General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

NASA
Technical Memorandum 87033

USAAVSCOM
Technical Report 85-C-4

DEAN: A Program for Dynamic Engine ANalysis

Gerald G. Sadler
Propulsion Laboratory
AVSCOM Research and Technology Laboratories
Lewis Research Center
Cleveland, Ohio

and

Kevin J. Melcher
Lewis Research Center
Cleveland, Ohio

Prepared for the
Twenty-first Joint Propulsion Conference
cosponsored by the AIAA, SAE, and ASME
Monterey, California, July 8–10, 1985

NASA

NASA
Technical Memorandum 87033

USAAVSCOM
Technical Report 85-C-10

DEAN: A Program for Dynamic Engine ANalysis

Gerald G. Sadler
Propulsion Laboratory
AVSCOM Research and Technology Laboratories
Lewis Research Center
Cleveland, Ohio

and

Kevin J. Melcher
Lewis Research Center
Cleveland, Ohio

Prepared for the
Twenty-first Joint Propulsion Conference
cosponsored by the AIAA, SAE, and ASME
Monterey, California, July 8-10, 1985

NASA

DEAN: A PROGRAM FOR DYNAMIC ENGINE ANALYSIS

Gerald G. Sadler*
Propulsion Laboratory
AVSCOM Research and Technology Laboratories
Lewis Research Center
Cleveland, Ohio 44135

and

Kevin J. Melcher
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135

SUMMARY

The Dynamic Engine ANalysis program, DEAN, is a FORTRAN code implemented on the IBM/370 mainframe at NASA Lewis Research Center for digital simulation of turbofan engine dynamics. DEAN is an interactive program which allows the user to simulate engine subsystems as well as a full engine systems with relative ease. The non-linear first order ordinary differential equations which define the engine model may be solved by one of four integration schemes, a second order Runge-Rutta, a fourth order Runge-Kutta, an Adams Predictor-Corrector, or Gear's method for stiff systems. The numerical data generated by the model equations are displayed at specified intervals between which the user may choose to modify various parameters affecting the model equations and transient execution. Following the transient run, versatile graphics capabilities allow close examination of the data.

DEAN's modeling procedure and capabilities are demonstrated in this paper by generating a model of a simple compressor rig.

E-2588

INTRODUCTION

System simulation is a powerful tool in the aerospace industry. These simulations can be used in conjunction with testing of a specific system in an attempt to anticipate possible problems and reduce risk. Also simulations can be utilized in subsystem development, specifically in the design and development of controls for turbofan engines.

These simulations must have several attributes in order to be useful tools. They must be reasonably accurate and computationally efficient.

Currently there are several programs in use that simulate turbofan engines. NNEP (ref. 1) is a generalized code for steady-state analysis of arbitrary engine configurations. Another code, DYNGEN (ref. 2) simulates the dynamics of specific engine configurations. A complex hybrid (analog/digital) computer simulation, HYDES (ref. 3) simulates dual shaft turbofan engines with up to three air streams. These simulations are limited as to their complexity

*Currently with NASA Lewis Research Center.

because the hybrid computer has hardware and memory limitations. DIGTEM (ref. 4), a recently written code allows steady-state as well as transient calculations to be performed. DIGTEM has the flexibility to model various engine configurations which are subsets of a two-spool, two stream turbofan. DIGTEM's interactive capability is limited and turbofan engine components cannot be developed separately. None of the above packages have the extensive flexible graphics provided in DEAN.

The Dynamic Engine ANalysis program, DEAN has been written specifically to model two-spool, two-stream turbofan engines and turbofan engine components. However, DEAN can be easily reconfigured to model other dynamic systems such as turboshaft engines, internal combustion engines, or electrical power systems. User interaction, modular engine component routines, various integration methods, and extensive graphics have been incorporated into the software to enhance the simulation capability of DEAN.

Implemented under the TSS operating system on an IBM/370 mainframe computer, DEAN was written with the intent of simulating dynamic systems interactively. The program utilizes the IBM/370's interactive debugging capabilities, graphics libraries, and the IMSL FORTRAN library which allows the user to develop models on-line in the sense of finding modeling errors and initiating transients. The interactive capability also allows the user to analyze system response to external inputs such as perturbations in fuel flow, nozzle area, and bleed flows. Temperatures, pressures, flows, and other modeling parameters otherwise known as state variables are displayed continuously during the simulation. Also, part of DEAN's user interaction includes its user friendliness which is achieved by using menus with logical default selections during program execution.

