HEWLETT-PACKARD

KEYBOARD

VOL. 8 NO. 2

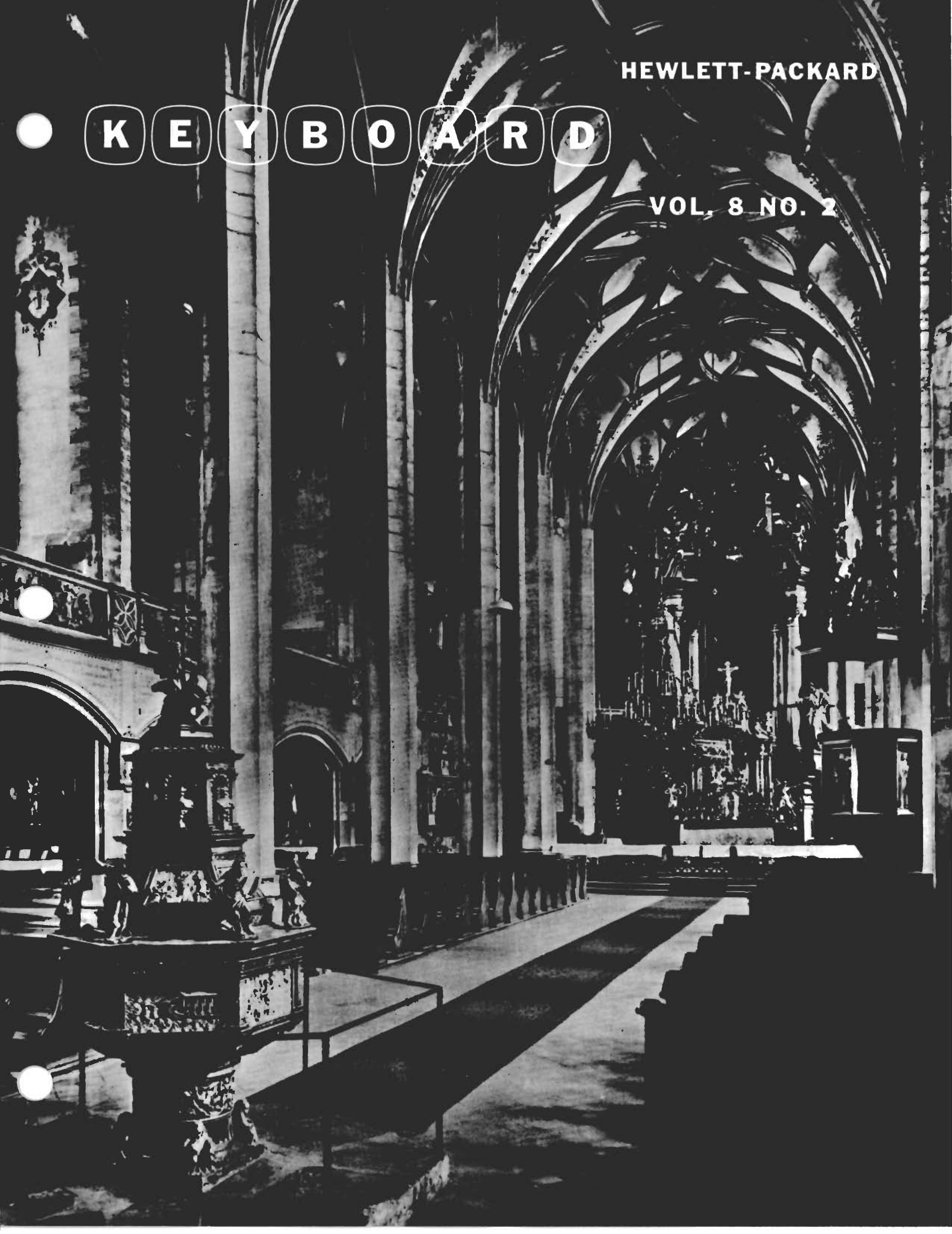


TABLE OF CONTENTS

Page

FOR HEWLETT-PACKARD
9800 AND 9100 SERIES CALCULATOR USERS
PUBLISHED AT P.O. BOX 301,
LOVELAND, COLORADO 80537

Editor: Nancy Sorensen   Artist/Illustrator: H.V. Andersen

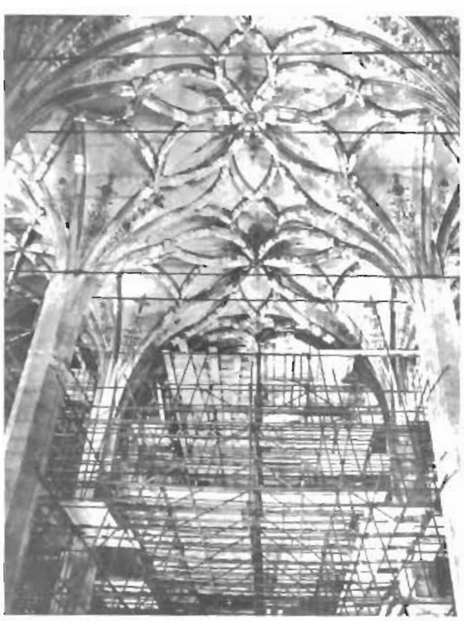
Historic Landmark Saved By Modern Technology

