.539
 9      13.95        -4.539
10      13.96         38.68
11      13.97         38.58
12      13.98         27.90
13      14.66         24.50
14      15.66         19.66
15      16.66         14.82
16      17.66         10.00
    OBJECTIVE VALUE (INCLUDING PENALTIES) 194.581877
LEAD TIME IS  1

THE FOLLOWING SEQUENCE CONSISTS OF   8 MEASURED VALUES
THE BALANCE HAS BEEN FORECASTED FROM THESE VALUES
                                            LOCATION
                          1         2
  TIME
    1              0.         0.
    2              0.         0.
    3              0.         0.
    4              0.         0.
    5               1.0000     1.0000
    6               24.000     40.610
    7               40.000     67.680
    8               24.000     40.610
                   -FORECAST--FORECAST--
    9               8.9205     15.095
TRANSFER   9 INITIAL D.+ Q. FOR THE DP STAGE 3
     9.573    13.954    13.963    13.973    13.983    14.659    15.662
    -4.539    -4.539    38.679    38.582    27.900    24.499    19.655
TRANSFER   3 INITIAL D.+ Q. FOR THE DP STAGE 2
    10.009     8.143     9.170
     3.994     3.994    -1.571
TRANSFER   5 INITIAL D.+ Q. FOR THE DP STAGE 1
     6.201     6.233     7.459     8.709    10.009
    24.000    -2.498     -.394     1.757     3.994
    OBJECTIVE VALUE (INCLUDING PENALTIES) 31.1159905
RES. NO. 2     GATE         .0427         .0703
RES. NO. 3     GATE        -.0156        -.0167
    OBJECTIVE VALUE (INCLUDING PENALTIES) 20.6517322
RES. NO. 2     GATE         .0516         .4985
RES. NO. 3     GATE        -.0188        -.0393
    OBJECTIVE VALUE (INCLUDING PENALTIES) 11.6200015
RES. NO. 2     GATE         .0024         .2893
RES. NO. 3     GATE        -.0217        -.0604
    OBJECTIVE VALUE (INCLUDING PENALTIES) 11.2961756
RES. NO. 2     GATE         .0222         .1920
```

Table 5. (continued).

```
RES. NO. 3     GATE          -.0143         -.0687
ENTER STORM HYDROGRAPH FOR  2 LOCATIONS.
  8.0000       13.540
 TRANSFER 2 GATE TRAJS FOR SIMULATION MODEL.
      .022       .192
 TRANSFER 2 GATE TRAJS FOR SIMULATION MODEL.
      .014       .069

SEC.    DEPTHS          FLOWS      ... FROM THE SIMULATION MODEL.
 1       5.762          8.000
 2       5.789         -5.517
 3       7.020         -5.051
 4       8.277         -4.723
 5       9.512         -3.621
 6       9.659         -3.621
 7       10.69         -5.806
 8       11.09         -9.963
 9       13.94         -9.963
10       13.95          6.354
11       13.96          6.429
12       13.97         -3.850
13       14.65         -1.237
14       15.65          2.504
15       16.65          6.250
16       17.65          10.00
    OBJECTIVE VALUE (INCLUDING PENALTIES) 187.358716
LEAD TIME IS  0

THE FOLLOWING STORM       * WAS *       THE REAL EVENT
WHICH   * WAS * CONSIDERED SEQUENTIALLY AND FORECASTED BY THE MODEL
                                             LOCATION
                          1          2
  TIME
    1                   0.00       0.00
    2                   0.00       0.00
    3                   0.00       0.00
    4                   0.00       0.00
    5                   1.00       1.00
    6                  24.00      40.61
    7                  40.00      67.68
    8                  24.00      40.61
    9                   8.00      13.54
```

LEAD = number of time steps ahead the forecast model will predict. Maximum dimensions of current version of the SWCP are for LEAD ≤ 3. If LEAD=0, this gives the reactive mode of operation with no forecasting.

```
READ(N5,*)IIT
READ(N5,*)WF
READ(N5,*)ID
READ(N5,*)(NAME(I),I=1,3)
```

IIT = number of time periods that the real storm hydrograph lasts.

WF = weighting factor for various storm events. The user may wish to weight some storms higher than others. Storms which are weighted higher have more of an influence on the modification of the historical baseline parameters if the update option is selected.

ID = identification number for a given storm event, if desired it can be any positive integer number.

NAME = date of input storm event, can be any three integer numbers. Example: 7, 12, 1984.