There is a natural division of components in a turbofan engine, i.e., inlet, fan, compressor, combustor, high pressure turbine, low pressure turbine, afterburner, nozzle, bypass duct, and engine control. DEAN is designed to simulate a turbofan engine with a set of modular routines, one for each engine component. The user may use the provided component models for his simulation or he may choose to provide/develop his own model for any or all of the components.

Generally the models are based on first principles, i.e., conservation of mass, momentum, and energy, and can be formulated as a system of first order nonlinear differential equations. DEAN uses these equations and one of several numerical integration techniques to approximate the system response. The integration technique may be selected by the user which best approximates the system being simulated. The appropriate use of the available integration schemes will be discussed later.

DEAN supports an extensive graphics capability for displaying results of a simulation. This capability includes the ability to plot any parameter, including time, versus any other parameter, superposition of performance maps on a plot, calculation of new parameters based on existing parameters, the option of having one or two plots per page, extensive labeling including date, time, and confidentiality of data, ability to plot the data for time subintervals, and the capability to save the data for later analysis. The plots generated by DEAN are of report quality.

There are several support packages required to model turbofan engines. Generally the engine performance maps are bivariate in nature, i.e., compressor pressure ratio is a function of corrected speed and corrected flow. Also various thermodynamic properties are required such as C_p (air and fuel), C_v, and γ, the ratio of specific heats, all of which are single valued functions. Therefore subroutines have been supplied to interpolate single and bivariate maps. Also there are subroutines for reading free formated datasets and a subroutine is included for simple proportional/integral control.

DEAN STRUCTURE

DEAN was designed to be a modular program. Subroutines were written to do specific tasks such as prompting the user for a steady-state point or running a transient simulation. This manner yields several advantages. DEAN can be expanded and modified easily because only specific modules may need minor changes. Also, the modularity in the program expedited the software development phase. Further, the programming flow of DEAN is more easily understood. Thus DEAN is a highly structured program which allows the user to easily identify and modify components.

DEAN consists of 46 subroutines that are called by one main program. Since there are so many subroutines, only the more important ones will be discussed here. Figure 1 shows a simplified block diagram of the subroutine interaction in DEAN.

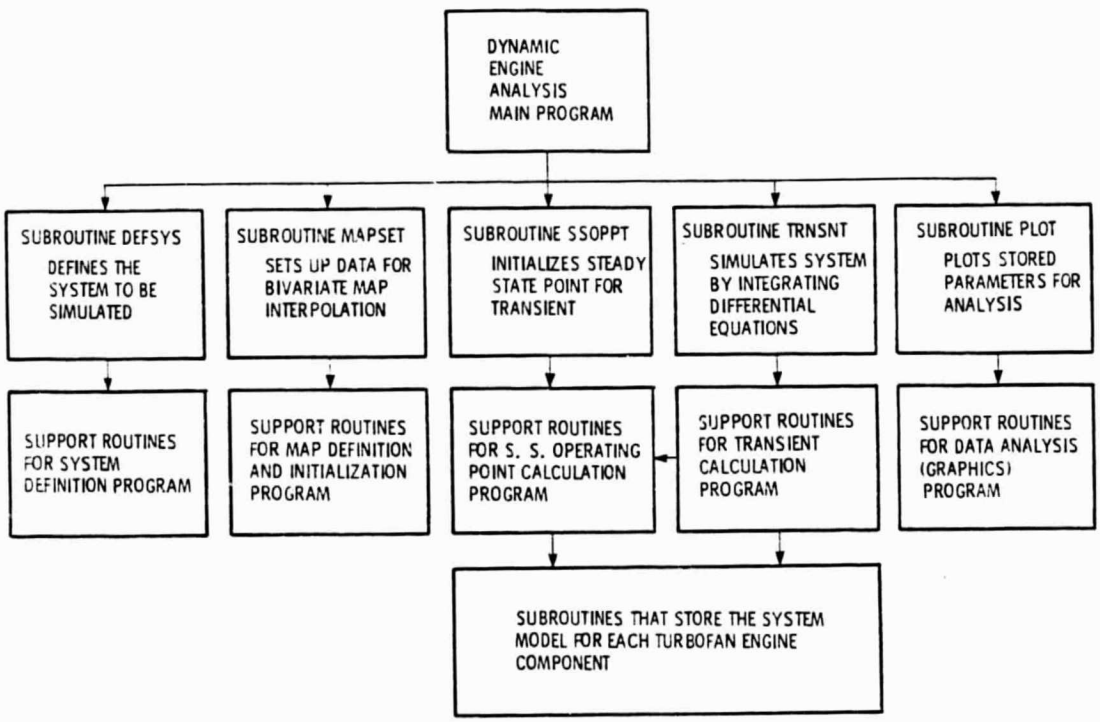

Figure 1. - Simplified diagram of Dean subroutines.

