Civil Engineering Systems Laboratory
Department of Civil Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

General ICES Information

This short report describes what ICES (Integrated Civil Engineering System) is, how it can be used, and where it can be obtained. Since it is an extremely large system, it is impossible to completely describe the various component parts of the system here. The reader who is interested in more detailed descriptions should order the ICES publications listed at the end of this report.

ICES was originally conceived in 1964 as a computer-based system for Civil Engineering practice and teaching. The ICES research and development effort was undertaken by the Civil Engineering Systems Laboratory (CESL) at MIT to advance the use of computers in the Civil Engineering profession. Although ICES was originally intended for use primarily in Civil Engineering, it can also be used in other branches of engineering, science, and management.

ICES is a cooperative project that was developed jointly by CESL and members of the professional community. The ICES research and development work, performed as sponsored research by CESL, was supported by the following organizations:

Ford Foundation (through support of the MIT Inter-American Program in Civil Engineering)

IBM Corporation

MIT Lincoln Laboratories

Massachusetts Bay Transportation Authority

McDonnell Automation Company

Massachusetts Department of Public Works (in cooperation with U.S. Bureau of Public Roads)

National Science Foundation

Portland Cement Association

U.S. Department of Transportation

The following organizations supplied manpower to work on ICES and cooperated in ICES field evaluation:

Penn State University

Skidmore Owings and Merrill

Wisconsin State Highway Commission

The ICES effort at CESL is being pursued as two parallel projects which may be identified as the Research Project (or ICES/Research) and the Development Project (or ICES/Development). ICES/Research is an effort to advance the use of computers in Civil Engineering through theoretical, applied, and experimental research. ICES/Research is generally not oriented toward a specific computer. Research results are communicated to the profession through the sponsors who funded the research and through professional publications issued in accordance with the terms of the agreement with the individual sponsors. The results are reported in research reports, professional papers, articles, books, theses, and other types of documents.

ICES/Development is the development of a fully operational ICES system on a conventional computer. The initial version of ICES, produced as part of the ICES/Development work, is designed to run on the IBM System 360 under Operating System 360 (OS/360). The minimum configuration required to run ICES is a 360/40 or above that operates under OS/360 with at least 128k bytes of core, two 2311 disk drives (or similar random access secondary storage devices), a card reader, card punch, and printer. ICES/360 is a non-proprietary system which may be obtained from the IBM Program Library.

ICES was designed as a machine independent system as part of the ICES/Research activities. In order to implement ICES on a specific computer, the System 360, certain machine dependent programming was required (less than 10% of the total programming effort). The machine dependent programming work was funded completely with support from IBM.

It is expected that ICES will be implemented on other computers. Experiences with past systems developed by CESL, such as COGO, STRESS, and DTM have shown that a single system developed by CESL will eventually result in many systems on different computers. However, it is still too early to predict when ICES will be implemented on other computers.

The initial version of ICES is oriented toward batch processing operations. Research is now underway and prototype experimental systems have been developed for remote processing and time sharing versions of ICES and incorporation of plotting and graphics capabilities.

An Overview of ICES

ICES is made up of two components. These are 1) the ICES System and 2) the various ICES Subsystems. The relationship between these components is hierarchical in that the subsystems run under the control of the ICES System. Furthermore, subsystems are developed using facilities provided in the ICES System.

Each of the subsystems is associated with a particular application area. For example, the STRUDL subsystem of ICES deals with the structural analysis area, while the COGO subsystem deals with geometrics. The engineer communicates with a subsystem via a problem oriented-language (POL) which allows him to state his problem to the subsystem in familiar and convenient terms. The subsystems are intended for use by engineers, not computer professionals. The use of POLs for engineer-subsystem communication has opened the door to computer usage by many non-computer-oriented engineers.

One should not confuse the programmer who develops an ICES subsystem and the engineer who uses the subsystem. The subsystem developer is usually better versed in computer usage than the engineer. The developer draws upon those capabilities made available by the ICES System to structure a subsystem that will be useful to the engineer.

To summarize, ICES can be used in two ways by an installation. First, the existing ICES subsystems may be used to solve engineering problems. Second, the ICES system may be used to create new subsystems.

First the various ICES subsystems are described. Following that we will discuss the ICES System.

The ICES Subsystems

ICES COGO I
(COordinate GeOmetry)

Applications

The COGO Subsystem of ICES may be used for describing and processing computational and design problems in the area of geometrics. Typical application areas include engineering surveys, highway design, right-of-way, construction layout, surveys, and subdivisions. While the COGO language is based on civil engineering terminology, the system can be applied and adapted to a variety of geometric problems in many branches of engineering and science.

Features

COGO operates on geometric objects and variables which can be named, defined, stored, retrieved, computed, combined, and manipulated by means of a command structured, problem-oriented language. The types of geometric objects which can be manipulated symbolically include points, lines, curves, distances, directions, angles, courses, traverses, alignments, parcels, profiles, and so forth. Data tables are maintained for each basic type of object.