```
READ(N5,*)(R(INL,IT),INL=1,NL)
```

R(INL,IT) = storm hydrograph input value at location INL, time IT.

F. DATA FOR BASE-LINE PARAMETER IDENTIFICATION (Program PARAM)

Each input card for the base-line historical identification is described in detail below. Variable locations on each card are shown by field number. All cards are divided into eight fields of ten columns each. The different values a variable may assume and the conditions for each variable are discussed.

(Note: all floating point entries must include decimal points)

Field	Variable	Value and Description
1	NG*	Any value (limited only by FORTRAN program dimensions to a maximum of 20 time periods)· NG indicates the number of discrete storm events for which inflow data are provided for baseline identification use.
2	NT*	2 to 4. NT indicates the number of preceeding time periods considered immediately relevant in the forecast. NT is equivalent to the autoregressive order p as given in Chapter III. The model requires NT time periods to have passed before forecasting commences.
3	NL*	1 to 10 (greater if dimensioning is changed). NL indicates the number of forecast locations considered in the model. Flow forecasted at a location must be associated with a particular stage in the staged DP model.
4	IOR*	1 to 4. Indicates the maximum power considered in the model (If IOR = 1, the model is linear).
5	IUD*	Parameter update control = 1, if forecast parameters are to be updated with each increment of the real event, 0, otherwise.
6	IRS*	For base-line parameter identification, set IRS = 0.

---

*These are all integer variables and must be right justified in the field.

Storm Description Card**

Field	Variable	Value and Description
1	IIT*	1 to 20. IIT indicates the number of time periods in the following storm.
2	WF	Any value. WF is the relative weighting factor for the following storm.
3	ID*	Any value. ID is a storm identification number (ID must be less than 10 digits).
4-6	Storm Name	Any combination of letters and numbers used to describe the storm (Name must be less than 50 characters).

Storm Inflow Cards

Field	Variable	Value and Description
1→	R(-,-)**	Any value. R(-,-) are the rainfall or runoff values (in any consistent set of units) for a particular location and time. Fields are filled sequentially for each particular time period, starting in field 1, with data for all locations.

**An excessive number of zeros can produce instabilities in the Forecast Program.

APPENDIX C

OPTIMAL DESIGN OF URBAN STORM WATER DRAINAGE SYSTEMS

C-2

1981 International Symposium on Urban Hydrology, Hydraulics, and Sediment Control
(University of Kentucky, Lexington, Kentucky – July 27-30, 1981)

OPTIMAL DESIGN OF URBAN STORM WATER DRAINAGE SYSTEMS

David K. Robinson
School of Civil Engineering
University of New South Wales
P.O. Box 1
Kensington NSW, 2033 AUSTRALIA

and

John W. Labadie
Department of Civil Engineering
Colorado State University
Ft. Collins, CO 80523

Abstract. Several researchers have developed screening models for minimizing the cost of sizing and vertical alignment of storm sewer systems. One of the disadvantages of these models is that they are designed with specific physical/economic assumptions in mind, as well as specific types of cost functions. These assumptions are effectively *hard-wired* into the program, which militates against modifying the program to suit situations not covered in the original assumptions. A generalized dynamic programming computer code has been developed at Colorado State University called CSUDP. A library of subroutines is being prepared as adjuncts to the basic code for solving a wide range of civil engineering problems. One of the packages includes a set of subroutines for storm sewer design. The coding includes data management routines for convenience to the user not familiar with dynamic programming. The code can consider complex branching networks with up to three pipes entering a given manhole. Default options are available for excavation, pipe, and manhole cost functions. If the costs for a particular case have a significantly different structure, the user can supply his own subroutine. There are several other default options which make this program extremely convenient for *normal* design conditions, as well as usable for unique problems. Extensive comparisons have been made between available published results of other dynamic programming screening models and the CSUDP package. Results are extremely close; differing in total cost by 0.8% to In most cases, results are extremely close. The one comparison resulting in a large deviation was due to slightly differing hydraulic and cost assumptions, which caused larger commercial pipe sizes to be chosen by CSUDP, even though actual pipe requirements deviated little.

Introduction

Throughout the world, many millions of dollars are invested annually in the construction of urban stormwater drainage systems. These projects range from the augmentation of an existing system in order to drain a new subdivision, to the development of a new drainage system to service new communities. The numerous small-scale projects, which actually account for the bulk of the total investment in urban drainage, are often designed by hand through use of the Rational Method. The larger systems are frequently designed with the use of a standard computer package such as the Road Research Laboratory Method. After reviewing the literature, and after numerous discussions with engineering consultants, the authors are convinced that there is currently little use made of optimization techniques to reduce the construction and operating costs of stormwater drainage systems.

The aim of the research reported herein has been to develop a basic computer package suitable for optimizing the design of drainage systems. Particular attention has been given to designing the package to be implementable on small computer systems typically available in engineering consulting offices. This package may also be readily installed on larger computer systems and can be expanded to handle large drainage networks.

This package is built upon a generalized dynamic programming computer code called CSUDP which has been developed at Colorado State University by the second author. An attempt has been made in CSUDP to develop a comprehensive code, without greatly impairing efficiency. The code is capable of solving discrete, finite stage problems of up to five dimensions in the state vector. One dimensional problems are solved by the standard dynamic programming procedure, whereas multidimensional problems are solved by an incremental dynamic programming technique. In addition, a *splicing* option is available for starting with a coarse grid of state variable increments and progressively refining the solution. Both deterministic and stochastic dynamic programming problems can be solved, including both independent and Markovian-type problems for the latter. The user supplies two subroutines: STATE (which defines the transformation of the state vector within a given stage) and OBJECT (which defines the costs, benefits, or other objectives associated with the state transformation and corresponding decisions).

It has long been assumed in the operations research field that dynamic programming (hereafter abbreviated as DP) is so problem specific, it is not possible to write an efficient, generalized DP code. This has left researchers and practitioners with the task of writing new DP computer codes for each new problem to be solved. This has discouraged practical usage of DP since the code must be written by someone with both a firm understanding of how dynamic programming works and the ability to write efficient code. This combination of expertise is rarely available, particularly among practicing engineers. A balance has been reached with development of CSUDP. The user can supply the peculiarities of his particular problem in Subroutines STATE and OBJECT without having to in any way alter the MAIN computer routine. It is admitted that a code written specifically for a particular problem can probably be made more efficient than an application of CSUDP to that problem. However, code developmental costs will likely be much greater for the latter case. Comparisons have been made between CSUDP and problem-specific codes. CSUDP has been found to be competitive because the writer of the problem-specific code is often in a great hurry to solve his problem and therefore pays insufficient attention to coding techniques that will minimize execution time on the computer.

As a further attempt to place dynamic programming in the hands of practitioners as a practical, workable tool, a package of problem-specific subroutines are being prepared to solve a number of problems in civil engineering. For these specific cases, the user need not code his own Subroutines STATE and OBJECT. Also, the data input and output are designed with the specific problem in mind. The user need only supply appropriate data specific to his problem. This paper describes a package called CSUDP-SEWER for optimal design of a drainage network.

A brief literature review in the area of optimal storm sewer design is given, followed by a summary of the CSUDP code and its available options. The assumptions and data requirements for the CSUDP-SEWER package is then presented showing the pipe, manhole and junction calculation scheme. A simple example serves to illustrate the methodology, followed by more in-depth comparative analyses with other published results.

Contemporary Developments

Since the mid-1960's, a number of researchers have developed various techniques for optimizing the vertical alignment of sewer systems, whether they be sanitary sewers or urban drainage systems. These techniques have used various optimizing techniques such as linear programming (Deininger 1970) and nonlinear separable programming (Holland 1966).[1,2] The most popular approach, however, has been dynamic programming, including the work of Meridith (1971), Merritt and Bogan (1973), Froise, et al. (1975), Yen, et al. (1976), and Froise and Burges (1978).[3,4,5,6,7] Meridith's work was in essence developmental and only extended to a nonbranching system. The cost functions he proposed, however, have been extensively used by other researchers such as Yen, et al. (1976).

The work of Merritt and Bogan extended the use of a DP model to include branching systems and minimum and maximum expected flows. This model appears to be particularly suited to the design of sanitary sewers. The work of Froise, et al. (1975) and Yen, et al. (1976) has been directed towards the development of large comprehensive models for finding least-cost solutions that can use input hydrographs at each manhole and then route them through the drainage system. These models are possibly too comprehensive for solving large-scale problems, but can be valuable for refining solutions obtained by a screening model. Labadie, et al. (1980) combined dynamic programming with a fully dynamic hydraulic routing model, but the focus of this work was on real-time stormwater control problems rather than sewer design.[8] Recent ongoing studies at the University of New South Wales, Australia (Howell, 1981) indicate that the larger programs using an incremental state dynamic programming approach with sewer routing require five to ten times more computer time than the standard, nonincremental dynamic programming approaches where only constant discharges are required for the design.[9] The approach proposed in this paper employs the latter methodology.

CSUDP-A Brief Description

Program CSUDP has been developed to solve a wide variety of dynamic programming problems. The objective function can be: (1) additive-type (i.e., over each stage), (2) multiplicative, or (3) min-max (or max-min). A backwards solution approach is adopted but the program can be *fooled* into solving a forward DP problem by appropriately redefining the state transformation function. For the additive case, it solves the following problem:

$$\max \text{ or } \min \sum_{i=1}^{N} f_i(\underline{x}_i, \underline{u}_i, \underline{x}_{i+1}) \qquad (1)$$

subject to:

$$\left.\begin{array}{l} \underline{x}_{i+1} = \underline{g}_i(\underline{x}_i, \underline{u}_i) \; \varepsilon \; X_{i+1} \qquad (2) \\ \underline{u}_i \; \varepsilon \; U_i \qquad (3) \end{array}\right\} i = 1, \ldots, N$$

$$x_1 \; \varepsilon \; X_1$$

where there are N *stages* and the bounded sets X_{i+1}, U_i represent discrete values the *state* and *decision variables*, respectively, can take on. Equation (2) is called the state transformation equation, or system dynamics. The functions $f_i(\cdot)$ and $\underline{g}_i(\cdot)$ can be of any form, including analytical or tabulated functions. Additional constraints of the form

$$\underline{h}_i(\underline{x}_i, \underline{u}_i) \leq \underline{0} \qquad (4)$$

can be indirectly handled via a penalty term approach.

The vectors $\underline{x}_i$ and $\underline{u}_i$ may have up to five dimensions. The general DP *recursion equation* is

$$F_i(\underline{x}_i) = \underset{\underline{u}_i \in \mathcal{U}_i}{\overset{\text{max or}}{\min}} [f_i(\underline{x}_i, \underline{u}_i, \underline{x}_{i+1}) + F_{i+1}(\underline{x}_{i+1})] \tag{5}$$

subject to equation (2), for $i=1,\ldots,N$, where $F_{N+1}(\cdot) \underline{\underline{\Delta}} 0$. Often, it is more convenient to solve an *invertable* form for the problem:

$$F_i(\underline{x}_i) = \underset{\underline{x}_{i+1} \in \mathcal{X}_{i+1}}{\overset{\text{max or}}{\min}} [f_i(\underline{x}_i, \underline{u}_i, \underline{x}_{i+1}) + F_{i+1}(\underline{x}_{i+1})] \tag{6}$$

where an invertible form of the state equation is used (if such a form exists):

$$\underline{u}_i = \underline{g}_i^{-1}(\underline{x}_i, \underline{x}_{i+1}) \tag{7}$$

This form avoids the problem of having to interpolate the optimal value function $F_{i+1}(\underline{x}_{i+1})$ as in the noninvertible form (equation (2)) when an $\underline{x}_{i+1}$ vector is computed for which $F_{i+1}(\cdot)$ values were not previously computed during stage i. If the cost function $f_i(\cdot)$ uses tabulated data with respect to $\underline{u}_i$, then this function may need to be interpolated in the invertible form. Actually, for Program CSUDP, the noninvertible form (equation (5)) can only be used for one-dimensional (i.e., in the state variable x_i) problems. Multidimensional problems are solved via a state incremental dynamic programming approach (Larson, 1969).[10]