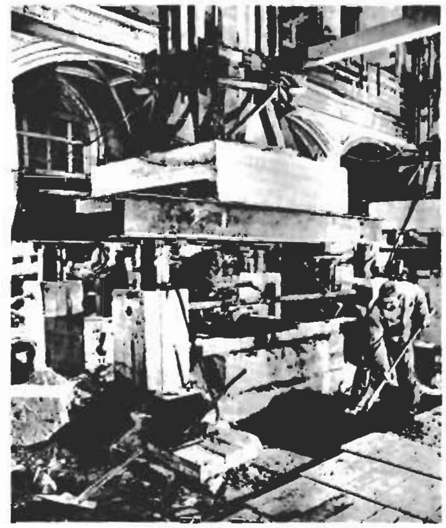
by Dipl. Ing. Karel Vrána, Dipl. Ing. Tomáš Cáp, and Dipl. Ing. Jiří Souček

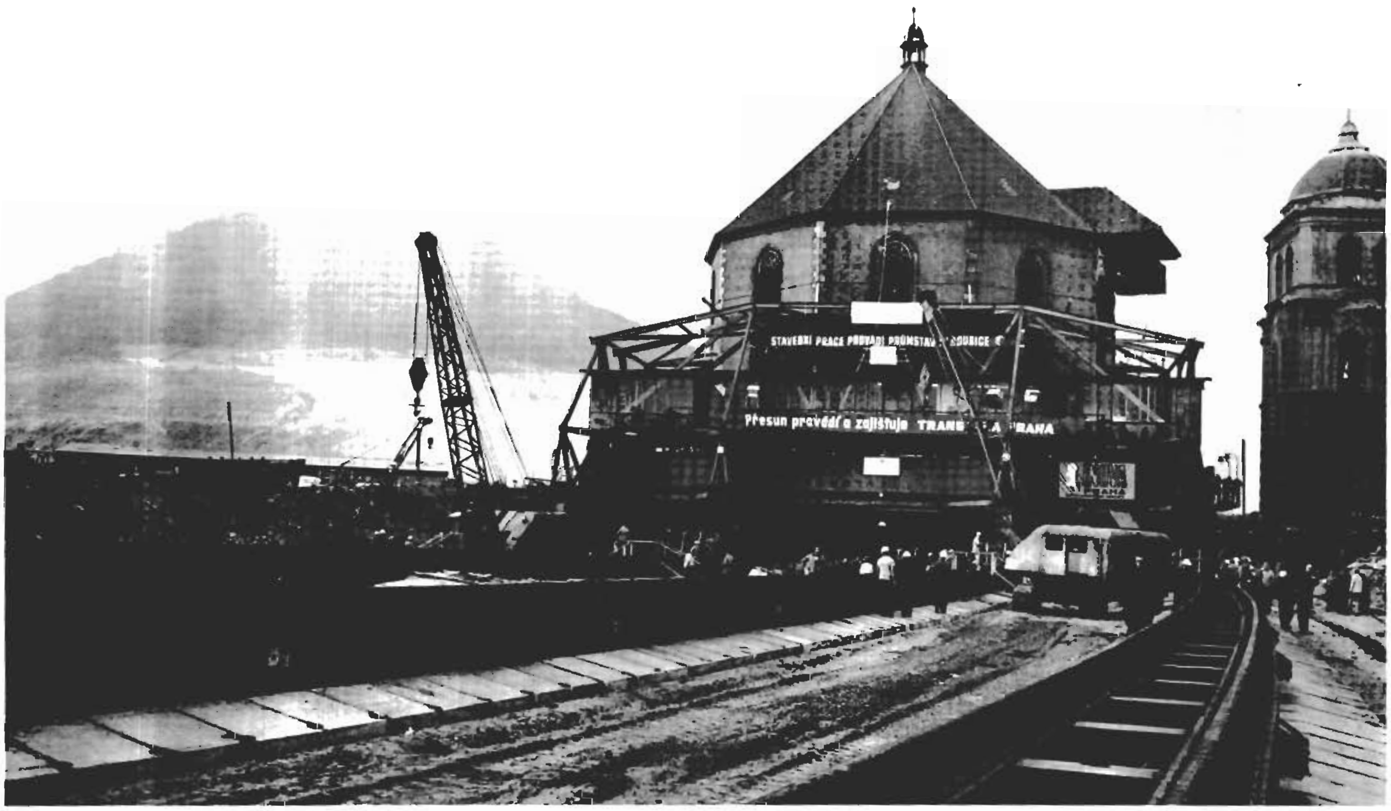
The town of Most was established in the thirteenth century near a major trade route in what is now the Bohemian area of Czechoslovakia. In 1517 the foundation was laid for the town's most important landmark — the Church of the Virgin Mary.