The system is designed to provide a large capacity of stored objects per problem, such as up to 10,000 points, 1000 lines, 1000 curves, and so forth. All computational subroutines are double precision. Basic data items may be expressed in a variety of forms, such as numeric, stored, or computed. All output is under command control and is fully labeled. Extensive error checking is provided. The system can be tailored to local conventions via external commands.

ICES COGO is based in part on the concepts and principles of the earlier COGO systems. However, it is a completely new language and is much more powerful and complete, exploiting the new capabilities represented by the ICES systems programs and third generation hardware. It is designed for natural extension to graphical input/output and is applicable to batch, remote computing, and time-sharing environments.

ICES STRUDL I
(The STRUctural Design Language)

Applications

The STRUDL subsystem of ICES is designed as a structural information system, to assist the engineer throughout his design process. STRUDL is designed for application to a wide range of structural types, both two and three dimensional structures consisting of truss, frame, and continuous finite elements. Any combination of these components may be used with a variety of analysis and design procedures.

Features

Using a command structured language easily understandable to the structural engineer, STRUDL allows the user to specify both data and procedures to operate upon this data. Information about a structure may be stored, retrieved, and manipulated by means of commands. Data may be subset in order to apply only part of the information to a procedure and may be output at any time.

The first version of STRUDL is applicable only to framed structures. Data may be specified for members, joints, and loading conditions, with arbitrary naming of these components. Data may be modified continually at any point in a problem. Analysis capabilities include an indeterminate analysis, a preliminary analysis using a set of given force assumptions, and a determinate analysis which does not require component properties. In addition to input data, output may consist of gross forces and deformations on any part of the structure. Forces and stresses within members and a combination of results such as maximum stresses and force and stress envelopes may be calculated and output. Results from different loading conditions and analyses may be combined to produce new results. The problem status may be saved on a disk dile and reused for continuing design problems.

A large variety of problem data may be specified. Information on members include their properties, incidences, and force releases. Properties may be specified for prismatic or variable segment members providing either the properties explicitly or by naming a handbook table and section name within a table generated with the TABLE subsystem. Stiffness or flexibility matrices for members may also be specified. Joint information includes geometry, support conditions, and force releases. Loads within a loading condition may consist of uniform, linear, or concentrated member loads, uniform or concentrated member distortions and temperature loads, and joint loads and joint displacements.

ICES TABLE I
(An ICES File Storage Subsystem)

Applications

ICES TABLE I is a subsystem of ICES which may be used to create and manipulate tabular data. It is designed to provide great flexibility in the specification of data which may be used in conjunction with any ICES subsystem. TABLE creates disk files in designated data sets and provides other subsystems with programmed functions useful for transmitting tabulated information to its problem data base.

Features

With TABLE, a user may input, update, modify, and subset tables structured as two dimensional arrays of real data. An arbitrary number of named rows and named columns may be specified. Dimensional units may be specified for each column, thus allowing arbitrary and variable input units with conversion to a common set of units. New rows and columns may be specified by a user at any time. Rows and columns may also be deleted and data values changed.

TABLE also allows the user to create a number of named orders based on a sequence of one of the properties in the table. Data stored with the orders may be the data upon which the sequence is based or any other column in the table. These orders are useful for the sequencing of design trials when the order represents a decreasing value of the design merit function or criterion.

TABLE supplies a subsystem programmer with an extensive documented set of retrieval functions with which he may assess data in any table. TABLE, row, column, and order names suffice for retrieval. One function also allows for the search of order data for a first trial selection.

TABLE may operate on both the subsystem and user data sets. A subsystem data set may be password protected in order to provide file security. A password protected file may not be modified without the specification of the correct password.

ICES SEPOL I
(SEttlement Problem Oriented Language)

Applications

ICES/360 SEPOL I is an ICES/360 subsystem for the calculation of stresses and strains in the soil due to shallow foundation loadings, and the magnitudes and progress of settlement of shallow foundations. Thus it may be used to perform settlement calculations for the design of buildings, earth dams, highway embankments, and other foundations which cause spread loadings at or near the surface of the soil.

Features

ICES/360 SEPOL segments settlement analysis into four problem areas. Each of these areas is called a function. The four functions are:

1) INSITU - A function which describes the soil system in its natural state, prior to construction.

2) STRESS DISTRIBUTION - A function which describes the loading configuration and leads to the calculation of stress, strain, and/or displacement components caused by the loading.

3) SETTLEMENT - A function which leads to the calculation of the magnitude of settlement at any point on the surface.

4) RATE - A function which leads to the calculation of the progress of settlement and of the excess pore pressure dissipation.

Soil properties can be specified in the form they usually are derived from laboratory data and can be altered during calculation. Any number of layers having different properties can be specified. The stress distribution can be calculated at any point for a distribution of vertical loadings of the same geometry over the surface of an elastic half-space. The magnitude of the settlement can be computed in several well-known ways. The settlement can be computed for both initial and consolidation deformation for any desired layer. The time dependent settlement may be calculated for a system of layers with different permeabilities and compressibilities and with the loading also varying with time. Time-settlement prediction is based on the Terzaghi one-dimensional theory.