The program divides the allowable state-space $\mathcal{X}_i$ into a finite number of increments, with the incremental spacing specified by the user. The optimal value function (equations (5) or (6)) is then recursively evaluated for all feasible combinations of state increments, starting with stage N and working backwards. For multidimensional problems, the discrete combinations are severely limited to a predefined *corridor*. At each stage, the optimal decisions $\underline{u}_i^*(\underline{x}_i)$ and end-of-period states $\underline{x}_{i+1}^*(\underline{x}_i)$ are stored for all feasible, discrete $\underline{x}_i$ for later use. When the first stage is reached, a *trackback* through the stages selects the specific optimal decisions $\underline{u}_i^*$ for each stage from among the stored *families* of optimal policies $\underline{u}_i^*(\underline{x}_i)$. For stochastic problems, the latter are printed out because specific optimal policies cannot be found due to the stochastic nature of the state vector.

The CSUDP code has a variety of print options and will find families of optimal policies for a range of initial state conditions contained in the set X_1. The code has a variety of other capabilities, such as a *tie-breaking* procedure associated with the maximizing (or minimizing), operation in equations (5) or (6). For stochastic problems, bounds on probability of failure to satisfy constraints may be input in order to facilitate risk analysis.

In developing a code such as this, there are ultimately difficult decisions with respect to anticipating the type of computer facilities available to the user. The current version of the code stores $F_i(\underline{x}_i)$ in fast core memory and the $\underline{u}_i^*(\underline{x}_i)$ and $\underline{x}_{i+1}(\underline{x}_i)$ in random access files. This saves storage, but increases computer time.

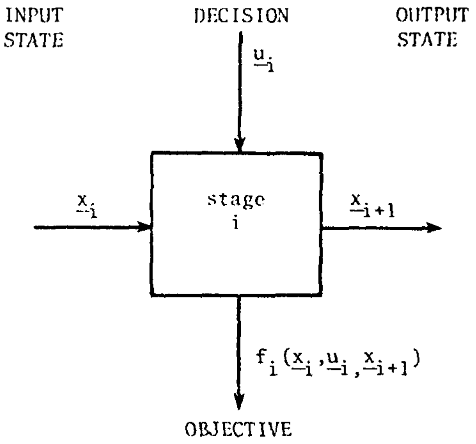


Figure 1. Typical stage for a sequential decision process solvable by Program CSUDP.

The CSUDP-SEWER Package

Introduction

A suite of programs have been developed in conjunction with Program CSUDP which optimize the vertical alignment of a stormwater collection system. The program is currently dimensioned to analyze a system where the sum of the number of manholes and the number of branch lines is less than 50. Larger systems may be analyzed if the array dimensions in Program CSUDP are increased. The program allows up to three pipe (or upstream manholes) to drain into any manhole, which in turn is drained by one pipe.

Problems such as urban drainage design require a program that will solve branching networks. Though CSUDP is designed to solve serial problems, the branched problem can be represented by a set of separate serial problems for each branch which are then linked together at the branching manholes or nodes. This was readily accomplished with a few minor modifications to the supporting subroutines of CSUDP. The formulation of the data set required by CSUDP for any reasonable sized problem is somewhat demanding, especially if the user is not familiar with the mechanics of the program. The problem of data input was greatly simplified by developing an auxiliary program which transforms a concise description of the problem into the specific data file required by CSUDP. This data management program (DATAGN) automatically partitions the branched problem into the required set of pseudo-serial problems. The program also employs, where applicable, global data descriptions such as minimum and maximum cover requirements to eliminate the repetitive data entries, but with the flexibility to override these global instructions when local conditions dictate.

The sewer design problem is related to the previously defined general format for CSUDP in the following way:

1. In the pseudo-serial formulation, each pipe section between manholes is represented as *a stage*. In addition, each junction manhole is represented by one stage for *each* incoming pipe. For example, if two pipes are entering a manhole, there are *two* stages associated with that manhole.
2. The one-dimensional state variable x_i represents pipe crown elevation.

The notational format is to represent x_{i+1} as the upstream crown elevation and x_i as the downstream crown elevation.

3. The decision variable u_i is pipe diameter for pipe section i.

4. The cost function $f_i(x_i,u_i,x_{i+1})$ represents all pipe costs, including excavation; or, if stage i is a manhole, all manhole costs.

5. The state transformation equation (equation (2)) is essentially Manning's equation if stage i is a pipe section. The *invertible form* is used:

$$u_i = \left[\frac{2.16nQ_i}{\sqrt{(x_{i+1}-x_i)/L_i}}\right]^{0.375} \qquad (8)$$

where $(x_{i+1}-x_i)/L_i$ is the slope of pipe section i, Q_i is the rate of flow in pipe section i, and n is Manning's roughness coefficient. Though equation (8) represents pipe-full flow, the program will also consider variable depth and n values.

6. The sets X_i and U_i represent upper and lower bounds on crown elevation and pipe diameter, respectively.

7. There are additional constraints of the general form of equation (4) that make sure pipe flow velocities do not violate certain specified minimum and maximum limits. This is indirectly accomplished via use of penalty terms in equation (6), which is explained in more detail in a subsequent section.

Solution Procedure

As mentioned previously, Subroutine DATAGN has been developed to process the generalized data into the specific data file required for CSUDP-SEWER.

For DATAGN, the manhole system must be defined in the following manner:

1. The most downstream manhole is numbered 1.
2. The remaining manholes are numbered with consecutive integer numbers in the upstream direction, along the length of each arbitrarily chosen branch.
3. No numbers may be omitted, so the number of the last manhole must equal the number of manholes in the system.

DATAGN partitions the branches into separate lines beginning at the branching manhole and continuing all the way upstream. Each branching manhole will appear once in its particular branch and again at the end of each of the branching lines that drain into it. These branches are then placed end to end in increasing numerical order for systematic solution by CSUDP. This is illustrated in Figure 2.

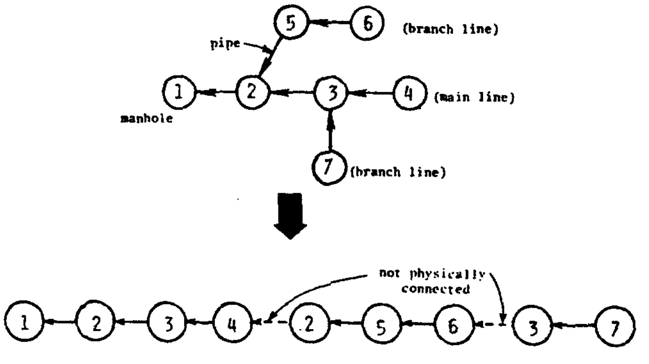


Figure 2. Transformation of branching to serial systems

In general, the program proceeds downstream manhole by manhole from the highest numbered manhole to the lowest numbered manhole for each branch. As it does this, it determines the least cost connection to each manhole in turn, for several possible downstream pipe crown elevations. These *families* of optimal solutions are stored until the last manhole in the branch is encountered. The *family* of optimal solutions for this branch are then stored as a function of pipe crown elevation of the last manhole on the branch. They are later retrieved when the analysis proceeds to the point where the junction manhole is being analyzed as a part of its own branch. At this point, the optimal costs of the branch line(s) is added into the costs for the main line. Once the end of a branch has been reached, the optimal value function and pipe diameters are reinitialized to starting values and the analysis of the next branch is commenced. This backward analysis terminates when the *family* of optimal values has been calculated for manhole no. 1 farthest downstream. The program then traces forward from manhole 1 and selects optimal vertical alignment and pipe diameters from the stored optimal families of solutions.

Pipe Calculations

At the upstream end of the pipe (i.e., at the end of stage i, or beginning of stage i+1), feasible crown elevations are set at discrete points x_{i+1}, whereas at the downstream end (i.e., beginning of stage i) elevations are set at the discrete points x_i (see Figure 3). If the discretization level is DELX, then the discrete x_i are defined as

GE(m) - XMXCOV
GE(m) - XMXCOV + DELX
GE(m) - XMXCOV + 2·DELX
GE(m) - XMXCOV + M·DELX ≤ GE(m) - XMNCOV

with M defined such that

GE(m) - XMXCOV + (M+1)·DELX > GE(m) - XMNCOV

where

XMXCOV = maximum ground cover over pipe crown
XMNCOV = minimum ground cover over pipe crown
GE(m) = ground level of manhole

In calculations previous to stage i (i.e., over stages i+1, i+2,...,N), the minimum total costs of getting to each discrete elevation point x_{i+1}, as well as the minimum pipe diameters required to do this, have been determined and stored. The program then starts at x_i = GE(m) - XMXCOV and connects a pipe to all discrete x_{i+1} values in turn. The least cost path over stage i is calculated with consideration of all previously stored costs

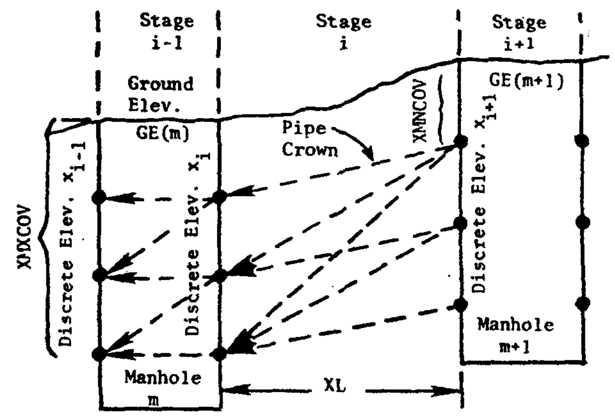


Figure 3. Illustration of crown elevation discretization

upstream of stage i for designs that start at elevation x_{i+1}. The pipe diameters required are determined from Manning's equation. If this diameter turns out to be smaller than any of the upstream pipes previously calculated, then the smallest upstream pipe is used instead. The minimum total costs for stage i and all upstream stages, as well as the optimal pipe sizes for stage i, are then stored as a function of elevation x_i. Control is then passed to the next lower stage i-1 for further calculations. The next downstream stage is a manhole calculation, which is covered in the following section.

A candidate pipe connection over stage i (i.e., connecting elevations x_i and x_{i+1}) is considered feasible only if:

1. It has a positive slope,
2. The discharge can be carried by the available set of pipe sizes, and
3. Flow velocity is within certain specified minimum and maximum levels.

If it does not satisfy the first two constraints, that path is recorded as infeasible. If it violates the velocity constraints, this alternative is given a high penalty cost which makes it unattractive but allows the analysis to continue. This penalty is eventually subtracted out so that final costs are actual costs.

The cost of supplying, excavating and laying the pipe is determined by a functional subroutine CPIPS. This is supplied by the user to suit his situation and calculates the cost per unit foot from the following parameters.

•DX (depth of downstream crown (ft.))
•DX1 (depth of upstream crown (ft.))
•DIA (pipe diameter (ins.))
•PIPTHK (pipe thickness (ins.))
for current stage i

If other parameters are required, then Subroutine OBJECS would have to be modified.

Manhole Calculations

The possible crown elevations at the downstream and upstream ends of the manhole are defined in the same way as with the pipe segments (refer again to Figure 3). If a vertical drop is allowed at a manhole, it is possible for all x_{i-1} elevations to be connected to any equal or higher x_i elevations. The least cost path to each x_i elevation is determined for all feasible paths. If no vertical drops are allowed, then x_{i-1} may only be connected to that x_i of the same elevation. That is, there is only one feasible path across the manhole for each x_{i-1} value in this case. A third option is for the user to specify a mandatory drop across a manhole to account for hydraulic losses. When the mandatory drop is specified, all feasible solutions will have this drop across the manhole even when the general drop option is not specified. When the general drops are allowed, each manhole will have the mandatory drop plus some multiple of the current DELX value.

The cost of the manhole is calculated by the functional subroutine CMNH. This function is supplied by the user and calculates the manhole cost from the following parameters:

•DX (depth of downstream crown (ft.))
•DX1 (depth of upstream crown (ft.))
•DIA (maximum of:
  (1) pipe diameters draining into the manhole
  (2) minimum allowable diameter for immediate downstream reach
  (3) diameter required to take downstream flow at maximum velocity)

Since we do not know the diameter of the pipe draining the manhole, a parameter DRPMNH may be specified. This is basically a second order correction and would normally have a value approximately equal to a pipe size increment.

•DRPMNH (over-excavation of manhole to compensate for not knowing the downstream diameter during stage i-1 calculations).

Junction Calculations

At a junction manhole, it is necessary to combine the least cost solutions for all the entering branch lines with the main line which is currently being analyzed. The procedure for doing this is described subsequently. In this discussion, the main line is the branch which is currently being analyzed, and may not be the main hydraulic carrier.

Refer again to Figure 2. In this problem, there is a main line 1-2-3-4, which is joined by branches 2-5-6 and 3-7. Assume that at an earlier stage in the analysis, the *family* of least cost solutions draining branch 2-5-6 have been stored as a function of the discrete pipe crown elevations for the pipe *exiting* manhole no. 2. Denote this total cost as $F_2^b(x_2)$. The superscript "b" signifies that this is the minimum cost of draining branch 2-5-6 through manhole no. 2, as a function of the exiting pipe elevation x_2. The cost of the manhole itself is included in $F_2^b(x_2)$.

Next, the main line itself is evaluated, which results in the minimum cost of draining the main line (as well as branch 3-7) into manhole no. 2, $F_2^m(x_2)$, as a function of the exiting pipe crown elevation, x_2. If the manhole cost is also included in this function, then the minimum total cost of draining the main line and all branches upstream of manhole no. 2 is

$$F_2(x_2) = F_2^b(x_2) + F_2^m(x_2) - CM(x_2) \qquad (9)$$

where the manhole cost $CM(x_2)$ is subtracted to avoid a *double counting* of manhole costs.