The main program (DEAN) is responsible for controlling the entire program. It presents the user with the following options: (1) system definition, (2) map definition, (3) steady-state operating point calculations, (4) transient simulation, (5) data analysis (graphics), (6) termination of DEAN - and prompts for a response. When the user has entered an appropriate response, the program calls the subroutine associated with that response.

Subroutine DEFSYS is prompts the user for the components of the system to be simulated. By using a menu of components (table 1.), DEFSYS allows the user to define a system for simulation. The order of these components corresponds to the configuration of a turbofan engine. If the user desires to simulate a system other than a turbofan engine, he must redesignate the components to correspond to his system. For example, when simulating a turboshaft engine, the low pressure turbine component would contain a model for torque loading. With minimal software modifications to various menus in DEAN, the system configuration may be formally adapted. The user will also be prompted for the required model conditions such as inlet temperature, pressure, and fuel flow if the combustor and/or the afterburner are being simulated without a fuel control.

From the components chosen by the user, subroutine MAPSET decides which bivariate maps may be needed for the simulation and prompts the user to execute one of two tasks related to initialization of these maps. The rough map data, supplied by the user, resides in specified format in a dataset. When beginning a simulation MAPSET processes the rough data into a uniform grid for the bivariate interpolation routines and, upon request, rewrites the data over the original dataset. The processing of rough data needs to be done only once because MAPSET can determine when the data in the dataset is rough data or the pre-processed grid. DEAN is capable of processing eight bivariate maps which have been designated as shown in table 2. The user may wish to change the designated maps and, may do so as long as no more than eight bivariate maps are specified. Also, upon request, MAPSET will graphically display the processed map data as each map is processed. This allows the user to verify the maps specified by the data he has supplied.

TABLE 1. - LIST OF COMPONENTS
SIMULATED BY DEAN

Component number	Turbofan engine component
1	Inlet
2	Fan
3	Compressor
4	Combustor
5	High pressure turbine
6	Low pressure turbine
7	Afterburner
8	Nozzle
9	Bypass
10	Rotor dynamics
11	Main fuel control

TABLE 2. - AVAILABLE BIVARIATE MAPS
SUPPORTED BY DEAN

Map number	Description
1	Fan pressure ratio
2	Fan temperature ratio
3	Compressor pressure ratio
4	Compressor temperature ratio
5	High pressure turbine pressure ratio
6	High pressure turbine enthalpy
7	Low pressure turbine pressure ratio
8	Low pressure turbine enthalpy

Subroutine SSOPPT is responsible for providing the simulation with an initial starting point or steady-state point. There are two mechanisms the user may choose to determine such a point. Generally this point consists of a vector of state variables that represent the pressure, temperatures, flows, shaft speeds, etc. of the system. There is one state variable for each differential equation in the system model as well as any parameters calculated by empirical or theoretical equations such as choked flow for example. The first mechanism is to input the states directly from the keyboard. DEAN will prompt the user for the states, component by component as specified by the system definition portion of DEAN. An option is provided to use this input as an initial guess in conjunction with the governing equations and iterate to a steady-state point by a modified Newton-Raphson technique. An alternate method for determining a steady point is to use the initial guess as a starting point for a transient and run the transient until the dynamics settle out. Once a steady state point has been calculated, it may be stored with the system configuration for future reference. This allows the user to select a saved steady-state point rather than entering the states each time a simulation is run.

The subroutine TRNSNT is responsible for running the transient simulation. TRNSNT prompts the user for one of four integration schemes and, after an acceptable input, requests more specific information such as initial time step, length of transient time, frequency at which states should be displayed at the terminal, and also whether the user wishes to interact directly with the simulation. Data generated by the transient simulation is stored at specified intervals for graphical analysis at a later time.

The subroutine PLOT is responsible for generating graphics output from DEAN. This subroutine supports various options highlighted by table 3. These options are discussed briefly later in the paper.

The last group of important subroutines are listed in table 4. These subroutines contain all of the modeling equations required by the simulation. Included in these subroutines are data for the component geometry, data for single variable functions, code to store data for graphical analysis, bivariate map calls, evaluation of specific thermodynamic properties such as C_p = f(temperature), etc. These subroutines can be written by the user who wishes to develop an analytical model, as opposed to using those provided. Currently DEAN is set up to integrate up to one hundred state variables.