ICES ROADS I
(ROadway Analysis and Design System)

Applications

ROADS is used in the solution of problems involving the location and design of almost any type of roadway. Although intended for use primarily in connection with all classes of highways, the basic system framework and a majority of the computer routines are applicable to a wide range of civil engineering problems requiring the excavation and embankment of materials, including railroads, waterways, and dykes. The system is applicable to a project as soon as a general location corridor has been defined and can be used throughout the resulting design process, including the calculations associated with final geometric design.

Features

Major phases of ROADS include a classification of surface and subsurface materials; alignment and profile geometry integration with ICES COGO; flexible roadway finish grade and subgrade links and any number and combination of side slopes and benches for each material layer; roadway template change capability, including changes to the standard roadway design procedure; and the simulation of individual vehicles to determine the effects of roadway geometry and traffic on vehicle performance.

Surface and subsurface material layers are described independently and referenced to a common baseline. Horizontal alignments can be different than the terrain baseline so that multiple alignments or alignment modifications can be easily studied using the same set of terrain data. Changes to the basic cross section during the roadway design and material volumes calculations include changing any part of the roadway or side slope descriptions, changing to a new typical section, superelevation, specifying independent ditch profiles, forcing a material balance, and ignoring selected stations.

Input is in the form of a flexible command-structured engineering language. A problem can be solved incrementally with the choice of commands and operations based on the actual problem being solved and on results of previous design and input calculations. Selective output requests control the generation of tabular reports and character printer plots so that only desired output need be obtained. Capability is provided to permanently save on direct access secondary storage such information files as the multi-material and as-designed roadway cross sections, alignments, profiles, typical roadway sections, and traffic data. Provision is made for transfer of data between

ROADS and COGO so that geometric data can be utilized in both subsystems.

ROADS is based on the concepts and experience of the Digital Terrain Model (DTM) Location and Design Systems, the original COGO System, and the Vehicle Simulation and Operating Cost System. Many of the features and capabilities are analogous to ones existing in these systems. ROADS is nevertheless a new system. Significant increases in engineering capability are provided by combining location, design, and traffic; by developing an information storage and retrieval structure for the system, thereby providing a natural framework for an engineering information system; and by using an engineer-oriented input language.

ICES TRANSET I
(TRANSportation NETwork Analysis)

Applications

The TRANSET Subsystem of ICES is designed to give the transportation planner a flexible and convenient method for predicting and analyzing the flows occurring in transportation networks. Many portions of the transportation planning process may be conducted with the aid of TRANSET. The TRANSET language is designed to analyze typical transportation networks representing highways or mass transportation facilities. However, the system can be applied to other types of networks, such as pipeline systems and electrical networks.

Features

The capabilities made available to the user of TRANSET can be separated into two groups. First of all, transportation planning capabilities are provided. Minimum path trees can be computed and interzonal trips can be assigned. Secondly, and perhaps of equal importance, file manipulation capabilities are provided by TRANSET. Large transportation network descriptions and large interzonal trip matrices can be stored in permanent secondary storage files, modified as desired, used as input to the network analysis procedures, and summarized into reports useful to the transportation planner.

The problem sizes which can be handled by TRANSET are not limited by the size of primary memory since direct access storage devices are treated as logical extensions to primary memory, through the use of the ICES dynamic memory allocator. However, primary memory size has a direct effect on the amount of time it would take to process various sized problems. Execution times increase substantially for problems for which the network data cannot be stored entirely in primary memory, due to the added time needed to store and retrieve data in secondary storage. Programming, rather than core size, limitations restrict the number of network links to 8191.

ICES TRANSET is based on previous minimum path and traffic assignment research and programming conducted at MIT. The methods of assignment available in earlier programs has been expanded to provide a more flexible system. A wide range of input, output, and data file options made possible by the ICES system programs and third generation hardware have been added.

ICES BRIDGE I
(BRIDGE Design System)

Applications

The ICES BRIDGE I Design system is a design tool for bridge engineers. It is used as a decision-making aid in 1) the determination of bridge span arrangements, 2) concrete deck design, and 3) the preliminary design of simple or continuous span girders.

Features

The ICES BRIDGE Design system is a design tool for bridge engineers and is used as a decision-making aid in the solution of bridge engineering problems. The bridge engineer uses a problem-oriented language to communicate with the machine. The initial BRIDGE system allows the engineer to describe the following stages of the bridge design process.