Consider the case where the main line diameter is large and enters the manhole at the mid-point of the elevation space, whereas the branch line is small and could enter the manhole at minimum cover. If drops are allowed across the manhole, the branch line will enter at minimum cover, but must drop into the manhole. The pipe exiting the manhole must have a crown elevation at mid-point since it cannot exit at a higher level than the incoming main line pipe, which dominates the solution. If, however, no drops are allowed, the slope of the branch line must be increased so that its crown enters the manhole at the mid-point, along with the larger pipe. This will be a more expensive solution because of the additional excavation needed. This procedure is then repeated for all feasible elevation points x_2 to generate the *family* of least cost solutions for manhole no. 2.

During the traceback mode, the program will find the optimal crown elevations upstream to manhole no. 2. The optimal crown elevation for the upstream end of the pipe exiting manhole no. 2, x_2^*, will be determined. Before the program proceeds to find the next optimal elevation, this value is recorded as the optimal starting elevation for each of the branch lines. When the traceback routine has reached the end of the main line, it then determines the optimal path for each of the branch lines in turn.

Refining the Solution

Once the traceback through the stages has been made and the least-cost solution found for the entire network, it may be desirable to reduce DELX and repeat the procedure in order to obtain a more accurate solution. Again, to save computer time, it is better to start with a larger DELX than desired and refine it, rather than start with a small DELX. The value of DELX is automatically halved by the Program if its value is greater than a user-specified terminal value, DELXF. In this case, a new solution space for the state variable is created for each stage which straddles the last calculated optimal pipe elevation path. This space will be 2·DELX units above and below the optimal path (5 points in all). Elevations violating cover requirements will be neglected. The solution process is then repeated until DELX $\leq$ DELXF. Care must be taken that the initial DELX is not too coarse or the code may miss the global optimal solution; or, it may not even be able to find a feasible solution.

Comments on Program Options

In the problem specification, there are a number of options that may be selected. The implications of choosing different parameter values will be briefly discussed.

Parameters Minimum Cover (XMNCOV) and Maximum Cover (XMXCOV). The system is designed such that no pipe crown will be set above the elevation of minimum cover. This may be overridden with individual manhole data specifications.

The system also sets the minimum elevation for the pipe crown. Since the pipe invert will be lower than this, then maximum cover will sometimes be violated. The system automatically checks the invert elevation. If it exceeds the maximum cover, then that pipe receives a cost penalty which makes it an unattractive option but still allows it to be considered as a feasible solution. The maximum crown elevation can be overridden by individual manhole specifications but the pipe will still attract a penalty if its invert exceeds maximum cover.

The system automatically sets the elevation of the downstream side of the first manhole to maximum cover unless it is overridden by a particular manhole specification. This is done so that the optimal elevation for the pipe or pipes draining into the last manhole can be found. With the last manhole, a drop across the manhole is always allowed unless the upstream elevation for the manhole is restricted to maximum cover. The cost of a manhole is usually small compared to the cost of the upstream pipe. Thus, the optimal path entering the last manhole will generally be as high as possible, and then drop across the manhole to maximum cover. The correct invert depth for the manhole can then be taken as the elevation of the crown of the lowest pipe entering the manhole minus its diameter. Pipe inverts below maximum cover are flagged in the output with a "#".

Parameter Defining Drops Across Manholes (IDROP). Drops may be allowed across any manhole in the system through the use of the parameter IDROP. It is recommended that the system be operated in this mode, since this flexibility will generally give lower cost solutions.

The specification may be changed so that no drops are allowed at any manholes except manhole no. 1. This will mean that at junctions, all upstream pipes enter at the same crown elevation, whether this is required by hydraulic considerations or not. This will therefore result in increased costs. Manholes with a drop are flagged in the output with a "<".

Parameter Defining Hydraulic Model (IVARN). Often, it is desirable to design a pipe that is not flowing full because of slope requirements or because the diameter cannot be reduced in the downstream direction. In this case, the actual velocities in the pipe cannot be calculated on the basis of full-pipe flow. The program uses the procedure in ASCE Manual No. 37 to estimate velocity for part-full flow and varying Manning "n" values.

The user may specify if the full-pipe or the part-full pipe model is used through the parameter IVARN. If the part-full model is used and the maximum velocity constraints are violated, a larger diameter pipe is tested to see if it allows the velocity constraint to be satisfied (i.e., for a given discharge and slope, velocity decreases with increasing pipe diameter).

Parameters Defining Minimum (MNVEL) and Maximum (MXVEL) Velocity. Through the parameters MNVEL and MXVEL, the velocity constraints for the problems may be specified. A situation may arise where the minimum pipe diameter at the beginning of a branch is so large with respect to the flow that its velocity is less than the allowable minimum velocity. When this happens, a large penalty is attached to this solution, which allows the analysis to continue. In all other cases where the velocity violates either the minimum or maximum velocity constraint, a much larger penalty is attached which will make this particular solution very unattractive while still allowing the solution to be completed. This approach has been adopted because one of the annoying and wasteful aspects of some other packages has been that the system will abort if these constraints are violated. This means that little if any information will be gained from the run. With the approach adopted in this package, an answer will more likely be generated. The user may then review it and check to see if some of the constraints can be relaxed somewhat without adverse impact on the system. For instance, suppose a maximum velocity value of 10 ft./sec. is specified, but the program can only get as low as 10.4 ft./sec. before an infeasibility occurs in some other constraint. A judgement can now be made if this is an allowable violation. If this is not acceptable, the problem must be reformulated. The output includes a display of percentage of pipe capacity used and corresponding velocity. If the velocity exceeds the maximum velocity constraint, it is flagged with "+"; or, if it violates the minimum velocity constraint it is flagged "*". It should be emphasized that the program will violate these constraints only as a last resort, if it has no other choice.

An Example Problem

Consider the typical sewer layout in Figure 4, whose characteristics are summarized in Table 1. Assume that crown elevation of the upstream end of the pipe draining manhole no. 2 is to be between 91 and 96 ft. with a diameter between 48 and 72 inches.

Table 1. Physical Characteristics of Sample Problem.

Manhole Numbers	Dischg. Into Manhole (cfs)	Lgth. of Pipe Downstream of Manhole (ft.)	Ground Elev. at Manhole (ft.)
1	—	—	100.
2	2.5	125.	102.
3	3.0	150.	103.
4	1.0	125.	105.
5	2.0	100.	106.
6	3.5	75.	108.
7	6.5	100.	109.
8	1.5	100.	103.
9	2.0	125.	105.
10	1.0	125.	109.
11	1.5	75.	110.
12	2.0	150.	107.

Other characteristics of this problem are:

- Minimum velocity - 2.0 ft./sec.
- Maximum velocity - 10.0 ft./sec.
- Manning's "n" - 0.013
- Minimum cover - 4.0 ft.
- Maximum cover - 10.0 ft.

For the calculations, the initial increment of elevation is set at 1.0 ft. and reduced by 50% until it reaches 0.0625 ft. Elevation drops are allowed at manholes and pipe diameters are not allowed to decrease in the downstream direction. The functions for the pipe costs and manhole costs are the same as those developed by Meredith (1971).

The problem was analyzed for the following set of conditions:

1. No drop allowed at manholes and pipe-full flow.
2. No drop allowed at manholes and part-full flow.
3. Drops allowed at manholes and pipe-full flow.
4. Drops allowed at manholes and part-full flow.

The results of these analyses are shown in Tables 2 and 3. Conditions 3 (not shown) and 4 (Table 3) give the least cost solution ($24,389). Notice that all the hydraulic drop (5.94 ft.) is concentrated at manhole no. 1 except for two of the three pipes draining into manhole no. 5 which both have a drop of 0.25 ft. Thus, the optimal crown elevation draining manhole no. 1 is 95.947 ft. The only difference between these two runs is that the velocities calculated under part-full hydraulic conditions are equal to or greater than those calculated with the pipe-full model. Since all velocities are well within the velocity constraint range, the design solutions for both approaches are the same.

Table 2 shows the results for Condition no. 1 (no. 2 not shown). In both of these cases, no drops are allowed and the crown elevations of the pipes draining into manhole no. 1 are set at 90.00 ft. This has the effect of forcing some of the pipe inverts below maximum cover (see values flagged with "#"). This also forces the downstream pipes to be placed at greater depths and steeper slopes. Also, the branch lines entering manhole no. 5 are at a lower elevation. These effects increase costs for Condition no. 2 to $27,412 (an increase of 12.4%) and for Condition no. 1 to $25,999 (an increase of 6.6%). The basic reason why Condition no. 1 costs more is that in order to meet the maximum velocity constraints, the larger pipes (e.g., between manholes 1 and 2) must be placed at a flatter grade. This means that more pipe lengths must be placed at greater depths.

These results show how sensitive the optimal design is to the system constraints chosen. The user would be well advised to evaluate the sensitivity of the optimal design to the constraints and perhaps perform a tradeoff analysis.

Comparison With Other Approaches

Introduction

The performance of the CSUDP-SEWER package was compared with published results of other researchers. No detailed comparisons are included on computer costs and execution times since these values are machine specific.

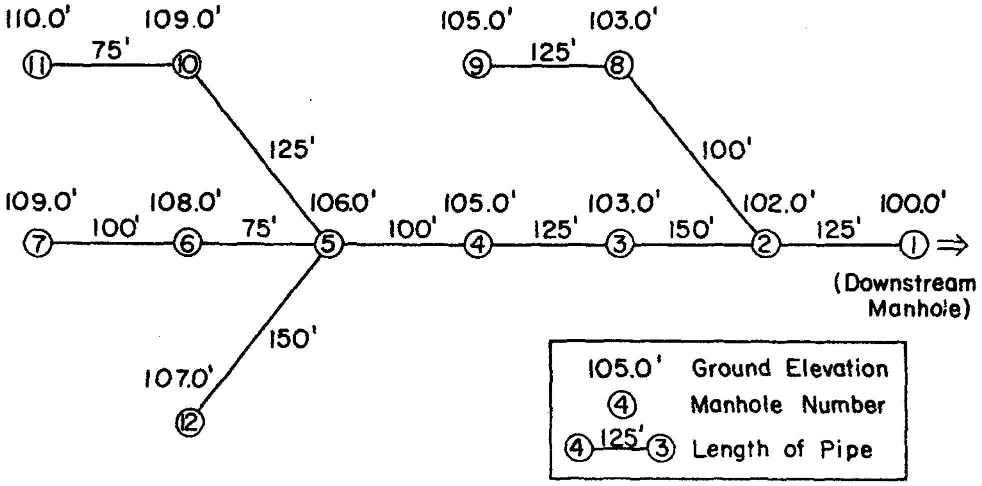


Figure 4. Typical layout of a storm sewer system.

Table 2. Example Problem: No Drops Allowed and Pipe-Full Flow.

Optimal Solution For X(1) = 90.0000

I	X*	U*	MNH. NO.	ELEV. (FT)	DROP (FT)	SLOPE	DIAM. (INS)	DIS. (CFS)	PT.DIS (CFS)	% CAP.	VEL. (FT/ SEC)	LENGTH (FT)
1	90.00000	12.00000	1 D/S	90.000 #	0.000							
2	90.00000	12.00000	1 U/S	90.000 #		.0440	48.0	26.50	301.86	8.8	2.11	125.00
3	95.50000	12.00000	2 D/S	95.000 #	0.000							
4	95.50000	6.000000	2 U/S	95.500		.0196	21.0	20.50	22.21	92.3	8.52	150.00
5	98.43750	6.000000	3 D/S	98.438	0.000							
6	98.43750	6.000000	3 U/S	98.438		.0125	21.0	17.50	17.75	98.6	7.28	125.00
7	100.0000	6.000000	4 D/S	100.000	0.000							
8	100.0000	6.000000	4 U/S	100.000		.0175	21.0	16.50	21.00	78.6	6.86	100.00
9	101.7500	5.000000	5 D/S	101.750	0.000							
10	101.7500	4.000000	5 U/S	101.750		.0292	15.0	10.00	11.05	90.5	8.15	75.00
11	103.9375	4.000000	6 D/S	103.938	0.000							
12	103.9375	4.000000	6 U/S	103.938		.0106	15.0	6.50	6.67	97.4	5.30	100.00
13	105.0000	3.000000	7 D/S	105.000	0.000							
14	105.0000	12.00000	7 U/S	105.000								
15	95.50000	12.00000	2 D/S	95.500 #	0.000							
16	95.50000	2.000000	2 U/S	95.500		.0359	10.0	3.50	4.11	85.2	6.42	100.00
17	99.00000	2.000000	8 D/S	99.000	0.000							
18	99.00000	2.000000	8 U/S	99.000		.0160	10.0	2.00	2.78	72.0	3.67	125.00
19	101.0000	1.000000	9 D/S	101.000	0.000							
20	101.0000	1.000000	9 U/S	101.000								
21	101.7500	2.000000	5 D/S	101.750	0.000							
22	101.7500	2.000000	5 U/S	101.750		.0245	10.0	2.50	3.44	72.8	4.58	125.00
23	104.8125	1.000000	10 D/S	104.813	0.000							
24	104.8125	1.000000	10 U/S	104.813		.0158	8,0	1.50	1.52	98.5	4.30	75.00
25	106.0000	1.000000	11 D/S	106.000	0.000							
26	106.0000	1.000000	11 U/S	106.000								
27	101.7500	2.000000	5 D/S	101.750	0.000							
28	101.7500	2.000000	5 U/S	101.750		.0083	10.0	2.00	2.00	99.8	3.67	150.00
29	103.0000	1.000000	12 D/S	103.000	0.000							
30	103.0000	1.000000	12 U/S	103.000								

Minimum Objective
Value = 25999.41

The first comparison uses the drainage problem presented in ASCE Manual No. 37. The results are compared to those of Yen, et al. (1976). The second comparison is for a problem presented by Mays, (1976). Both of these studies use the Illinois Storm Sewer Design (ILSSD) Model, which is based on a discrete differential dynamic programming approach.[11]

Table 4 summarizes the options used in the published studies and those analyzed by CSUDP-SEWER. In order to compare the results for the different computer packages, a CSUDP-SEWER analysis was made with the solution constrained to the same optimal elevations and pipe diameters as the published studies. These are the first CSUDP-SEWER runs shown in each section of Table 4. For all comparisons, Manning's n=0.013 and minimum velocity is 2 ft./sec. For the ASCE problem, maximum velocity is 10 ft./sec., whereas for the Mays problem it is 8 ft./sec. The final DELX elevation accuracy for CSUDP-SEWER was 0.0625 ft., which is believed comparable to the ILSSD results.

ASCE Problem