TABLE 3. - GRAPHICS PACKAGE OPTIONS
FROM SUBROUTINE PLOT

Option	Description
1	Read plot labels from labels dataset
2	Read plot data from dataset
3	Save plot data on dataset
4	Define or modify plot parameters
5	Display a menu of available plot parameters
6	Change the time interval of the data plotted
7	Automatically make a hardcopy of each plot
8	Generate a standard cartesian plot
9	Superimpose a map on a standard cartesian plot
10	Put data classification labels on all plots

TABLE 4. - LIST OF DEAN'S
COMPONENT SUBROUTINES

Subroutine	Description
EQUNS	Subroutine for "global" equations
INLET	Inlet duct model
FAN	Fan model
COMP	Compressor model
BURNER	Combustor model
HPTURB	High pressure turbine model
LPTURB	Low pressure turbine model
AFTBRN	Afterburner model
NOZZLE	Nozzle model
BYPASS	Bypass duct model
MFC	Engine control model
ROTDYN	Dynamic rotor balance equations

It may be noted, that while the component subroutine names correspond to those of a turbofan engine, they may also be used to model components of virtually any dynamic system.

INTERACTIVE ASPECTS OF DEAN

The simulation program, DEAN, has five major areas of user interaction. The first area involves defining the system components (listed in table 1). DEAN will prompt the user for the types of components required for the simulation, and thus subsystems of the turbofan engine may be modeled, i.e., compressor and nozzle. Based on the component models used, DEAN will prompt the user for specific information such as inlet pressure and temperature of the model or fuel flow for a combustor.

The second area of interaction is map grid definition for bivariate maps ($z = f[x,y]$). DEAN will prompt the user for information concerning the definition of the grid. The program can presently handle as many as eight bivariate

maps as listed in table 2. Such maps are usually required for fan, compressor, high pressure turbine, and/or low pressure turbine performance data.

The next area of interaction concerns the initial values of the states for the simulation. The user must initially enter values for the states which give the integration schemes a starting point. These values, which usually correspond to a steady point are entered from the keyboard. The user may also choose to select a previously entered steady point.

The most useful area of interaction is during the actual system simulation. As the transient proceeds, the program will save data on specified parameters (up to 100) with a maximum of 1000 points per parameter. The method of saving the data is an integral part of the component subroutines. During the transient the user may change various parameters based on the simulation's output.

The final area of interaction is in data analysis. DEAN supports a menu driven graphics package (table 4) with capabilities for basic mathematical manipulation of the transient data.

MODULAR ENGINE COMPONENTS

A turbofan engine is generally comprised of those components listed in table 1. In the Dynamic Engine ANalysis program there is a subroutine allocated for each of these components. The subroutines consist of FORTRAN code for evaluating derivatives and storing data for analysis. Also, there is code to generate parameters, perform map lookups for single and bivariate maps, and evaluate thermodynamic properties for specific components. These component subroutines are always called by DEAN in the order shown in table 1.

This modular simulation structure allows the user to develop turbofan engine systems and subsystems, component by component. This method is useful because it allows easier debugging of both coding and theoretical problems. Currently, the only restriction is that any subsystem must have its components ordered as per table 1. For example, a turbofan core consisting of a compressor, a combustor, and a turbine in line may be simulated. However, a change in component arrangement or system topology would require minor code changes and possible renaming or addition of component subroutines.

INTEGRATION SCHEMES

The mathematical model of the system is represented generally by ordinary first order nonlinear differential equations. There are several methods which can be used to integrate these equations. DEAN supports the following integration schemes: (1) second order Runge-Kutta, (2) fourth order Runge-Kutta, (3) Adams predictor-corrector, and (4) Gear's method for stiff systems. Each of these integration schemes has its own appropriate applications.

The second and fourth order Runge-Kutta schemes (ref. 5) execute similarly. The second order scheme requires fewer function evaluations, and thus, less CPU time. However, the fourth order scheme is more accurate. In general, these explicit methods should be used when model frequencies and accuracy requirements are high.

The Adams predictor-corrector integration (ref. 6) has several appealing attributes. It is a variable order (up to 12), variable time step method. The algorithm is a multistep process in which there is a predictor for extrapolating to the solution and a corrector for interpolating a solution based on the extrapolation. The interpolation procedure involves iteration and, therefore, requires more computation time than the one-step Runge-Kutta (fourth order method) if there are more than two iterations in the corrector process.