> The horizontal alignment, the vertical alignment, and the roadway cross-section of the proposed bridge. Combined with the intersection and the geometric description of the roadway (or terrain) requiring the bridge separation, alternate span arrangements that are feasible may be determined and then displayed.
>
> The design (or analysis) of a concrete bridge deck, with applicable sidewalk and appurtenances, supported by a stringer girder system. Deck design constraints may be given by the engineer to override an automatic deck design. The bridge deck geometry, loading configuration, deck design and design stresses may be displayed.
>
> The preliminary design of a simple or continuous span variable moment of inertia girder. The girder design logic and display requests for moment, or shear, or reaction design values are specified by the engineer using a design directive language. The girder geometric configuration and section modulus requirements may be determined prior to the specification of section material and element dimension for a final girder design.

The bridge design project may require the referencing of one or more of the above stages. The normal referencing sequence is as indicated where each stage (after the first) is highly dependent on information generated from the previous stage. If this sequence is altered, or only one design block is entered, those parameter values normally generated from a previous stage must originate from commands given by the engineer.

The bridge design system command structure allows the engineer to participate in and direct the problem solution of a bridge project. The command structure is in the same format and terminology that the bridge engineer would discuss (or direct) with a colleague the design stages of a bridge project.

ICES PROJECT I
(PROJect Engineering ConTrol)

Applications

PROJECT is a computer system created to assist in the planning and control of projects. PROJECT accepts, processes, and saves information supplied by the Engineer-User by means of a language, a set of programs, and a set of information files. Every job that can be represented by a network of work items and associated interdependencies can be processed by PROJECT. The meanings of the commands are easily understandable to any user familiar with project management techniques.

Features

The heart of PROJECT is a CPM network analysis system capable of processing three types of networks: activity-on-node, activity-on-arrow, and event-oriented. The user may specify external timing constraints, a calendar reference point, and non-working dates to be used in the calculation of the Critical Path schedule for the project. Whenever necessary, the user may refer to a point in time with either calendar dates or working days.

PROJECT permits the processing and storage of any number of independent job networks during a single computer run. Once an individual network has been input, the information remains on the disk files as an active network until it is deleted by the user. Additions, deletions, and modifications to the data for any active network may be specified at any time. The user has the ability to output any subsets of computed network data he desires. In addition, projects can be subdivided into unique or overlapping groups for output purposes. Certain schedules can be output in bar chart form, in addition to the normal tabular form.

Although PROJECT is primarily a scheduling system, several financial and resource usage capabilities have been incorporated. These features enable the user to assign costs for activities either directly or through the extension of resource consumption estimates. Both graphical and tabular presentations of predicted costs and scheduled resource consumption can be obtained.

In PROJECT I, a new approach known as repetitive dependent sequences has been developed for the treatment of logistics which are normally represented as network activities when they are considered at all. Generalized sequences of logistic procedures are formulated by the user and used to schedule the preliminary operations of any applicable items required by the activities of the project network.

The ICES System

Externally each of the ICES subsystems consists of a set of problem-oriented language commands which enable an engineer to communicate his problem solving requests to the computer. Internally, each subsystem consists of a series of computer programs to perform the engineering operations requested in the problem-oriented language requests. These operations are performed on engineering data which are internally represented using some computer data structure. The basic computer capabilities provided in ICES are concerned with the external problem-oriented language commands and the internal computer programs and data structures. These capabilities include:

1) A command definition language (CDL) to enable a programmer to specify the structure and processing of problem-oriented language commands.

2) An engineering programming language (ICETRAN) to enable a programmer to write the computer programs which perform the engineering operations.

3) Internal computer data structures specifically designed for the representation of engineering information.

The above capabilities are used by programmers to develop engineering subsystems. In addition, other system capabilities are provided to run the developed subsystems. These include:

1) A command interpreter to interpret each command input by the engineer, and to supervise the processing of the command.

2) Dynamic program structures to dynamically link together the engineering programs used in the processing of the command.

3) Dynamic memory allocation to dynamically organize the data and programs in primary memory based on the current processing requirements.

4) Data management to organize and coordinate files of engineering data stored on secondary storage.

ICES Operating System and ICES System Programs

The computer capabilities which enable programmers to create, modify, and run subsystems are referred to as system programs. Each computer manufacturer generally supplies a set of basic system programs which comprise a computer operating system. These system programs provide many of the computer capabilities needed to develop and run an engineering system such as ICES. However, certain important system programs specifically needed for an engineering computer system are not included. To provide the total environment needed for engineering computer system development and operation, special ICES system programming capabilities have been developed. These ICES system programs represent an expansion of and addition to the normal system programs provided by a computer manufacturer. The net result of the combination of the manufacturer-supplied system programs and ICES system programs is an operating system specifically designed for engineering computer systems.

This one engineering operating system is used by all the ICES subsystems. It is, therefore, quite easy for each organization to use the system programs provided in ICES to modify existing ICES subsystems and incorporate new subsystems. Whereas in the past, development of engineering computer systems first required the development of complex system programming capabilities, ICES now relieves the programmer of this difficult, time-consuming and expensive task.