The ASCE problem has been analyzed with CSUDP-SEWER with the options shown in Table 4. This network has 12 manholes and 11 pipe links. The optimal costs for all CSUDP-SEWER options and the cost determined by ILSSD are very similar. The optimal cost calculated by ILSSD was $69,062, whereas the cost calculated by CSUDP-SEWER for the same elevations and pipe diameters was $69,377. This 0.5% increase in cost is due to the slightly differing costing model used. When the elevation and diameter constraints on CSUDP-SEWER were relaxed, the cost for a model with no drops was $69,362. However, when drops were allowed, costs lowered to $68,822. This is about a 0.8% decrease, which is still not very significant. The main reason that there is so little difference is that many of the manholes are at minimum cover and hence the optimizing packages have not been extended to test their potential. The computer output for the case of drops allowed and pipe-full flow assumptions is shown in Table 5. Note that the velocity in the pipe draining manhole 9 is slightly less than the minimum velocity constraint in this case. Interestingly enough when part-full assumption were used, the velocity was acceptable. Execution time on the CSU computer* was 7.6 sec.

Mays' Problem

As with the ASCE-Problem, the optimal published crown elevation and pipe diameters were analyzed for the design problem. This network includes 21 manholes and 20 pipe sections. CSUDP-SEWER was not able to find a solution for these conditions. The basic reason for this was the hydraulic model used in CSUDP-SEWER determined that some reaches required larger diameter pipes than those found by the ILSSD package in order to give a feasible solution. In

---

*CDC Cyber 172 Computing System

Table 3. Example Problem: Drops Allowed at Manholes and Part-Full Flow.

Optimal Solution For X(1) = 90.0000

I	X*	U*	MNH. NO.	ELEV. (FT)	DROP (FT)	SLOPE	DIAM. (INS)	DIS. (CFS)	PT.DIS (CFS)	% CAP.	VEL. (FT/SEC)	LENGTH (FT)
1	90.00000	12.00000	1 D/S	90.000 #	5.93B <							
2	95.93750	12.00000	1 U/S	95.937		.0005	48.0	26.50	32.18	83.4	2.11	125.00
3	96.00000	12.00000	2 D/S	96.000	0.000							
4	96.00000	7.000000	2 U/S	96.000		.0163	24.0	20.50	28.89	71.0	6.53	150.00
5	98.43750	6.000000	3 D/S	98.438	0.000							
6	98.43750	6.000000	3 U/S	98.438		.0125	21.0	17.50	17.75	98.6	7.28	125.00
7	100.0000	6.000000	4 D/S	100.000	0.000							
8	100.0000	6.000000	4 U/S	100.000		.0175	21.0	16.50	21.00	78.6	6.86	100.00
9	101.7500	5.000000	5 D/S	101.750	.250 <							
10	102.0000	4.000000	5 U/S	102.000		.0258	15.0	10.00	10.40	96.1	8.15	75.00
11	103.9375	4.000000	6 D/S	103.938	0.000							
12	103.9375	4.000000	6 U/S	103.938		.0106	15.0	6.50	6.67	97.4	5.30	100.00
13	105.0000	3.000000	7 D/S	105.000	0.000							
14	105.0000	12.00000	7 U/S	105.000								
15	96.00000	12.00000	2 D/S	96.000	0.000							
16	96.00000	2.000000	2 U/S	96.000		.0300	10.0	3.50	3.80	92.1	6.42	100.00
17	99.00000	2.000000	8 D/S	99.000	0.000							
18	99.00000	2.000000	8 U/S	99.000		.0160	10.0	2.00	2.78	72.0	3.67	125.00
19	101.0000	1.000000	9 D/S	101.000	0.000							
20	101.0000	1.000000	9 U/S	101.000								
21	101.7500	2.000000	5 D/S	101.750	.250 <							
22	102.0000	2.000000	5 U/S	102.000		.0225	10.0	2.50	3.29	75.9	4.58	125.00
23	104.8125	1.000000	10 D/S	104.813	0.000							
24	104.8125	1.000000	10 U/S	104.813		.0158	8.0	1.50	1.52	98.5	4.30	75.00
25	106.0000	1.000000	11 D/S	106.000	0.000							
26	106.0000	1.000000	11 U/S	106.000								
27	101.7500	2.000000	5 D/S	101.750	0.000							
28	101.7500	2.000000	5 U/S	101.750		.0083	10.0	2.00	2.00	99.8	3.67	150.00
29	103.0000	1.000000	12 D/S	103.000	0.000							
30	103.0000		12 U/S	103.000								

Minimum Objective
Value = 24389.43

some cases, calculated diameters were only 0.5 inches larger than the constrained values. Unfortunately, this still requires the next larger commercial pipe size to be selected, which represents a six inch increase in diameter. The problem was rerun for the same crown elevations but without restricting the pipe set. The solution became feasible, but six pipes were determined to have a diameter one size larger than Mays' results. These results show the sensitivity of the solution to the hydraulic assumptions. The cost of this layout was determined to be $298,400, which is 12% greater than the value of $265,355 quoted by Mays. This is due to larger diameter pipes specified by CSUDP-SEWER.

When the problem was rerun without constraining the elevations, the cost dropped to $274,462 (an 8.8% reduction) for the pipe-full model, with or without drops (see Table 6). In this case, the pipe sets were similar to Mays' results with some pipes one size larger and others one size smaller. When the part-full model was used without drops, the cost only fell to $291,682, which represents a 2.3% decrease. When drops were allowed, cost decreased to $277,877 (a 6.9% decrease). The reason the part-full model gives a slightly higher cost is that some of the larger pipes near the downstream exit are placed at rather steep slopes and are only running part full. This means that their velocities are far

Table 4. Comparison of CSUDP-SEWER with ILSSD Results.

Problem Analyzed	ASCE PROBLEM						MAY'S PROBLEM					
Computer Package	ILSSD	CSUDP - SEWER					ILSSD	CSUDP - SEWER				
Constraints		*						*				
1. Drops Allowed	Yes	Yes	No	No	Yes	Yes	Yes	Yes	No	No	Yes	Yes
2. Pipe Flow Model	Full	Full	Full	Part.	Full	Part.	Full	Full	Full	Part.	Full	Part.
3. Min. Cover (ft)	3.5	3.5	3.5	3.5	3.5	3.5	8	8	8	8	8	8
4. Max. Cover (ft)	†	8,5	8.5	8.5	8.5	8.5	†	14	14	14	14	14
Costs ($)	69062	69377	69362	69362	68822	68822	265355	298400	274463	291682	274463	277878

* Pipe elevations and diameters set to the ILSSD results
† Not specified

Table 5. ASCE Problem: Drops Allowed and Pipe-Full Flow.

I	X*	U*	MNH. NO.	ELEV. (FT)	DROP (FT)	SLOPE	DIAM. (INS)	DIS. (CFS)	PT.DIS (CFS)	% CAP.	VEL. (FT/SEC)	LENGTH (FT)
1	82.00000	10.00000	1 D/S	82.000 #	.500 <							
2	82.50000	10.00000	1 U/S	82.500		.0080	36.0	44.40	59.77	74.3	6.28	125.00
3	83.50000	10.00000	2 D/S	83.500	0.000							
4	83.50000	10.00000	2 U/S	83.500		.0031	36.0	35.30	37.35	94.5	4.99	400.00
5	84.75000	10.00000	3 D/S	84.750	0.000							
6	84.75000	10.00000	3 U/S	84.750		.0017	36.0	27.70	27.95	99.1	3.92	400.00
7	85.45000	7.000000	4 D/S	85.450	0.000							
8	85.45000	6.000000	4 U/S	85.450		.0071	21.0	11.70	13.40	87.3	4.86	400.00
9	88.30000	4.000000	5 D/S	88.300	0.000							
10	88.30000	4.000000	5 U/S	88.300		.0077	15.0	5.10	5.70	89.5	4.16	400.00
11	91.40000	3.000000	6 D/S	91.400	0.000							
12	91.40000	3.000000	6 U/S	91.400		.0088	12.0	2.00	3.34	59.9	2.55	400.00
13	94.90000	3.000000	7 D/S	94.900	0.000							
14	94.90000	1.000000	7 U/S	94.900								
15	83.50000	3.000000	2 D/S	83.500	0.000							
16	83.50000	3.000000	2 U/S	83.500		.0115	12.0	2.00	3.83	52.3	2.55	400.00
17	88.10000	3.000000	8 D/S	88.100	0.000							
18	88.10000	1.000000	8 U/S	88.100								
19	84.75000	3.000000	3 D/S	84.750	0.000							
20	84.75000	3.000000	3 U/S	84.750		.0111	12.0	2.50	3.76	66.4	3.18	400.00
21	89.20000	3.000000	9 D/S	89.200	0.000							
22	89.20000	3.000000	9 U/S	89.200		.0048	12.0	1.50	2.46	61.0	1.91 *	400.00
23	91.10000	3.000000	10 D/S	91.100	0.000							
24	91.10000	1.000000	10 U/S	91.100								
25	85.45000	4.000000	4 D/S	85.450	0.000							
26	85.45000	4.000000	4 U/S	85.450		.0084	15.0	5.90	5.92	99.6	4.81	400.00
27	88.80000	4.000000	11 D/S	88.800	0.000							
28	88.80000	4.000000	11 U/S	88.800		.0097	15.0	4.70	6.39	73.6	3.83	400.00
29	92.70000	3.000000	12 D/S	92.700	0.000							
30	92.70000		12 U/S	92.700								

Minimum Objective
Value = 69361.75

greater than the maximum allowed velocity of 8.0 ft./sec. A velocity of 11.8 ft./sec. was calculated for one of the pipe sections. These values were only selected by the model because there were no feasible solutions that satisfied the velocity constraints. The execution time on the computer was 12.4 sec.

These problems have served to illustrate the applicability of CSUDP-SEWER and document its performance. The readers are referred to Robinson and Labadie (1981) for a discussion of the data requirements for CSUDP-SEWER.[12]

Conclusions

A computer model utilizing a generalized dynamic programming routine and a package of subroutines specifically designed for storm sewer design has been compared with published results of other modeling studies. In most cases, the differences in results are small, with regard to both total cost and pipe size selection. Larger discrepancies can be explained in the differences in hydraulic model assumptions. The restriction to commercially available pipe sizes can produce cost discrepancies because even though the required pipe diameter may be only slightly higher than next smaller commercial size, the larger commercial size must be selected. One could argue that with the inaccuracies and arbitrariness involved in selecting a design storm and utilizing steady flow assumptions, there would be some justification in using the smaller size pipe.

The modeling package has been designed for ease of use by practitioners, and includes most of the design factors that would be encountered in the field. This computer package should be regarded as a screening model. If complex hydraulics are known to exist in certain sections of the network, the optimal design should be further evaluated by an appropriate unsteady flow model.

Acknowledgments

This research was supported in part by the Office of Water Research and Technology, U. S. Department of the Interior, as authorized by the Water Research and Development Act of 1978. Contents of this paper do not necessarily reflect the views of the Office of Water Research and Technology. The authors are grateful to Mr. Mark Calabrese for his help with some of the computer work.

References

1. Deininger, R. A., "Systems Analysis for Water Supply and Pollution Control," Natural Resources Systems Models in Decision Making, Ed. by G. H. Toebes, Water Resources Center, Purdue University, Lafayette, Ind., 1970.

2. Holland, M. G., "Computer Models of Wastewater Collection Systems," Water Resources Group, Harvard University, Cambridge, Mass., 1966.

Table 6. Mays' Problem: No Drops Allowed and Pipe-Full Flow.

I	X*	U*	MNH. NO.	ELEV. (FT)	DROP (FT)	SLOPE	DIAM. (INS)	DIS. (CFS)	PT.DIS (CFS)	% CAP.	VEL. (FT/SEC)	LENGTH (FT)
1	431.0000	9.000000	1 D/S	431.000 #	6.000 <							
2	437.0000	9.000000	1 U/S	437.000		.0050	48.0	94.00	101.76	92.4	7.48	600.00
3	440.0000	9.000000	2 D/S	440.000	0.000							
4	440.0000	9.000000	2 U/S	440.000		.0060	48.0	89.00	111.47	79.8	7.08	500.00
5	443.0000	9.000000	3 D/S	443.000	0.000							
6	443.0000	9.000000	3 U/S	443.000		.0100	48.0	87.00	143.91	60.5	6.92	400.00
7	447.0000	9.000000	4 D/S	447.000	0.000							
8	447.0000	8.000000	4 U/S	447.000		.0167	42.0	71.00	130.12	54.6	7.38	600.00
9	457.0000	8.000000	5 D/S	457.000	0.000							
10	457.0000	7.000000	5 U/S	457.000		.0100	36.0	44.00	66.82	65.8	6.22	500.00
11	462.0000	7.000000	6 D/S	462.000	0.000							
12	462.0000	5.000000	6 U/S	462.000		.0182	24.0	22.00	30.56	72.0	7.00	550.00
13	472.0000	5.000000	7 D/S	472.000	0.000							
14	472.0000	2.000000	7 U/S	472.000		.0200	15.0	9.00	9.15	98.3	7.33	350.00
15	479.0000	2.000000	8 D/S	479.000	0.000							
16	479.0000	2.000000	8 U/S	479.000		.0200	15.0	7.00	9.15	76.5	5.70	400.00
17	487.0000	2.000000	9 D/S	487.000	0.000							
18	487.0000	1.000000	9 U/S	487.000		.0143	12.0	4.00	4.27	93.8	5.09	350.00
19	492.0000	1.000000	10 D/S	492.000	0.000							
20	492.0000	1.000000	10 U/S	492.000								
21	447.0000	3.000000	4 D/S	447.000	0.000							
22	447.0000	3.000000	4 U/S	447.000		.0143	18.0	9.00	12.58	71.6	5.09	350.00
23	452.0000	2.000000	11 D/S	452.000	0.000							
24	452.0000	2.000000	11 U/S	452.000		.0098	15.0	6.00	6.40	93.7	4.89	300.00
25	454.9375	1.000000	12 D/S	454.938	0.000							
26	454.9375	1.000000	12 U/S	454.938		.0127	12.0	4.00	4.02	99.6	5.09	400.00
27	460.0000	1.000000	13 D/S	460.000	0.000							
28	460.0000	1.000000	13 U/S	460.000								
29	457.0000	5.000000	5 D/S	457.000	0.000							
30	457.0000	5.000000	5 U/S	457.000		.0143	24.0	20.00	27.09	73.8	6.37	350.00
31	462.0000	5.000000	14 D/S	462.000	0.000							
32	462.0000	4.000000	14 U/S	462.000		.0143	21.0	16.00	18.97	84.3	6.65	350.00
33	467.0000	4.000000	15 D/S	467.000	0.000							
34	467.0000	2.000000	15 U/S	467.000		.0200	15.0	9.00	9.15	98.3	7.33	500.00
35	477.0000	2.000000	16 D/S	477.000	0.000							
36	477.0000	1.000000	16 U/S	477.000								
37	462.0000	4.000000	6 D/S	462.000	0.000							
38	462.0000	4.000000	6 U/S	462.000		.0143	21.0	16.00	18.97	84.3	6.65	350.00
39	467.0000	4.000000	17 D/S	467.000	0.000							
40	467.0000	3.000000	17 U/S	467.000		.0222	18.0	12.00	15.69	76.5	6.79	450.00
41	477.0000	3.000000	18 D/S	477.000	0.000							
42	477.0000	3.000000	18 U/S	477.000		.0100	18.0	8.00	10.52	76.0	4.53	500.00
43	482.0000	2.000000	19 D/S	482.000	0.000							
44	482.0000	1.000000	19 U/S	482.000								
45	472.0000	3.000000	7 D/S	472.000	0.000							
46	472.0000	3.000000	7 U/S	472.000		.0115	18.0	8.00	11.28	70.9	4.53	430.00
47	476.9375	2.000000	20 D/S	476.938	0.000							
48	476.9375	1.000000	20 U/S	476.938		.0127	12.0	4.00	4.02	99.6	5.09	400.00
49	482.0000	1.000000	21 D/S	482.000	0.000							
50	482.0000		21 U/S	482.000								

Minimum Objective
Value = 274462.6

3. Meredith, D. D., "Dynamic Programming with Case Study on Planning and Design of Urban Water Facilities," Sec. IX, Treatise on Urban Water Systems, Colorado State University, pp. 590-652, July 1971.

4. Merritt, L. B., and Bogan, R. H., "Computer-Based Optimal Design of Sewer Systems," Jour. Environmental Engineering Division, ASCE, Vol. 99, No. EE1, pp. 35-53, Feb. 1973.

5. Froise, S., Burges, S. J., and Bogan, R. H., "A Dynamic Programming Approach to Determine Least Cost Strategies in Urban Network Design," paper presented at ASCE Specialty Conference on Water Resources Planning and Management, Colorado State University, Fort Collins, Colorado, July 9-11, 1975.

6. Yen, B. C., Wenzel, H. G., Mays, L. W. and Tang, W. H., "Advanced Methodologies for Design of Storm Sewer Systems," Research Report No. 112, Water Resources Center, University of Illinois at Urbana-Champaign, Urbana, Illinois, August 1976.

7. Froise, S., and Burges, S., "Least-Cost Design of Urban-Drainage Networks," Journal of the

Water Resources Planning and Management Division, ASCE, Vol. 104, #WR1, pp. 75-92, November 1978.

8. Labadie, J.W., Morrow, D. M., and Lazaro, R.C., "Urban Stormawater Control Package for Automated Real-Time Systems," Colorado State University, Report C 6174, prepared for U.S. Dept. of Interior, Office of Water Research and Technology, December 1978.