The Gear's algorithm for integrating differential equations (ref. 7) was developed for systems that have a wide range of frequencies (stiff systems). A turbofan engine can be considered as such a system. The Gear's method is a backward difference scheme capable of variable order (up to five) and variable time step. It is useful for simulations that require a substantial amount of computer time for function evaluations.

GRAPHICS CAPABILITIES

The last of DEAN's main menu options is the Data Analysis package. This set of subroutines allows the user to manipulate, as well as plot, data generated during a transient. The main subroutine which controls execution of this set of code is subroutine PLOT. PLOT contains a menu of ten options for the user to choose from.

The ten PLOT options allow the user to: (1) read user specified plot labels corresponding to the parameters saved during a transient, (2) read plot data stored in a dataset from a previous transient, (3) write transient data into a dataset for future reference, (4) perform basic mathematical manipulation of the transient data and/or modify plot title and labels, (5) display a list of the plot labels, (6) change time interval over which the data is plotted, (7) automatically hardcopy plots (assuming hardcopy capability exists), (8) produce cartesian plots of specified parameters, (9) plot specified parameters with a bivariate map (corresponding to table 2) super-imposed on the data, (10) write security classification labels on plots of the transient data.

CONCLUDING REMARKS

DEAN is a versatile simulation program written in FORTRAN IV. Designed for interactive dynamic simulation of turbofan engine systems and subsystems, DEAN has successfully simulated a number of component models, such as the compressor rig example in the Appendix and a full turbofan engine. Further, the interactive graphics capabilities have proven to be useful for simulation development and analysis.

DEAN is supported by interactive debugging, the IMSL library (9th edition), and a FORTRAN accessible graphics package.

APPENDIX A

COMPRESSOR RIG MODEL EXAMPLE

The Dynamic Engine ANalysis program can best be understood by actually developing a simulation of a system. Here, a simple simulation of a compressor rig is used to demonstrate the capabilities of DEAN. Also, the model development will not overshadow the use of DEAN as a simulation program.

First the model is defined. Let us assume that the compressor rig can be modeled as a constant speed, one-dimensional, multi-lumped volume, incompressible system with three major components: (1) inlet volume, (2) compressor, and (3) turbine nozzle. Note that these three elements correspond to those in table 1. The inlet will be modeled as a single lumped volume and the compressor will also be represented as a single volume employing the actuator disk concept suggested by Greitzer (ref. 8) for compressor stall. The turbine nozzle will be modeled by using a choked flow relation between temperature and pressure.

Generally three parameters provide communication between components: here (1)pressure (P), (2) mass flow ($\dot{W}$), and (3) temperature (T) are used. These parameters are calculated by integrating the non-linear first order differential equations that result from the conservation of mass, momentum, and energy and the equation of state in differential form. Figure 2 shows a typical lumped volume with the various parameter designations. The general form of these equations for such a lumped volume analysis is as follows, where ρ is density, t is time, V_{ol} is volume, R is gas constant of air, L is the volume length, A is cross sectional area, F_{ext} denotes sum of frictional and pumping forces, Q denotes the net heat transfer to the volume, W denotes net work done by the volume, C_v is the constant volume specific heat of the fluid, and γ the ratio of specific heats.

conservation of mass

$$\frac{\partial \rho}{\partial t} = \frac{\Sigma \dot{W}_{in} - \Sigma \dot{W}_{out}}{V_{ol}} \tag{1}$$

equation of state

$$\frac{dP}{dt} = RT \frac{\partial \rho}{\partial t} + \frac{P}{T} \frac{dT}{dt} \tag{2}$$

conservation of momentum

$$\frac{d\dot{W}}{dt} = \frac{g_c}{L} \left\{ P_{in} A_{in} - P_{out} A_{out} + F_{ext} \right\} \tag{3}$$

conservation of energy

$$\frac{dT}{dt} = \frac{RT}{PV_{ol}} \left\{ \Sigma \dot{W}_{in} (\gamma T_{in} - T) - \Sigma \dot{W}_{out} (\gamma T_{out} - T) + \frac{Q - W}{C_v} \right\} \tag{4}$$

Substituting equation (1) into equation (2) yields:

modified equation of state

$$\frac{dP}{dt} = \frac{RT}{V_{ol}} (\Sigma \dot{W}_{in} - \Sigma \dot{W}_{out}) + \frac{P}{T} \frac{dT}{dt} \tag{5}$$

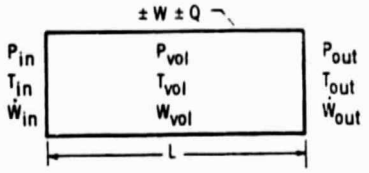

Figure 2. - Typical lumped volume diagram.