ICES Problem-Oriented Languages

The ICES Command-Definition Language (CDL) and the Command Interpreter (CI), together provide the mechanism for the development of ICES problem-oriented languages. These problem-oriented languages provide the engineering user with the problem-solving capability of the ICES system without requiring him to learn a new computer-oriented language. Each ICES subsystem solves problems in a particular civil engineering discipline; corresponding to the particular discipline there is an appropriate language used by the engineer to specify his problems. These languages are designed to be sufficiently similar to the engineer's natural language to allow him to use them without special training and, most important, without putting artificial restraints on his work. For example, the Structural Design Language (STRUDL) includes vocabulary such as MEMBER, JOINT, LOAD, FORCE, MOMENT, etc.

In the past, the development of such languages required special competence in system programming, often to a degree found only in a computer professional. CDL and the CI eliminate the need for systems programming in the development of ICES subsystems, allowing application professionals with a knowledge of computer programming to develop quite general user languages. This is appropriate as engineers, not computer professionals, should develop languages for engineers to use.

CDL is a problem-oriented language for the specification of problem-oriented languages. The subsystem programmer uses it to describe his problem-oriented language in the same manner as the engineer uses a problem-oriented language to describe an engineering situation. The output of CDL is a dictionary and set of rules which are used by the CI in translating the new language. Once a language has been defined by CDL, it can be easily changed and extended. It is expected that a class of users will want to customize ICES subsystems to better match their needs. CDL makes this possible; the modular programming of ICES subsystems makes it easy.

ICETRAN Programming Language

An engineer uses problem-oriented language commands to ascertain that particular engineering operations be performed by the internal computer programs. Typically, these programs, which perform the requested operations, are written using a procedure-oriented programming language such as FORTRAN. Although FORTRAN incorporates many excellent capabilities, it does not include all programming operations that are required for engineering programming, and does not incorporate many capabilities provided in "third generation" computer operating systems.

Historically programmers have extended the capabilities of FORTRAN through the use of assembly language subroutines and the FORTRAN CALL statement. Though such a method works, the programmer becomes burdened with the problem of knowing the rules for constructing calling sequences to the multiplicity of routines necessary for running in an ICES-type environment. Moreover, the complexity of such calling sequences tends to obscure the logically simple processes which they describe, as well as creating simple programming errors.

ICETRAN (ICES-FORTRAN) minimizes these problems, At its simplest level, ICETRAN is FORTRAN: a programmer familiar with FORTRAN can use it directly. When he wishes to utilize extra ICES features such as dynamic data structures, dynamic program structures and data management, he can use simple FORTRAN-like statements instead of complicated calling sequences.

An ICES system program called a precompiler, pre-processes each ICETRAN program. FORTRAN statements within the program are not modified, whereas the new ICETRAN statements are translated into CALL's to the ICES system programs which will perform the requested operations during program execution. The precompiler, therefore, accepts an ICETRAN program as input and produces an equivalent FORTRAN program which can then be processed by a conventional FORTRAN compiler.

ICETRAN enables a FORTRAN-level programmer to use new system programming capabilities without requiring a knowledge of how these system programs operate or how they are called. ICETRAN can be viewed as an extension of FORTRAN, specifically designed for engineering programming.

ICES Program Structure

Program structures in ICES are designed to give the subsystem programmer the capability for treating programs in an appropriate manner for engineering problem solving. Each ICES subsystem is composed of a great many programs, the total size of which is typically many times the size of the primary memory of the computer. Although the total size of all programs is quite large, only a small subset of these programs are required at any one time during execution. The program requirements are governed for the most part indirectly by the engineer whose problem-oriented language commands imply the execution of a particular set of programs.

With the above in mind, program structures in ICES do not require that all subsystem programs be in primary memory throughout the run. Rather, the subsystem programmer, by using the ICETRAN dynamic program statements (LINK, TRANSFER, etc.) can request a program to be loaded into primary memory from secondary storage at any time during execution. If the program is already in primary memory at this point, this in-core copy is used. The programmer is not aware of the physical location of his program when he requests it. It may be on secondary storage or in primary memory: there is no logical difference to the programmer. This implies that the programmer does not need to know the 'history' of the process (what programs have already been used). Also, programs that are no longer needed will be removed from core, if space is required for some other entity (another program or dynamic data).

The ICES program structure capabilities enable a subsystem programmer to dynamically modify the flow of computer process. He can specify and change the names of the programs he wishes to use at execution time (as opposed to compilation time). For example, if the rate of convergence of an iterative process is slow, the programmer can, using the same linkage statement, switch to a different set of programs incorporating a more appropriate iteration technique.

ICES program structure provides a very flexible dynamic control of the computer process. This flexible dynamic environment is necessary to reflect the highly complicated, result-dependent engineering process.

ICES Data Structures

Problem-oriented languages enable engineers to specify many different types of problems to the computer. Each of these problems will generally involve different types and amounts of engineering data. The ICES system, therefore, provides data flexibility required for both the engineer who requests engineering operations and the ICETRAN programmer who specifies procedures to execute the engineer's requested operations. Dynamic data treatment allows a programmer to use data logically in a sequence meaningful to his procedures without significant concern about the size or physical location of the data. He may structure his data to suit his procedural needs at the execution time based on the problem-solving requests of the engineer. Dynamic data structures eliminate the constraints present with traditional FORTRAN static-dimensioned-array data structures where the size and structure of the array are specified at compilation time.