9. Howell, David, University of New South Wales, Australia, private communication, 1981.

10. Larson, R.E., State Increment Dynamic Programming, American Elsevier Publishing Company, Inc., New York, 1968.

11. Mays, L.W., "Optimal Layout and Design of Storm Sewer Systems," Ph.D. Thesis, Department of Civil Engineering, University of Illinois at Urbana-Champaign, Illinois, 1976

12. Robinson, D.K. and Labadie, J.W., "Optimal Design of Urban Stormwater Drainage Systems: User's Manual," Department of Civil Engineering, Colorado State University, Ft. Collins, Colorado, January 1981.

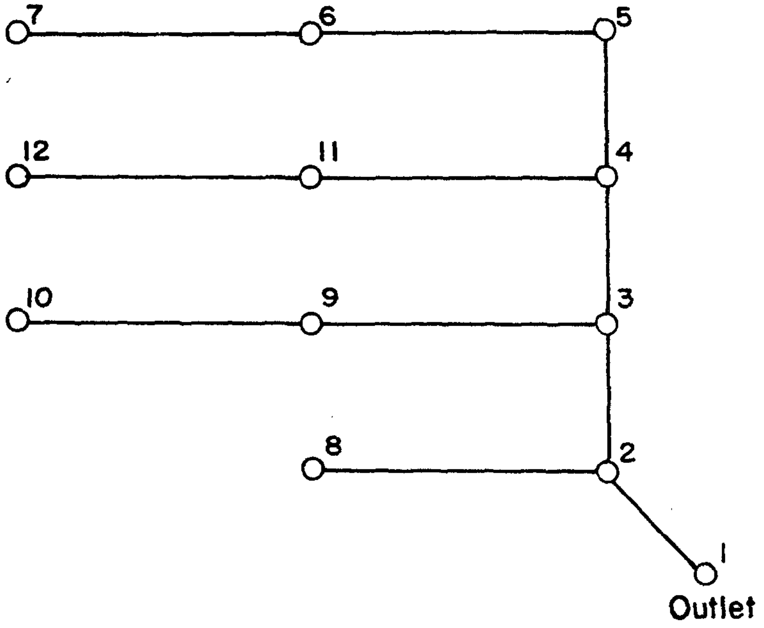


Figure 5. Layout for ASCE Problem.

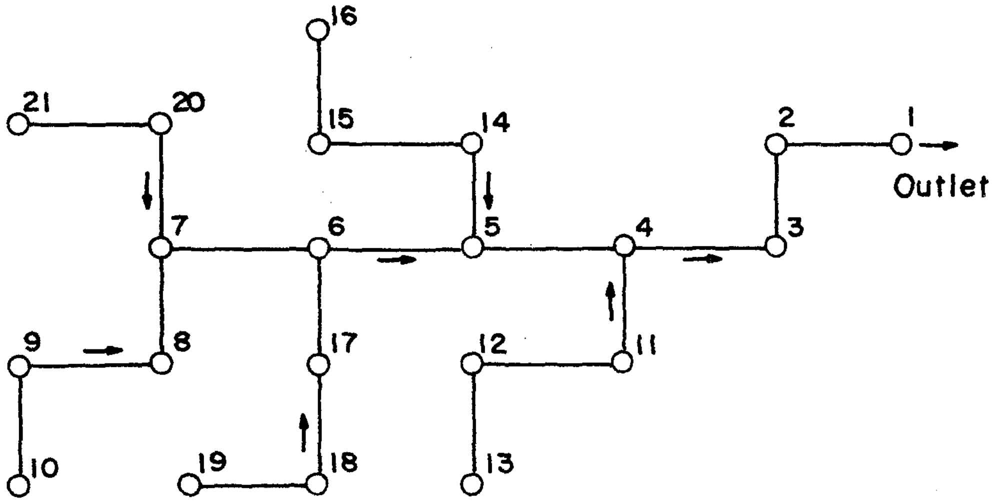
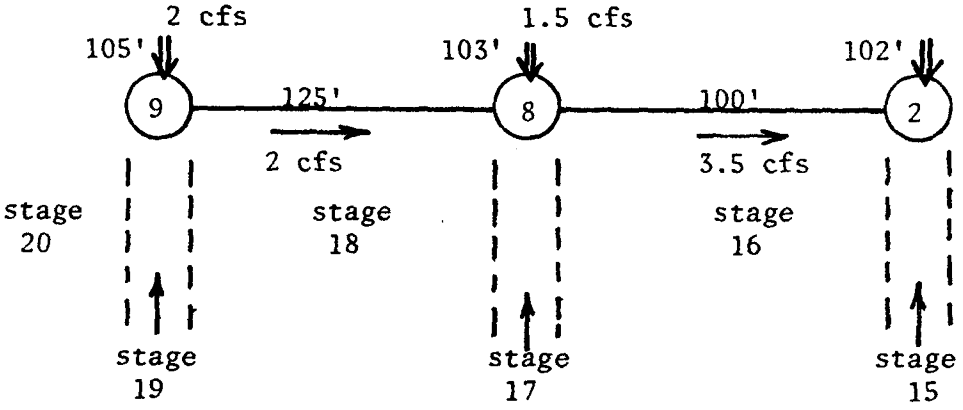


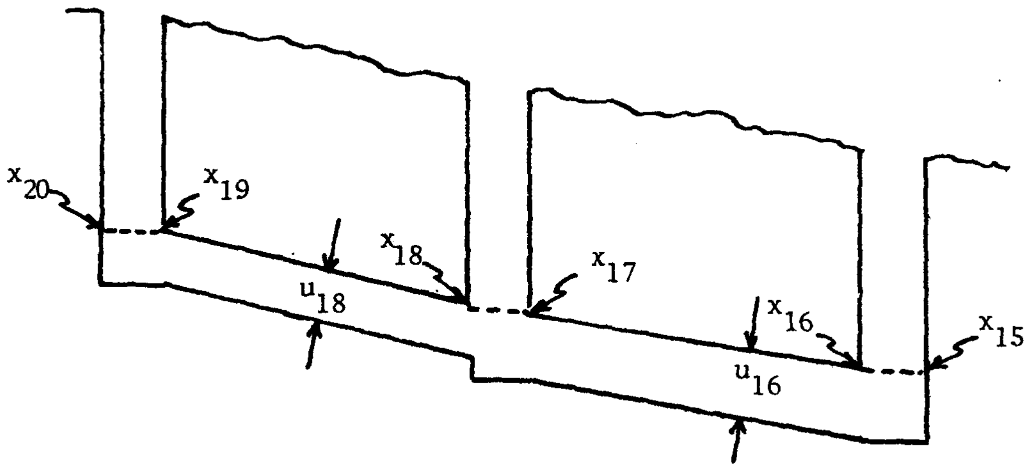
Figure 6. Layout for Mays' Problem.

EXAMPLE PROBLEM

Program CSUDP-SEWER*

We will just look at the lateral converging to manhole #2.





Assume:

- no drops allowed
- pipe full flow for velocity computations $\left\{ \begin{array}{l} V_{min} = 2 \text{ ft./sec.} \\ V_{max} = 10 \text{ ft./sec.} \end{array} \right.$
- n = 0.013
- minimum cover = 4 ft.
- maximum cover = 10 ft.

We will ignore these for this example

---

*pg. 7 of "Optimal Design of Urban Stormwater Drainage Systems," D. Robinson and J. Labadie.

- pipe cost/ft. $= C_i = \{13.0 + 0.8(\overset{\text{(ave. invert depth)}}{\overline{H}_i} - 10) + 0.915(\overset{\text{(diam. in inches)}}{u_i} - 12)\}$
- manhole cost $CM_i = 250 + (\underset{\nwarrow\text{manhole depth in ft.}}{MH_i})^2$

$$u_i = \left[\frac{2.16nQ_i}{\sqrt{(x_{i+1} - x_i)/L_i}}\right]^{0.375} \cdot 12 \qquad \overline{H}_i = \frac{(GE_{i+1} - x_{i+1} + u_i) + (GE_i - x_i + u_i)}{2}$$

To simplify — the possible crown elevations are:

$$x_{20} = x_{19} = \begin{bmatrix} 102' \\ 101' \\ 100' \end{bmatrix} \qquad x_{18} = x_{17} = \begin{bmatrix} 100' \\ 99' \\ 98' \end{bmatrix} \qquad x_{16} = x_{15} = \begin{bmatrix} 96.5' \\ 95.5' \\ 94.5' \end{bmatrix}$$

DP Recursion Equation:

$$F_i(x_i) = \min_{x_{i+1}} [C_i(x_i, x_{i+1}, u_i) \text{ for pipe}$$

or

$$CM_i(GE_i - x_i + \underset{\uparrow\text{approximation}}{u_i}) \text{ for manhole } + F_{i+1}(x_{i+1})]$$

where

$$\underset{\text{in inches}}{u_i} = \left[\frac{2.16nQ_i}{\sqrt{(x_{i+1} - x_i)/L_i}}\right]^{0.375} \cdot 12 \qquad \boxed{\begin{matrix} x_{i+1} = x_i \text{ if} \\ i = \text{manhole} \end{matrix}}$$

(increase u_i to 1st commercial size)

Then check velocity constraints:

$$2 \leq \left[\frac{Q_i}{(\pi u_i^2/576}\right] \leq 10 \qquad \text{*pipe full model}$$

Stage 20 [for initialization only]

$$F_{20}(x_{20}) = 0$$

Stage 19

$$\underset{\substack{\text{stage}\\ \#19}}{F_{19}(x_{19})} = \underset{\substack{\text{manhole}\\ \#9}}{CM_{19}} = 250 + (GE_{19} - x_{19} + \overset{\substack{\text{must guess at}\\ \text{this time}}}{u_{18}})^2 + F_{20}(x_{20})$$

$$= 250 + (105 - x_{19} + 0.8)^2 + 0 \quad [x_{19} = x_{20}]$$

x_{19}	$F_{19}(x_{19})$
102	264.4
101	273.0
100	283.6

Stage 18

$$F_{18}(x_{18}) = \min_{x_{19}}\{[13.0 + 0.8(\overline{H}_{18} - 10) + 0.915(u_{18} - 12)]\cdot L_{18} + F_{19}(x_{19})\}$$

x_{18}	x_{19}	u_{18}	(comm) u_{18}	velocity check	C_{18}	$F_{19}(x_{19})$	minimum $F_{18}(x_{18})$	$u^*_{18}(x_{18})$
100	102	8.85"	10"	3.67	779.6	264.4	1044	10"
	101	10.08	12"	2.55	1075.	273.0	1248	
	100	(INFEAS)						
99	102	8.20"	10"	3.67	829.6	264.4	1094	10"
	101	8.85"	10"	3.67	879.6	273.0	1152.6	
	100	10.08"	12"	2.55	1175.	283.6	1458.6	
98	102	7.75"	8"	5.73	634.17	264.4	898.6	8"
	101	8.20"	10"	3.67	929.6	273.0	202.6	
	100	8.85"	10"	3.67	979.5	283.6	1263.	

store this information

Stage 17

$$F_{17}(x_{17}) = \underset{\text{manhole } \#8}{CM_{17}} + F_{18}(x_{18})$$

{ no minimization over x_{18} since no drops allowed: $x_{17} = x_{18}$

$$CM_{17} = 250 + (103 - x_{17} + \underset{\text{guessed value}}{0.8})^2$$

x_{17}	CM_{17}	$F_{18}(x_{18})$	$F_{17}(x_{17})$
100	264.4	1044	1308.4
99	273.0	1094	1367
98	283.6	898.6	1182

Stage 16

$$F_{16}(x_{16}) = \min_{x_{17}}\{[13.0 + 0.8(\overline{H}_{16}-10) + 0.915(u_{16}-12)]\cdot L_{16} + F_{17}(x_{17})\}$$

x_{16}	x_{17}	u_{16}	(comm) u_{16}	velocity check	C_{16}	$F_{17}(x_{17})$	$F_{16}(x_{16})$	$u^*_{16}(x_{16})$
96.5	100	9.43	10"	6.42	724	1308.4	$\boxed{2032.4}$	10"
	99	10.04	*10"	6.42	764	1367	2131	
	98	11.05	12"	4.46	987	1182	2169	
95.5	100	8.99	10"	6.42	764	1308.4	2072	
	99	9.43	10"	6.42	804	1367	2171	
	98	10.04	*10"	6.42	844	1182	$\boxed{2026\circledast}$	10"
94.5	100	8.66	10"	6.42	804	1308.4	2112.4	
	99	8.99	10"	6.42	844	1367	2211	
	98	9.43	10"	6.42	884	1182	$\boxed{2066}$	10"

*the program would increase this to 12", even though we are only slightly above 10" (would choose $x^*_{17} = 19$ as optimum).

TRACEBACK

$x^*_{16} = \underline{\underline{95.5}}$ - so $x^*_{17} = \underline{\underline{98}} = x^*_{18}$

now go back to stage 18

If $x_{18} = 98$ - then $x^*_{19} = \underline{\underline{102}}$

Note: If there are still downstream sections to be sized and vertically aligned, we would store the optimal information above $F_{16}(x_{16})$, for each discrete x_{16} and continue downstream. Other branches coming into manhole #2 would be solved the same way, and then their optimal value functions would be added together.

APPENDIX D

USER GUIDE TO PROGRAM CSUDP/SEWER

A. INTRODUCTION

A suite of programs have been developed in conjunction with a generalized dynamic programming code called CSUDP which optimize the vertical alignment of a stormwater collection system. Details of the procedure can be found in the attached paper by Robinson and Labadie (1981). The program is currently dimensioned to analyze a system where the sum of the number of manholes and the number of branch lines is less than 100. Larger systems may be analyzed if the array dimensions in Program CSUDP are increased. The program allows up to three pipes (or upstream manholes) to drain into any manhole, which in turn is drained by one pipe.

An auxiliary program called DATAGN transforms the multi-branched sewer system into a pseudo-serial system for solution by CSUDP-SEWER. In the pseudo-serial formulation, each pipe as well as each manhole is represented as a stage. Each junction manhole is represented by one stage for each incoming pipe. The operation of the system is illustrated with a simple example which highlights the various features of the system. A brief description of the structure of the key subroutines serves to illustrate the method of solution adopted. A precise description of the data requirements if given as well as a general discussion of the implications of the significant options that have been incorporated.

In CSUDP-SEWER, the manhole must be numbered in the following way:

1. The most downstream manhole is numbered 1.
2. The remaining manholes are numbered with consecutive integer numbers in the upstream direction along the length of each branch. The branches may be chosen arbitrarily, but adjacent

manholes must be numbered consecutively, except at junctions. In this case, the number of the first manhole on the branch may not be numbered consecutively with the manhole number at the junction. The manhole numbers should continue on consecutively as we move upstream along that branch. An example problem is presented in a later section.

3. No numbers may be omitted and the number of the last manhole must equal the number of manholes in the system.

B. SOLUTION PROCEDURE

Program CSUDP solves a serial dynamic programming problem. The subroutines associated with the SEWER package (i.e., OBJECS, STATS, READINS, PIPVEL, HDVAL, CPIP and CMNH) require the branching sewer system to be reformulated into a pseudo-serial problem which can be solved by CSUDP. Subroutine DATAGN has been developed to process the generalized data (described subsequently in a Section on Data Requirements) into the specific data file required for CSUDP-SEWER.

In general, the program proceeds downstream manhole by manhole from the highest numbered manhole to the lowest numbered manhole. As it does this, it determines the least cost connection to each manhole in turn, for several possible downstream pipe crown elevations. These families of optimal solutions are stored until manhole number 1 is encountered. A traceback procedure is then employed which works upstream and finds the best overall design. The details of the pipe and manhole calculations will be discussed separately. Refer to Figure 1 for the following discussions.

B.1 Pipe Calculations

At the upstream end of a pipe (i.e., at the end of stage n+1, or beginning of stage n+2) its crown elevations are set at the discrete

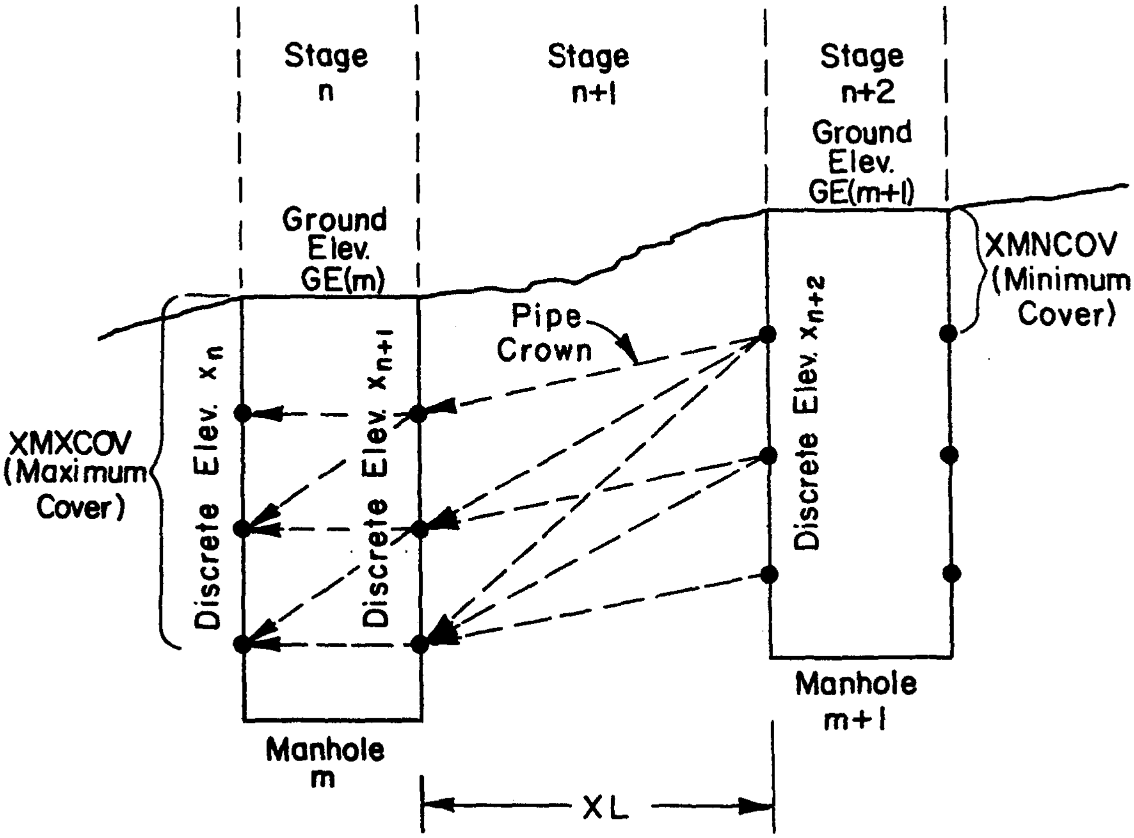


Figure 1. Illustration of pipeline calculations showing alternative pipe slopes and elevations.

points x_{n+2}, whereas at the downstream end (i.e., beginning of stage n+1) elevations are set at the discrete points x_{n+1}. If the discretization level is DELX, then the discrete x_{n+1} are defined as

$$GE(m) - XMXCOV$$

$$GE(m) - XMXCOV + DELX$$

$$GE(m) - XMXCOV + 2\ DELX$$

$$GE(m) - XMXCOV + M\ DELX \leq GE(m) - XMNCOV$$

with M defined such that

$$GE(m) - XMXCOV + (M+1)\ DELX > GE(m) - XMNCOV$$

where

XMXCOV = maximum ground cover over pipe crown

XMNCOV = minimum ground cover over pipe crown

Similar discretizations are defined for all other stages, with differing ground elevations and maximum and minimum cover, but DELX stays the same.

In calculations previous to stage n+1 (i.e., over stages n+2,n+3,...,N), the minimum total costs of getting to each discrete elevation point x_{n+2}, as well as the minimum pipe diameters required to do this, have been determined and stored. The program then starts at x_{n+1} = GE(m)-XMXCOV and connects a pipe to all discrete x_{n+2} values in turn. It then calculates the least cost path over stage n+1, with consideration of all previously stored costs upstream of stage n+1 for designs that start at elevation x_{n+2}. The pipe diameters required are determined from Manning's equation. If this diameter turns out to be smaller than any of the upstream pipes previously calculated, then the smallest upstream pipe is used instead. The minimum total costs for stage n+1 and all upstream stages, as well as the optimal pipe sizes for stage n+1, are then stored as a function of elevation x_{n+1}. Control is then passed to the next lower stage n for further calculations. The next downstream stage is a manhole calculation, which is covered in the following section.

A candidate pipe connection over stage n+1 (i.e., connecting elevations x_{n+1} and x_{n+2}) is considered feasible only if:

1. it has a positive slope
2. the discharge can be carried by the available set of pipe sizes
3. flow velocity is within certain specified minimum and maximum levels

If it does not satisfy the first two constraints, that path is recorded as infeasible. If it violates the velocity constraints, this alternative is given a high penalty cost which makes it unattractive but allows the analysis to continue. This penalty is eventually subtracted out so that final costs are actual costs.

The cost of supplying, excavating and laying the pipe is determined by a functional subroutine CPIPS. This is supplied so as to suit the user's particular situation and calculates the cost per unit foot from the following parameters.

DX (depth of downstream crown (ft or m))

DX1 (depth of upstream crown (ft or m))

DIA (pipe diameter (in. or mm))

for current stage n

PIPTHK (pipe thickness (in. or mm))

If other parameters are required, then Subroutine OBJECS would have to be modified.

B.2 Manhole Calculations

The possible crown elevations at the downstream and upstream ends of the manhole are defined in the same way as with the pipe segments.

If a vertical drop is allowed at a manhole (i.e., when the user sets IDROP=1) it is possible for all x_n elevations to be connected to any equal or higher x_{n+1} elevations. The least cost path to each x_{n+1} elevation is determined for all feasible paths. If no vertical drops are allowed (i.e., the user sets IDROP=0), then x_n may only be connected to that x_{n+1} of the same elevation. That is, there is only one feasible path across the manhole for each x_n value in this case. A third option is for the user to specify a mandatory drop across a manhole through use of parameter FRCDRP. If IDROP=0, all feasible solutions will have a drop of FRCDRP; otherwise, the drop will be FRCDRP plus some multiple of the current DELX value.

The cost of the manhole is calculated by the functional subroutine CMNH. This function is supplied by the user and calculates the manhole cost from the following parameters.

DX - (depth of downstream crown (ft or m))

DX1 - (depth of upstream crown (ft or m))

DIA - (maximum of

- (i) pipe diameters draining into manhole
- (ii) minimum allowable diameter for immediate downstream reach
- (iii) diameter required to take downstream flow at maximum velocity)

Since we do not know the diameter of the pipe draining the manhole, a parameter DRPMNH may be specified. This is basically a second order correction and would normally have a value approximately equal to a pipe size increment.

DRPMNH - (over-excavation of manhole to compensate for not knowing downstream diameter during stage n calculations)

B.3 Refining the Solution

Once the program has calculated the entire family of least cost solutions for the entire network, traceback of the final least cost solution is made. The value of DELX is then halved and if its value is less than or equal to DELXF, calculations are terminated. If not, a new solution space is created for each stage which straddles the last calculated optimal pipe elevation path. This space will be 2 DELX units above and below the optimal path (5-points in all). Elevations violating cover requirements will be neglected. The solution process is then repeated until DELX $\leq$ DELXF. In this way, computer time can be saved by starting with a relatively coarse level of DELX and refining it. Care must be taken that the initial DELX is not too coarse or it may not be possible for the code to find a feasible solution.

C. DATA REQUIRED TO SPECIFY PROBLEM