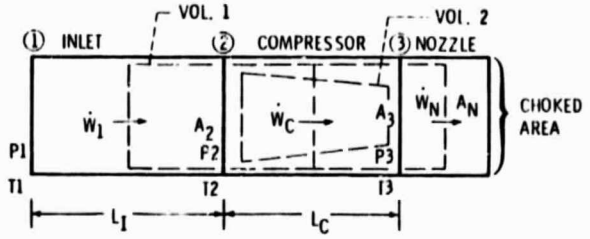

Figure 3. - System block diagram.

TABLE 5. - STEPS REQUIRED TO DEVELOP A SIMULATION

Step number	Instructions
1	Define the system to be modeled
2	Divide the system into components
3	Write dynamic equations for each component
4	Write FORTRAN component subroutines
5	Create datasets to be used by simulation
6	Access and execute DEAN

From these equations, models of an inlet and compressor can be developed. Table 5 shows the general steps involved in developing the simulation. A block diagram of the system is shown in figure 3. Using the block diagram and the general conservation equations the inlet equations may be written as follows:

pressure (state)
$$\frac{dP_2}{dt} = \frac{P_2 T_2}{V_{ol,1}} (\dot{W}_1 - \dot{W}_c) + \frac{P_2}{T_2} \frac{dT_2}{dt}$$

mass flow (momentum)
$$\frac{d\dot{W}}{dt} = \frac{g_c}{L_1} \left\{ P_1 A_1 - P_2 A_2 - R \dot{W}_2^2 \right\}$$

temperature (energy)
$$\frac{dT_2}{dt} = \frac{RT_2}{P_2 V_{ol,2}} \left\{ \dot{W}_1 (\gamma T_1 - T_2) - \dot{W}_c (\gamma T_2 - T_2) \right\}$$

where F_{ext} is modeled as $F_{ext} = -R\dot{W}^2$ (R = constant to give appropriate pressure drop). Also, for the inlet, P and T are constant inlet conditions. The compressor is modeled in much the same way. The equations for the compressor are as follows:

pressure (state)
$$\frac{dP_3}{dt} = \frac{RT_3}{V_{ol,2}} (\dot{W}_c - \dot{W}_n) + \frac{P_3}{T_3} \frac{dT_3}{dt}$$

mass flow (momentum)
$$\frac{d\dot{W}_c}{dt} = \frac{g_c}{L_c} \left\{ P_2 A_2 - P_3 A_3 + F_{ext} \right\}$$

$$F_{ext} = P_{3,map} A_3 - P_{2,map} A_2$$

temperature (energy)
$$\frac{dT_3}{dt} = \frac{RT_3}{P_3 V_{ol,2}} \left\{ \dot{W}_c (\gamma T_2 - T_3) - \dot{W}_n (\gamma T_3 - T_3) - \frac{\dot{W}_c}{\gamma} (T_{3,map} - T_{2,map}) \right\}$$

where the values of $P_{2,map}$ and $P_{3,map}$ are the pressures corresponding to a steady state compressor performance map in terms of pressure ratio (pressure ratio is a function of corrected flow and corrected speed) and $T_{2,map}$ and $T_{3,map}$ are temperatures corresponding to another steady-state performance map in terms of temperature ratio (temperature ratio is a function of corrected flow and corrected speed.) The turbine nozzle will be modeled by using the following choked flow nozzle equation:

mass flow $$\dot{W}_n = A_n P_3 \left\{\frac{2\gamma}{\gamma + 1}\right\}^k \left\{\frac{RT_3}{g_c}\right\}^{1/2}, \quad k = \left\{\frac{\gamma + 1}{\gamma + 1}\right\}$$

The governing equations are now coded into separate subroutines for each component. Once these subroutines are written for all of the component models the user must collect all the data required for the model. For the compressor example there are two data requirements; (1) compressor map data and (2) geometry data.

Once all of the preliminary work is done for the simulation DEAN may be run. A typical run is similar to that discussed in the INTERACTIVE ASPECTS OF DEAN section.