With dynamic data structures, the size and structure of arrays are specified at execution time. Both array size and structure can change as problem execution proceeds. Space for arrays is created only when needed. If an array is not needed for a particular problem, no space is needed.

The ICES dynamic data structure capabilities are incorporated in the ICETRAN programming language. ICETRAN statements are provided to allow a programmer to designate, define, release, and destroy dynamic arrays. Referencing to dynamic array elements is programmed using conventional FORTRAN subscripting notations.

A set of ICES system programs perform the requested operations on dynamic arrays. One of the primary functions of these programs is to manage the space allocation of the dynamic arrays. The dynamic data structures incorporate mobility and flexibility so that the system programs can shift the arrays or portions of the arrays within primary memory or between primary memory and secondary storage. Whenever space is needed in primary memory for a new array, and the necessary space is not available, the contents of primary memory are automatically reorganized. During reorganization, arrays residing in primary memory that are not currently needed, are shifted to secondary storage to make room for new arrays. If an array which resides on secondary storage is subsequently referenced, it is brought back into primary memory by the system programs. Within this framework, secondary storage becomes a logical extension of primary memory, so that the size of a problem that can be solved is limited by the capacity of secondary storage rather than the capacity of primary memory.

Dynamic data structures provide the user with the most efficient use of the computer's memory for each particular problem, without placing constraints or limitations upon the problem-solving capacity. This allows him to solve small and large problems with the same system.

ICES Data Management

The ICES data management capabilities are designed to allow the subsystem ICETRAN programmer the use of direct access secondary storage. This use is provided in a flexible manner for the storage and retrieval of data in a direct, as well as sequential, manner. These routines are used by the ICES system programs to store and retrieve system data on secondary storage, and by various subsystems such as STRUDL, BRIDGE, PROJECT, and ROADS, to store and retrieve problem data or handbook and design tables.

ICETRAN statements provide access to system programs which interface with the data management routines of Operating System/360. The basic capability of a data set is extended to allow the user to define files within a data set consisting of one or more logical records. The logical records in a file may be updated, extended, truncated, or entirely deleted. Additional logical records may be added to a file at any time; and/or a portion of any logical record may be read from secondary storage directly. A file may be deleted in its entirety, renamed, or protected against accidental modification.

The ICES data management capability is designed to operate with any direct access secondary storage device. An ICETRAN programmer uses the same data management statements regardless of what type of direct access device is being used. Examples of direct access secondary storage devices include disk, drums, and data cells.

Obtaining ICES

CESL has submitted ICES to the IBM Program Library for distribution as a type IV program. (Type IV program means that it was developed by an organization other than IBM but is distributed by IBM). If you have any questions regarding the ordering of type IV material from the Program Library, consult your IBM representative.

There are currently ten separate ICES packages that can be ordered from IBM. These are:

1) ICES/360 Basic System

2) ICES/360 Basic System and Language Processors

3) The 8 subsystems of ICES-I

The first two packages are ICES System packages as opposed to the last eight which are subsystem packages. In order to run any subsystem, either ICES/360 Basic System or ICES/360 Basic System and Language Processors is required. If the installation is intending only to run existing subsystems and never to create new subsystems, then the ICES/360 Basic System should be ordered. On the other hand, if the installation plans to create new subsystems, ICES/360 Basic System and Language Processors is required. Thus, the installation should order either 1) or 2) plus any desired subset in 3). Each of the above packages consists of program material plus some basic descriptive material.[1]

The program material is distributed to the installation on tape. A simple system generation procedure is required to transfer the program material from tape to disk for day to day use. This procedure is discussed in the descriptive material of Packages 1) and 2) above. If the program material requires less than 200 feet of tape, IBM will send a DTR tape (Disposable Tabe Reel) to the organization (no tape need be sent by the organization). If more than 200 feet is required, the installation must send a 2400 tape to IBM, which will be returned with the program material on it. Note that in either case, a 9 track tape will be made.

A User Registration Form is included in the descriptive material for each package. We strongly recommend that this form be completed and returned to CESL for each package that the installation intends to make use of. From time to time, CESL will distribute information sheets to the user community about its various packages (for example, pitfalls, tips to increase efficiency, updates, etc.). These information sheets will be called TIPS (for Technical Information Poop Sheets). Only those organizations that have registered with CESL as a user of a particular package will receive TIPS about this package from CESL.

---

[1]The distributed descriptive material may be augmented by the purchase of ICES related books and reports described later in this document.