The data needed to specify the problem in the FORTRAN formatting required for the program are given as follows. The Program may operate in either English or metric units. All measurements are in feet or meters, except for dimensions related to pipe diameter, thickness or overexcavations, which are in inches or millimeters.

C. Global Data

(Note: Ranges are given as a guide only)

CARD 1 (I5)

IECHO

[(=0, no echo print of processed data is required)]

(=1, echo print of processed data is required)

(=-1, echo print of both original and processed data is required)

CARD 2 (8A(10))

Title (An alpha-numeric title may be specified - up to 80 characters; if not, supply a blank card)

CARD 3 (I5)

NMNHS (=the number of manholes in the system; up to 49 allowed for current dimensioning, which is easily increased)

CARD 4 (2F10.4)

DELXI (=the initial increments of the elevation space under each manhole (ft or m))

(suggested range 0.1 to 1.0)

DELXF (=the final increment of the elevation space under each manhole (ft or m))

CARD 4A (I5)

IUNITS (=1 English dimensions used)

(=2 metric dimensions used)

CARD 5 (3I5)

IDROP (=0, no elevation drops allowed at a manhole)

(=1, elevation drops allowed at a manhole)

IVARN (=0, pipefull, fixed 'n', Manning's hydraulic model)

(=1, variable depth, variable 'n', Manning's hydraulic model (ASCE Manual No. 37))

IDIAM (=0, only increasing diameters in downstream direction)

(=1, diameters based on hydraulic requirements only)

CARD 6 (2F10.4)

VMIN (minimum allowable velocity (ft/sec or m/s))

(suggested level, 2.0 ft)

VMAX (maximum allowable velocity (ft/sec or m/s))

(suggested range 8.0 to 10.0 ft)

CARD 7 (F10.4)

CMAN (Manning's coefficient)

CARD 8 (3F10.4)

PIPTHK (nominal pipe thickness used for costing calculations only (in. or mm))

(suggested range 0.0 to 4.0 in.)

DRPMNH (over-excavation of the manhole, which is used in costing calculations only; allows downstream diameter to be slightly larger than any previous upstream diameters since there is some uncertainty at current Stage I as to what the diameter of the exiting pipe will be. The program ensures that the nominal diameter used to calculate manhole costs will accomodate discharge leaving the manhole with a velocity less than VMAX (in. or mm)) (suggested range 0.0 to 6.0 in.)

FRCDRP (mandatory drop across a manhole to account for energy losses etc., used in hydraulic and costing calculations (ft or m)) (suggested range 0.0 to 0.5 ft)

CARD 9 (2F10.4)

XMNCOV (minimum allowable cover to pipe crown (ft or m))

XMXCOV (maximum allowable depth of pipe invert (ft or m))

C.2 Individual Manhole Data (2 cards for each manhole.

The MNHLK(I) values must be in numerical order; starting with 1 and ending with NMNHS)

CARD 10a (4I5,6F10.4)

MNHLK(1) (number of current manhole being considered, i.e., the Ith manhole)

MNHLK(2) (number of 1st upstream manhole connected to the Ith manhole)

MNHLK(3) (number of 2nd upstream manhole connected to the Ith manhole or blank (b))

MNKLK(4) (number of 3rd upstream manhole connected to Ith manhole or blank (b))

Q (the discharge entering the Ith manhole (cfs or m^3s); NOTE: leave blank for the first manhole)

XL (the length of the pipe from the Ith manhole to the next manhole downstream (ft or m); NOTE: leave this blank for the first manhole)

GE (the ground elevation for the Ith manhole (ft or m))

MNDIA (the minimum diameter pipe allowed to drain the Ith manhole (in. or mm). This card restricts the available pipe set. If the following value (MXDIA) is not set, the system sets MXDIA = MNDIA. If left blank (b), the available pipe diameters are not restricted)

MXDIA (the maximum diameter pipe allowed to drain the Ith manhole (in. or mm)

CARD 10b (4F10.4) (if not required, include a blank card)

[Note: the values on Card 10b will override the global specifications of XMXCOV and XMNCOV].

MNELDS (the minimum allowable downstream crown elevation of the pipe draining the manhole (ft or m))

MXELDS (the maximum allowable downstream crown elevation of the pipe draining the manhole (ft or m))

MNELUS (the minimum allowable upstream crown elevation of the pipe(s) entering the manhole (ft or m))

MXELUS (the maximum allowable upstream crown elevation of pipe(s) entering the manhole (ft or m))

NOTE: There will be MNNHS sets of two cards required (2xNMNHS cards in all)

C.3 Pipe Data

The system automatically supplies the following pipe diameter set:

8', 10', 12', 15', 18', 21',24', 27', 30', 36', 42', 54', 60', 66', 72'

This set may be used, or overridden as follows:

CARD 11a (I5)

ND (=0, default diameters used)

(=N, the program will read N diameter values from the next card(s)

(Note: $N \leq 16$))

CARD 11b (8F10.4) (only required if ND > 0)

DIAMS (J) (pipe diameter set, J=1,...,N (in. or mm))

D. COMMENTS ON PROGRAM OPTIONS

D.1 Parameters XMNCOV and XMXCOV

The system is designed such that no pipe crown will be set above the elevation of minimum cover. This may be overridden with the individual manhole data specification (MXELDS and MXELUS).

The system also sets the minimum elevation for the pipe crown. Since the pipe invert will be lower than this, then maximum cover will sometimes be violated. The system automatically checks the invert elevation. If it exceeds the maximum cover, then that pipe receives a cost penalty which makes it an unattractive option but still allows it to be considered as a feasible solution. The maximum crown elevation can be overridden by the individual manhole specification (MNELDS and MNELUS) but the pipe will still attract a penalty if its invert exceeds maximum cover.

The system automatically sets the elevation of the downstream side of the first manhole to maximum cover unless it is overridden by a specific manhole specification (MNELDS). This is done so that the optimal elevation for the pipe or pipes draining into the last manhole can be found. With the last manhole, a drop across the manhole is always allowed unless the upstream elevation for the manhole is restricted to maximum cover.

The cost of a manhole is usually small compared to the cost of the upstream pipe. Thus, the optimal path entering the last manhole will generally be as high as possible, and then drop across the manhole to maximum cover. The correct invert depth for the manhole can then be taken as the elevation of the crown of the lowest pipe entering the manhole minus its diameter. Pipe inverts below maximum cover are flagged in the output with a '#'.

D.2 Parameter IDROP

If IDROP is set equal to 1, drops are allowed across any manhole in the system. It is recommended that the system be operated in this mode, since this flexibility will generally give lower cost solutions.

If IDROP is set equal to 0, no drops are allowed at any manholes except the first one. This will mean that at junctions, all upstream manholes enter at the same crown elevation, whether this is required by the hydraulics or not. This will therefore increase costs.

Manholes with a drop are flagged in the output with a '<'.

D.3 Parameter IVARN

Often, it is desirable to design a pipe that is not flowing full because of slope requirements or because the diameter cannot be reduced in the downstream direction. In this case, the actual velocities in the pipe cannot be calculated on the basis of full pipe flow. The program uses the procedure in ASCE Manual No. 37 to estimate velocity for part full flow and varying Manning 'n' values.

The user may specify if the full pipe or the part full pipe model is used. If the part full model is used and the maximum velocity constraints are violated, a larger diameter pipe is tested to see if it allows the velocity constraint to be satisfied (i.e., for a given discharge and slope, velocity decreases with increasing pipe diameter).

1. Pipe Full Case

The continuity equation is used.

$V = Q/A$

V = full bore velocity (ft/sec)

Q = discharge (ft^3/sec)

A = cross-sectional area of pipe (ft^2)

The actual program statement is

$V = 183.34\ QI/(DIA \times DIA)$

QI = discharge (ft^3/sec)

DIA = pipe diameter (in.)

183.34 = factor included π and conversion from inches ($4x12^2/\pi$)

2. Part-Full Case

The capacity of the pipe at a given slope is calculated using Manning's Equation

$$QF = \frac{1.49}{n} AR^{2/3}S^{1/2}$$

$$= \frac{1.49}{n} \times \frac{\pi D^2}{4} \times \frac{D^{2/3}}{4^{2/3}} \times \frac{S^{1/2}}{12^{8/3}}$$

$$= \frac{1.49\pi}{4^{5/3} x 12^{8/3}} \frac{D^{8/3}S^{1/2}}{n}$$

$$= 0.0006153 \frac{D^{8/3}S^{1/2}}{n}$$

QF = pipe capacity (ft^3/sec)

D = pipe diameter (ms)

S = pipe slope

n = Manning's coefficient

The fraction of capacity used is then calculated.

QR = QI/QF

QR = fraction of capacity used ($R_{Q/QF}$)

QI = design flow (ft^3/sec)

QF = pipe capacity (ft^3/sec)

This value ($R_{Q/QF}$) is then used to enter Figure 24 of ASCE Manual 37 to determine the Ratio of Depth to Diameter $R_{d/D}$ using the discharge curve for variable n. The ratio of part-full velocity to velocity at capacity (Pipe-Full) is then found from the velocity curve for variable n $R_{V/VF}$ using $R_{d/D}$. The part-full velocity is then found by multiplying the ratio $R_{V/VF}$ by the capacity velocity of the pipe VF.

$V = R_{V/VF}\ VF$

VF = QF/A

V = part-full velocity (ft/sec)

VF = velocity in pipe flow full (ft/sec)

A = cross-sectional area of pipe (ft^2)

In the CSUDP/SEWER program, this procedure is approximated by digitizing the curves discharge and velocity curves in Figure 24 of ASCE Manual No. 37 and expressing $R_{V/VF}$ directly as a function of QR. This is accomplished by linear interpolating between the following values of QR (0, 0.025, 0.05, 0.1, 0.2, ..., 0.9, 1.0). In the program, QR, is multiplied by 10.0 and then transformed to an integer (IQ) to select the appropriate values in the array of $R_{V/VF}$ values.

3. Example

Consider the following case.

- Design discharge for a pipe = 2.1 ft^3/sec (Q)

- Capacity discharge for that pipe = 3.0 ft/sec (QF)
- Pipe diameter = 12 in. (D)

Then the pipe full velocity

$$V = 183.34 \times 2.1/12^2$$

$$= 2.67 \text{ ft/sec}$$

The part full velocity is calculated as follows

$$QR = 2.1/3.0 = 0.7$$

Entering Figure 24 gives

$$R_{d/D} = 0.7$$

which in turn gives

$$R_{V/VF} = 0.95$$

The velocity of the pipe flowing to capacity is

$$VF = 183.34 \times 3/12^2 = 3.82 \text{ ft/sec}$$

$$V = 0.95 \times 3.82 = 3.63 \text{ ft/sec}$$

D.4 Parameters MNVEL and MXVEL

A situation may arise where the minimum pipe diameter at the beginning of a branch is so large with respect to the flow that its velocity is less than the allowable minimum velocity. When this happens, a large penalty is attached to this solution which allows the analysis to continue. In all other cases where the velocity violates either the minimum or maximum velocity constraint a much larger penalty is attached which will make this particular solution very unattractive but will still allow the analysis to continue. The user should examine the output to check to see if any constraint violations have occurred and if they are acceptable. If not, the problem may need to be reformulated to give an acceptable solution. To assist the user's understanding of the output, any pipe which has a velocity less than the allowable is flagged with a '*' and any pipe that has a greater velocity is flagged with a

'+'.

E. PROGRAM ORGANIZATION

The following instructions are written on the assumption that the user has access to a remote terminal and will be preparing and editing his data files in an interactive mode. If the user wishes, a batch mode input can be used to establish the required temporary or permanent files.

To use CSUDP-SEWER, the user must supply the previously described data file (called PRDATA), which is a description of the problem, and two functional Subroutines (CPIP and CMNH) which calculate the unit cost of the pipe and the cost of the manhole. The complete package is shown in Figure 2.

Program DATAGN accesses the data file PRDATA and transforms it into the form required by CSUDP. Program CSUDP is a general dynamic programming optimizing routine with its own package of subroutines. Subroutine READIS reads in the data needed for CSUDP and for the Subroutines STATS and OBJECS and reinitializes certain arrays after each elevation increment DELX. STATS is the hydraulic subroutine and calls PIPVEL to calculate pipe velocities. Subroutine OBJECS is the costing subroutine and calls CPIP to calculate pipe costs and CMNH to calculate manhole costs. Subroutine TRBACS determines the optimal path through the system and organizes the printout. Subroutine HYDVAL prepares hydraulic data for the printout.

F. EXAMPLE PROBLEM

Consider the sewer layout in Figure 3, whose characteristics are summarized in Table 1. Assume that the crown elevation of the upstream end of the pipe draining manhole 2 is to be between 91 and 96 ft and have a diameter in the range of 48' - 72'.

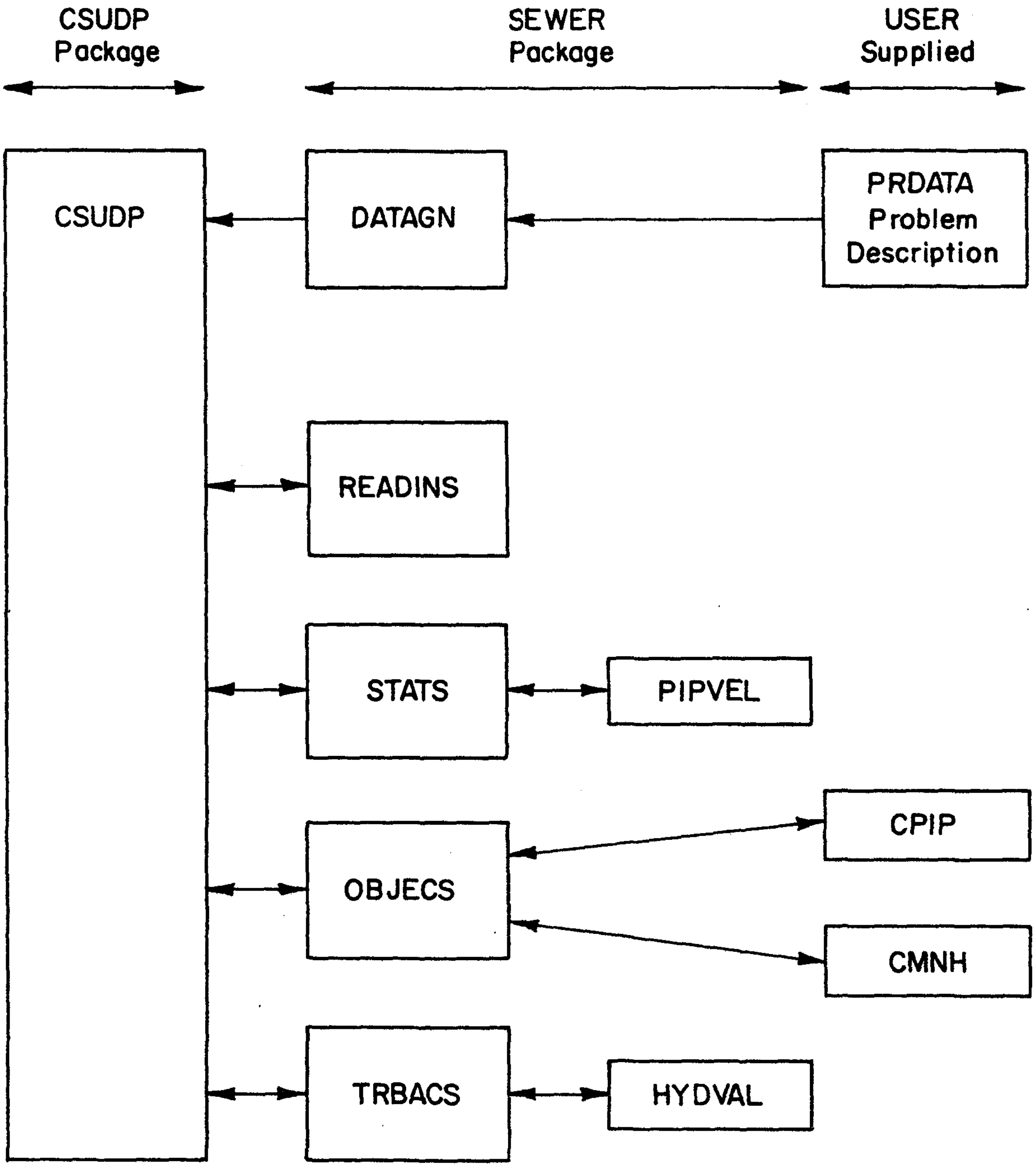


Figure 2. Program Structure.

Table 1. Physical Characteristics of Sample Problem.

Manhole Numbers	Discharge Into Manhole (cfs)	Length of Pipe Downstream of Manhole (ft)	Ground Elevation at Manhole (ft)
1	-	-	100.
2	2.5	125.	102.
3	3.0	150.	103.
4	1.0	125.	105.
5	2.0	100.	106.
6	3.5	75.	108.
7	6.5	100.	109.
8	1.5	100.	103.
9	2.0	125.	105.
10	1.0	125.	109.
11	1.5	75.	110.
12	2.0	150.	107.

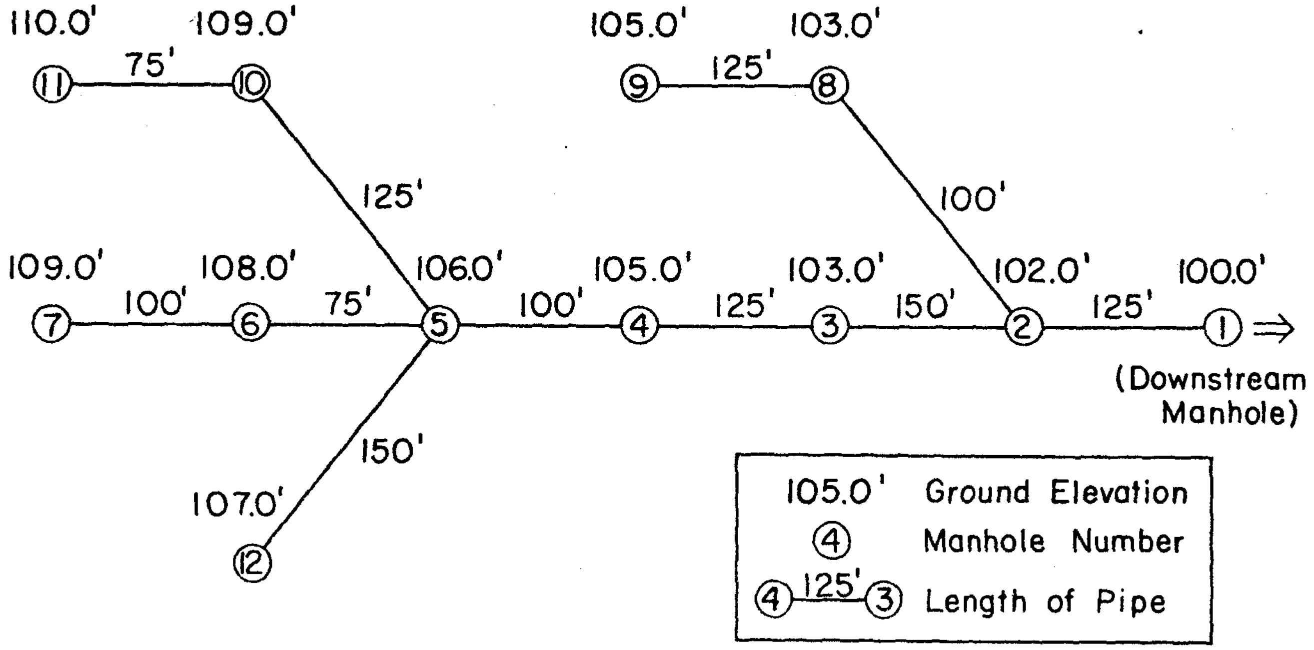


Figure 3. Example layout of a storm sewer system.

Other characteristics of this problem are:

Minimum velocity - 2.0 ft/sec

Maximum velocity - 10.0 ft/sec

Manning's 'n' - 0.013

Minimum cover - 4.0 ft

Maximum cover - 10.0 ft

For the calculations, the initial increment of elevation will be set at 1.0 ft and reduced by 50% until it reaches 0.0625 ft.

Elevation drops will be allowed at manholes, the hydraulic model will assume pipe full Mannings flow, and pipe diameters will not be allowed to decrease in the downstream direction. No drop in crown elevation will be required across each manhole (FRCDRP = 0). Both DRPMNH and PIPTHK will be set to zero (for details, see discussion of cost functions). The data setup and user supplied subroutines are given on the following pages. The solution output is given in Table 2 of Appendix C.