The designer of the three-aisled, lierne-vaulted edifice was Saxonian builder Jacob Heilmann, who supervised the construction until his death. Work continued under the supervision of Georg of Maulbroun, Master Peter and Master Yorke, and the church was completed in 1549.

Some portals, the ogival tower roof and sculptural decorations of Saxon Renaissance style were added in the mid-sixteenth century during a second phase of construction. Baroque influence is evident in the elaborate main altar and the ornamental sculptures fitted to the buttresses at a later date. The church was painted and provided new furnishings between 1880 and 1883. Preserved in that form, it links the various Saxonian forms of art and is the most elegant representation of Late Gothic architecture remaining in North Bohemia.






STAVEBNÍ PRÁCE PROVÁDÍ PRŮMSTAV
Přesun provádí a zajišťuje

The area surrounding Most is rich in coal deposits, and Most itself is now a mining town. An estimated 87 million tons of brown coal is deposited under the valley in which Most is situated. To facilitate extraction, the CSSR Government decided in 1964 the town should be torn down and rebuilt on a new site. Officials simultaneously stipulated preservation of the Church of the Virgin Mary, and debate began on whether to transfer the church as a whole, leave it on its site and excavate around it, dismantle and reconstruct the building on a new site, or preserve only the important cultural architectural details.

A committee of the Ministry of Culture of the CSSR ultimately recommended transferring the church and delegated responsibility for the plans and implementation to Transfera, a corporation founded for studying and organizing jobs of this kind. During the three-year study and preparation period which followed, Transfera engaged the services of a number of institutions and works, including the INOVA Research and Development Works, which was responsible for the design, development, and implementation of the automatic leveling system controlled by a Hewlett-Packard 9821A Calculator.

The VZKG Institute of Applied Mechanics, Brno, developed a system of 525 sensors for continuous measurements of the movement of church brickwork and arches, as well as checking the steel structure strains and loads of transport trucks, the forces in pushing and braking cylinders, and for measurements of the transfer dynamics.

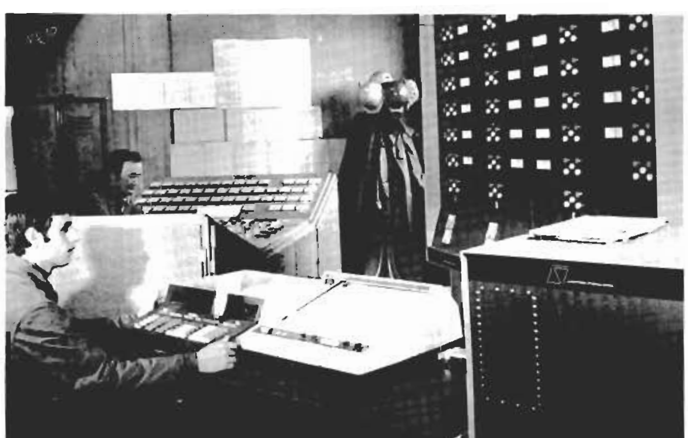
Additional equipment which had to be designed and manufactured for the transfer included:

- An extensive hydrostatic system, including the tanks, a method for liquid filling, remote control of the liquid level, and provisions for maintaining equal barometric pressure throughout the system.
- Hydrostatic apparatuses with high-resolution electric outputs for measuring liquid levels.
- Input circuits using galvanic separation by optoelectronic elements, allowing the mixing of measured signals with alarm signals.
- Scanner of measuring points for 64 channels.
- Analog-to-digital converter.
- Interfacing circuits for data input to the calculator, including a special circuit that accelerated the entry capability by generating control signals to automatically shift the scanner to a further measuring point.
- Interfacing circuits for calculator output, including a circuit to generate control signals causing the channel selector to shift a step after each data output.
- Interfacing circuits to generate control signals, reset the scanner of measuring points, reset the channel selector, generate time signals, etc.
- Special memory circuits with preset counters.
- Signaling panels to indicate the function of the scanner, output signals transmitted to servos, alarm situations, etc.
- Power amplifiers of output channels to control the servos.
- Control room with various power supplies, distribution networks, protection devices, cabinets, and miscellaneous equipment.

Preparation of the church itself included protecting the vaults by a supporting reinforced concrete structure. A reinforced concrete ring beam was constructed under the outside walls to link the historical masonry and the bracing steel structure. Reinforcement of the building as a whole was effected by an orthogonal lattice structure inside and outside the church. A tower was dismantled and taken to the new site to be rebuilt after the transfer.