For this particular example, the nozzle exit area is ramped closed during the transient while compressor shaft speed is held constant as shown in figure 4. The Adam's method was chosen as the integration scheme with an initial time step of 0.0001. The length of the transient was 0.50 sec which required 6.63 sec of CPU to integrate six states. This disturbance causes a compressor instability called surge. Figure 5 shows graphically the instability superimposed on the appropriate performance map. Figures 6 through 8 show time traces of inlet and compressor exit flow, temperature, and pressure, respectively.

REFERENCES

1. Fishbach, L.H., and Caddy, M.J., "NNEP - The Navy NASA Engine Program," NASA TM-X-71857, 1975.

2. Sellers, J.F., and Daniele, C.J., "DYNGEN - A Program for Calculating Steady-State and Transient Performance of Turbojet and Turbofan Engines," NASA TN-D-7901, 1975.

3. Szuch, J.R., "HYDES - A Generalized Hybrid Computer Program for Studying Turbojet or Turbofan Engine Dynamics," NASA TM-X-3014, 1974.

4. Daniele, C.J., Krosel, S.M., Szuch, J.R., and Westercamp, E.J., "Digital Computer Program for Generating Dynamic Turbofan Engine Models (DIGTEM)," NASA TM-83446.

5. Carnahan, B., Luther, H.A., and Wilkes, J.O., Applied Numerical Methods, Wiley, New York, 1969, pp. 361-380.

6. Hornbeck, R.W., Numerical Methods, Quantum, New York, 1975, pp. 196-202.

7. Gear, C.W., Numerical Initial Value Problems In Ordinary Differential Equations, Prentice-Hall, Englewood Cliffs, NJ, 1971.

8. Greitzer, E.M., "Surge and Rotating Stall in Axial Flow Compressors. Part I: Theoretical Compression System Model," Journal of Engineering for Power, Vol. 98, No. 2, April 1976, pp. 190-198.

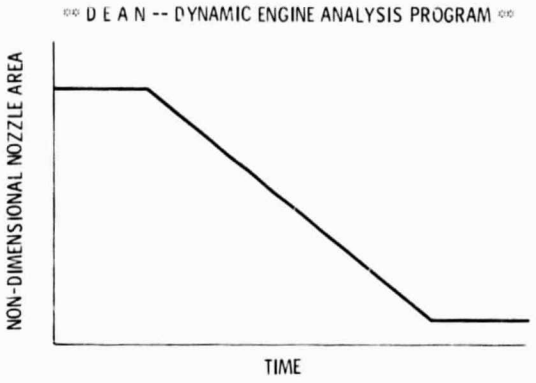

Figure 4. - Plot of exit area vs. time.

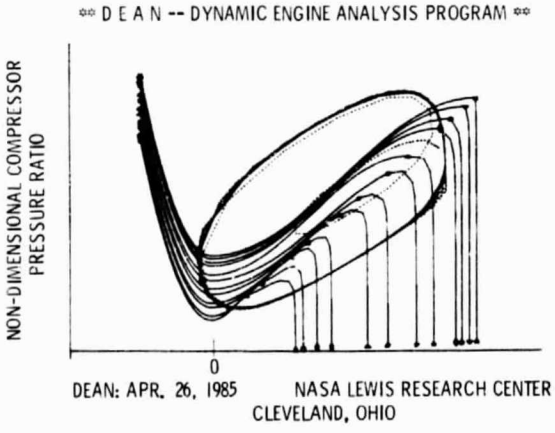

Figure 5. - Plot of compressor instability.

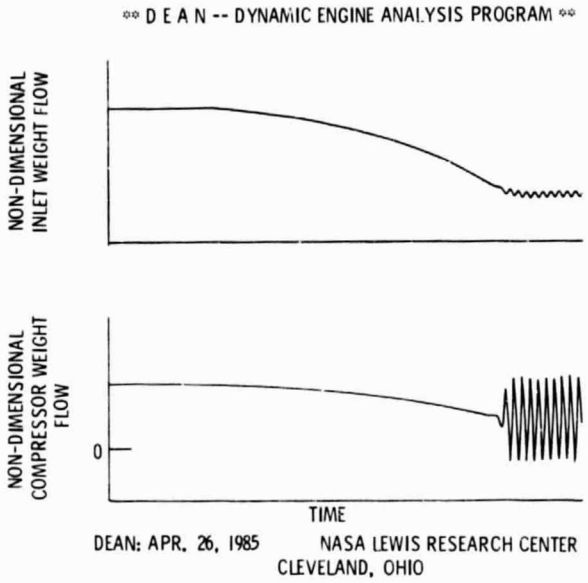

Figure 6. - Time traces of weight flow.