```
/JOB
TC50,T25,PR75.
USER,EQRGWJL,AAWR.
ROUTE,OUTPUT,DC=PR,UN=AD,DEF.
GET,BDATAGN.
GET,TAPE5=PRDATA.
BDATAGN.
REPLACE,TAPE7=PRDATAP.
GET,CPIP.
FTN,I=CPIP,B=BCPIP,OPT=2,R=3.
REPLACE,BCPIP.
GET,CMNH.
FTN,I=CMNH,B=BCMNH,OPT=2,R=3.
REPLACE,BCMNH.
RETURN,BDATAGN.
GET,LGO=BCSUDP.
GET,BSEWER,BCPIP,BCMNH,PRDATAP.
LDSET,PRESET=NGINF,MAP=BS/OUTPUT.
LOAD,BSEWER,BCPIP,BCMNH.
LGO,PRDATAP.
DAYFILE.
EXIT.
DAYFILE.
/EOF
```

```
      FUNCTION CMNH(DX,DX1,DIA,DRPMNW)

      D12=(DIA + DRPMNH)/12
      DXD=DX + D12
      DXD1=DX1 + D12
      DMX=AMAX1(DXD,DXD1)
      CMNH=250. + DMX*DMX

      RETURN
      END
```

```
      FUNCTION CPIP(DX,DX1,DIA,PIPTHK)

      HBAR=(DX+DX1)/2 + (DIA+PIPTHK)/12
      IF (DIA.GT.36.0001) GO TO 10
      IF (HBAR.GT.10.0) GO TO 20
      CPIP=13.0+.8*(HBAR-10.0)+.915*(DIA-12.0)
      GO TO 30
   20 CONTINUE
      CPIP=13.0+(1.67+.042*(DIA-12.0))*(HBAR-10.0)+.915*(DIA-12.0)
      GO TO 30
   10 CONTINUE
      CPIP=128.0+4.9*(HBAR-11.0)+2.5*(DIA-72.0)
   30 CONTINUE

      RETURN
      END
```

IBM FORTRAN Coding Form

GX28-7327-6 U/M 050**
Printed in U.S.A

PROGRAM PRDATA for TYPICAL PROBLEM.
PROGRAMMER
DATE
PUNCHING INSTRUCTIONS GRAPHIC PUNCH
PAGE 1 OF
CARD ELECTRO NUMBER

STATEMENT NUMBER | CONT | FORTRAN STATEMENT | IDENTIFICATION SEQUENCE

```
   -1
TYPICAL PROBLEM
   12
    1.0        .0625
    1    0    0
    2.0      10.0
    0.013
    0.0      0.0       0.0
    4.0      10.0
    1    2              0.        0.      100.
    2    3    8         2.5     125.      102.        48.        72.
   91.       96.
    3    4              3.      150.      103.
    4    5              1.      125.      105.
    5    6   10   12    2.      100.      106.
    6    7              3.5      75.      108.
    7                   6.5     100.      109.
    8    9              1.5     100.      103.
```

IBM FORTRAN Coding Form

GX28-7327-6 U/M 050**
Printed in U.S.A

PROGRAM TYPROB
PROGRAMMER
DATE
PUNCHING INSTRUCTIONS GRAPHIC PUNCH
PAGE 2 OF
CARD ELECTRO NUMBER

STATEMENT NUMBER | CONT | FORTRAN STATEMENT | IDENTIFICATION SEQUENCE

```
    9                   2.      125.      105.
   10   11              1.      125.      109.
   11                   1.5      75.      110.
   12                   2.0     150.      107.
    0
EOR
```

```
   -1
 TYPICAL PROBLEM
   12
    1.0          .0625
    1    0    0
    2.0      10.0
    0.013
    0.0       0.0       0.0
    4.0      10.0
    1    2              0.        0.      100.
    2    3     8        2.5     125.      102.       48.       72.
   91.       96.
    3    4              3.      150.      103.
    4    5              1.      125.      105.
    5    6    10   12   2.      100.      106.
    6    7              3.5      75.      108.
    7                   6.5     100.      109.
    8    9              1.5     100.      103.
    9                   2.      125.      105.
   10   11              1.      125.      109.
   11                   1.5      75.      110.
   12                   2.0     150.      107.
    0
/EOR
```