ICES Distribution Packages

Package Name	Order Number	Program Material	Descriptive Material
ICES/360 Basic System	360D 16.2.004	will fit on DTR reel, Basic systems programs, Basic systems secondary storage data files	An Introduction to ICES, ICES Operations Manual, ICES Error Procedures
ICES/360 Basic System and Language Processors	360D 16.2.005	will fit on DTR reel, Basic systems programs, Basic systems secondary storage data files, Language Processors (CDL, ICETRAN, FORTRAN)	An Introduction to ICES, ICES Operations Manual, ICES Error Procedures
ICES/360 COGO I	360D 16.2.006	will not fit on DTR reel[1], COGO programs, COGO system & subsystem data files, COGO sample problem	COGO Operations Manual
ICES/360 STRUDL I	360D 16.2.007	will fit on DTR reel, STRUDL program, STRUDL system and subsystem data files, STRUDL sample problems	ICES STRUDL I Operations Manual, ICES STRUDL I General Description, Abstract of STRUDL Commands, ICES STRUDL I Error Messages, The Use of ICES STRUDL I
ICES/360 TABLE I	360D 16.2.008	will fit on DTR reel, TABLE programs, TABLE system data file, TABLE sample problem	ICES TABLE I Operations Manual, ICES TABLE I Table Retrieval, ICES TABLE I Error Messages

[1] Installation must send 2400 foot tape when ordering this package

Package Name	Order Number	Program Material	Descriptive Material
ICES/360 SEPOL I	360D 16.2.009	will fit on DTR reel, SEPOL programs, SEPOL system data file, SEPOL sample problem	ICES SEPOL I Operations Manual, ICES SEPOL I General Description
ICES/360 ROADS I	360D 16.2.010	will not fit on DTR reel[1], ROADS programs, ROADS system and subsystem data files, ROADS sample problems	ICES ROADS I: A General Description, Part I - Engineering Capabilities, Part II - Example Problem, ICES ROADS I: Generation and Operations Manual
ICES/360 TRANSET I	360D 16.2.011	will fit on DTR reel, TRANSET programs, TRANSET sample problem	TRANSET Operations Manual
ICES/360 BRIDGE I DESIGN SYSTEM	360D 16.2.013	will fit on DTR reel, BRIDGE programs, BRIDGE sample problems, BRIDGE system data file	BRIDGE Operations Manual, BRIDGE Description
ICES/360 PROJECT I	360D 15.4.004	will fit on DTR reel, PROJECT programs, PROJECT sample problems, PROJECT system data file	ICES PROJECT I Operations Manual, ICES PROJECT I General Description, Abstract of PROJECT Commands, ICES PROJECT I Error Messages

[1] Installation must send 2400 foot tape when ordering this package

ICES Publications

In addition to the material obtained from IBM, other ICES publications may be ordered directly from:

ICES Publications
Room 1-163
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

ICES Distribution Publications

The following publications are directly associated with the ICES distribution packages:

An Introduction to ICES

This publication describes the philosophy and uses of ICES. It introduces the basic elements of the ICES system and considers ICES as a component of an information system.

Number of pages: 20 Price: $1.00

ICES System - General Description

This publication is designed for programmers who will develop new ICES subsystems and modify existing ICES subsystem. It describes the components of the basic system and language processors and explains how these components are used. The topics discussed include the development and generation of a problem-oriented language, the ICETRAN programming language, dynamic data structures, dynamic program structures, primary memory management, and secondary storage management. This publication is not a programming guide but rather a text describing basic new concepts incorporated in ICES. Programming details are discussed in a companion publication "ICES Programmers Reference Manual."

Number of pages: 133 Price: $4.00

ICES Programmers Reference Manual

This publication is designed for programmers who will develop new ICES subsystems and/or modify existing subsystems. It assumes the reader is familiar with basic ICES concepts described in the publication "ICES System-General Description." The "ICES Programmers Reference Manual" gives the complete language specifications for both ICETRAN and CDL.

Number of pages: 95 Price: $4.00

Engineer's Guide to ICES COGO I

This publication describes how ICES COGO can be used for geometric problem solving and describes all commands which are contained in the ICES COGO processor. It will serve as reference guide for ICES COGO users.

Number of pages: 117 Price: $5.00 each for order up to 7
$4.00 each for order of 10 or more

Example Problems for ICES COGO I

This manual contains 34 problems with solutions to serve as an aid in learning the language and use of ICES COGO I.

Number of pages: 80 Price: $2.00

ICES STRUDL I - General Description

ICES STRUDL I, the Structural Design Language, is an ICES subsystem for general structural engineering problems. This publication describes the purpose of STRUDL and the scope of its first version. Its use is also illustrated with a short simple sample problem. Computer output from this problem illustrates a computer solution.

Number of pages: 22 Price: $1.00

ICES STRUDL I - Engineering User's Manual

This manual provides complete and detailed instructions for the use of STRUDL to solve structural engineering problems. Capabilities and problem solving procedures are described, primarily in terms of the capabilities of the STRUDL language commands. Detailed descriptions of how commands may be used to input and modify data request analyses and process output, is provided. An extensive example is included which illustrates most of the capabilities of ICES STRUDL I. Appendices provide descriptions of error messages and an abstract of STRUDL commands illustrating allowable syntax.