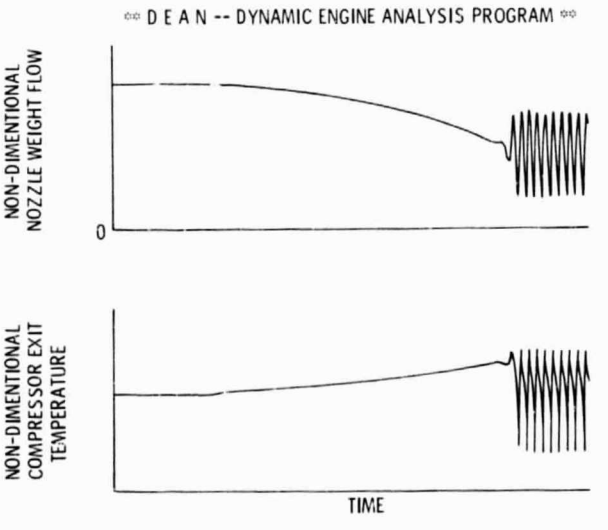

Figure 7. - Time traces of weight flow and temperature.

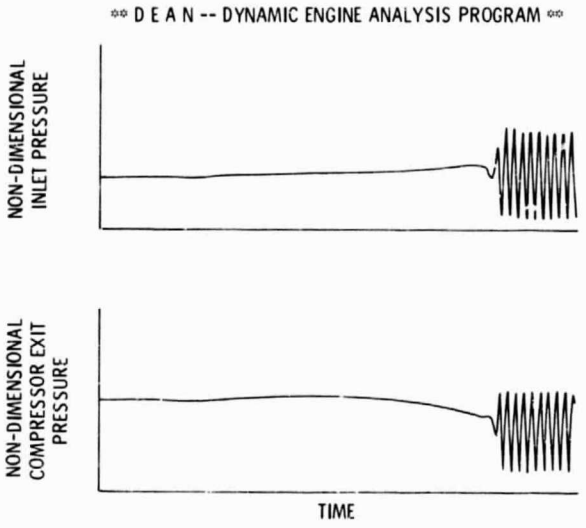

Figure 8. - Time traces of pressure.

1. Report No. NASA TM-87033 USAAVSCOM TR-85-C-10	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle	5. Report Date
DEAN: A Program for Dynamic Engine ANalysis	
	6. Performing Organization Code 505-40-14
7. Author(s) Gerald G. Sadler and Kevin J. Melcher	8. Performing Organization Report No. E-2588
	10. Work Unit No.
9. Performing Organization Name and Address NASA Lewis Research Center and Propulsion Laboratory U.S. Army Research and Technology Laboratories (AVSCOM) Cleveland, Ohio 44135	11. Contract or Grant No.
	13. Type of Report and Period Covered Technical Memorandum
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546 and U.S. Army Aviation Systems Command, St. Louis, Mo. 63120	14. Sponsoring Agency Code

15. Supplementary Notes

Gerald G. Sadler, Propulsion Laboratory, AVSCOM Research and Technology Laboratories, Lewis Research Center, Cleveland, Ohio; presently with NASA Lewis Research Center. Kevin J. Melcher, NASA Lewis Research Center. Report prepared for the Twenty-first Joint Propulsion Conference, cosponsored by the AIAA, SAE, and ASME, Monterey, California, July 8-10, 1985.

16. Abstract

The Dynamic Engine ANalysis program, DEAN, is a FORTRAN code implemented on the IBM/370 mainframe at NASA Lewis Research Center for digital simulation of turbofan engine dynamics. DEAN is an interactive program which allows the user to simulate engine subsystems as well as a full engine systems with relative ease. The non-linear first order ordinary differential equations which define the engine model may be solved by one of four integration schemes, a second order Runge-Kutta, a fourth order Runge-Kutta, an Adams Predictor-Corrector, or Gear's method for stiff systems. The numerical data generated by the model equations are displayed at specified intervals between which the user may choose to modify various parameters affecting the model equations and transient execution. Following the transient run, versatile graphics capabilities allow close examination of the data. DEAN's modeling procedure and capabilities are demonstrated in this paper by generating a model of simple compressor rig.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Simulation Gas turbine engines Dynamic analysis Controls	Unclassified - unlimited STAR Category 07

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price*
Unclassified	Unclassified		

*For sale by the National Technical Information Service, Springfield, Virginia 22161