Number of pages: 190 Price: $5.00

The Use of ICES STRUDL I

This publication illustrates through a series of example problems how the various STRUDL commands might be put together to solve meaningful engineering problems. Examples start with simplest forms of problem solving using a minimum of commands to more complex problems which illustrate more extensive capabilities of STRUDL. Natural computer output is shown to illustrate the results of computer solutions.

Number of pages: 118 Price: $2.00

ICES TABLE I - Engineering User's Manual

TABLE is an ICES file storage subsystem which allows the creation, editing, and subsetting of tabular data files. Row and columns of a table may be identified by arbitrary names. Any number of tables may be created and in any data set. This manual provides detailed instructions on the use of TABLE including the detailed description of all engineering commands necessary to operate on a table. Appendices include a description of all error messages provided by TABLE and a description of how TABLE retrieval functions may be incorporated into any subsystem of ICES and used to retrieve, symbolically, data from arbitrary tables.

Number of pages: 56 Price: $3.00

ICES SEPOL I - General Description

This publication serves as an introduction to the SEPOL subsystem. It contains descriptions of the use of the computer in Soil Engineering, the scope of SEPOL, and the nature of the SEPOL language.

Number of pages: 60 Price: $1.50

ICES SEPOL I - User's Manual

This publication describes how to use SEPOL for problems in settlement analysis. It contains a complete description of all commands and descriptions of the error messages.

Number of pages: 120 Price: $4.00

ICES ROADS I - General Description

This report describes the basic concepts and engineering capabilities of ICES ROADS I and presents a short example problem, including computer output, illustrating the usage of ROADS.

Number of pages: 110 Price: $2.00

ICES ROADS I - Engineer's Reference Manual

This manual is intended to serve as a reference document for users of ICES ROADS and contains complete descriptions of the available commands for each of the major phases of ROADS--multi-material terrain model, alignment and profile geometry, roadway design and material volumes, and vehicle performance. In addition, the principal information files and data tables utilized in the system are described including their contents, capacity, and storage.

Number of pages: 200 Price: $5.00

ICES TRANSET I - Engineering User's Manual

This manual provides complete instructions for the use of TRANSET to solve transportation network analysis problems. It includes a general description of the capabilities provided in TRANSET, a detailed description of the commands which constitute the TRANSET language, and a detailed description of TRANSET error procedures.

Number of pages: 175 Price: $5.00 each for order up to 7
$4.00 each for order of 10 or more

ICES BRIDGE I Design System - General Description and Engineering User's Manual

This is a two-part document. Part one describes the concepts of the design system. The foundation of the language hierarchy, command structure, information files, and the partition of the design system capabilities are developed. Part two is the Engineering User's Manual describing the existing capabilities of the design system. Command description, abstract of commands, and command examples are given for the geometry, deck design, and preliminary girder design capabilities.

Number of pages: 205 Price: $5.00

ICES BRIDGE I Design System. Problem Formulation and Solutions

This document is a set of five example problems illustrating many of the design system capabilities. The problems are formulated using the applicable commands and the problem solutions are a part of this document.

Number of pages: 120 Price: $2.00

ICES PROJECT I - General Description

This publication describes the purpose of PROJECT, the Project Engineering Control language, and describes the scope of the first version. Examples of the versatility of input and processing commands are given to show the ease with which the system may be used. A short example problem listing, including two sample output reports, are given.

Number of pages: 28 Price: $1.50

ICES PROJECT I - Engineering User's Manual

This manual provides complete and detailed instructions for the use of PROJECT for critical path scheduling, cost accounting, and progress reporting of projects. The control commands for data input and scheduling and cost computations are described, including examples and sample output reports. The manual is structured to introduce basic network commands followed by more advanced capabilities, thereby serving as a self-instruction aid. Included with the manual are abstracts of the commands and complete lists of all error messages.

Number of pages: 200 Price: $5.00

ICES System Publications

1) ICES System Design - Daniel Roos, MIT Press, Cambridge, Massachusetts. Price: $10.00. This is a basic background reference and should be ordered directly from the MIT Press by anyone having a serious interest in ICES.

2) Primary Memory Management in ICES: An Engineering Oriented Computer System - J.M. Sussman, CESL Report R67-68, Price: $5.00.

ICES Subsystem Development Primer R68-56

This manual is designed to aid the subsystem developer in programming and implementing new ICES subsystems or in modifying existing subsystems. The case study approach is used to explain the step by step procedures for subsystem developments. Three subsystems are discussed (FJSBEAN, F2JAREA, and GEOMETRY) and complete listings of these three subsystems are included in the manual. This manual serves as a companion document to 'ICES System: General Description' and 'ICES: Programmer's Reference Manual' in that it takes the ICES system components described in the former and the programming steps described in the latter and relates these to subsystem development and implementation.

Number of pages: 175 Price: $5.00