Study and preparations completed, the church, 60 m (196.85 ft) long, 29,7 m (97.44 ft) wide, 31,5 m (103.35 ft) high and weighing 9.600 metric tons (10,560 tons) with its supporting structure, was ready to be shifted on 424 wheels along a four-pair railway track to its new site.

The steel carrying structure, despite its impressive size, was not able to maintain the brickwork in the desired position, so it was supported at 53 points, where longitudinal and transverse girders intersected, by vertical pistons of hydraulic servos situated on 53 trucks. The servos compensated for track irregularities and subsoil movements or deflections so that the supporting points could be maintained in the required vertical positions within an accuracy of ± 1 mm. This was the purpose of INOVA's unique automatic leveling system controlled by the HP 9821A Calculator.



This was the control room for the automatic leveling system. Opposite the 9821 operator is the bank of red and green lights used to signal the frequency, duration, and position of servo activity.

Mr Frantisek Sramek (front right), Minister of the Building Industry of the Czech Socialist Republic, pressed the RUN PROGRAM key and sets the automatic leveling control system into operation.

Hydrostatic vertical position-measuring devices were installed at each of the 53 support points and interconnected by hoses to the equalizing tank, providing for equal barometric pressure throughout the system. The liquid level in the system constituted a measure for determining the respective vertical deviations. The electric output of the measuring apparatuses was periodically sampled by a scanner controlled by the calculator through a suitable interface and applied to an analog-to-digital converter. The digital signals were then transferred to the calculator memory. Also stored in the memory were parameters of the transient response of the individual measuring devices, which differed considerably because of various lengths of the connecting hoses.

To allow for the influence of inertia on the hydrostatic part of the system, the scanning of the vertical position-measuring devices was carried out in three sequences. Thus the calculator, in each cycle, obtained 3 x 53 data, from which it calculated the first and second derivative of signals from the measuring apparatuses. This made it possible for the calculator, despite considerable inertia of the measuring devices and differences in their transient response parameters, to calculate a correct value of disturbances, i.e., the deviations of individual supporting points from required vertical positions. The deviations were related to the leveling plane, the position of which was optimized by calculation during each cycle, so that the number and magnitude of corrections would be minimal.

The servos controlling the positions of the supporting points only allowed movement at a constant speed. Correction commands could be made for direction (up or down), and the length of servo operation time determined the magnitude of movement. The data determining the direction and time of movement of each servo were transmitted by the calculator through the interfacing circuit and channel selector to 53 external semiconductor memories. Each memory was equipped with two flip-flops ("up" and "down") and a counter preset to the number of times steps corresponding to the required servo operation time.

Counters and flip-flops set, the flip-flop outputs incorporating an enabling pulse were applied through transistorized amplifiers and power relays to the electrohydraulic pistons of the servos, producing a force of up to 500 tons each. The enabling pulse was initiated by the calculator after all memories were filled. Thus the respective servos were set into motion in the required direction for a length of time determined by the number of time steps and the presetting of the counters.

Servo operation was indicated by 53 pairs of red and green bulbs located on a display to correspond with the location of the church support points. The red bulbs signalled upward movement, and the green bulbs signalled downward movement. The operator could "see" all the automatic leveling operations, their frequency and duration, and positions of the corrections.

The software for the 9821 contained programs for regulation (automatic leveling), system adjustment, church lift-up and set-down, as well as data saving and diagnostic programs. The most extensive was the regulation program. It occupied most of the 1447 registers of the extended memory. In addition to automatic leveling, this program allowed a change of its basic parameters, e.g., the structure and time of the regulation cycle, adjustment of the dead zone range, and setting the shape of the transport "plane" during operation.

The 9821's alphanumeric display provided easy dialogue between the operator and calculator, thereby simplifying operation of the program. By depressing the "set flag" key, the operator informed the calculator he required a function. The calculator finished the regulation cycle in progress and sequentially offered the possible functions, to which the operator responded "yes" or "no " If it received a yes, the 9821 either carried out the required functions and immediately proceeded with automatic regulation, or it asked for a more detailed specification and then performed the function.

Strip mining continued as the Church of the Virgin Mary (at the right of photo) was prepared for the move. The tower was dismantled before the church was moved and will be rebuilt at the new site.

This procedure occurred repeatedly throughout the duration of the transfer (645 hours, 6 minutes) as the church was moved at the rate of 2,173 cm (0.855 in.) per minute along the tracks, which had an arc with a radius of 548,5 m and a downgrade of 1.23%.

The effectiveness of the automatic leveling system is best exemplified by the maximal change of the brickwork occurring in a preexisting crack, which amounted to ±0,4 mm (0.016 in.), with the temperature change alone affecting the crack by ±0.005 mm (0.0002 in.).

On October 27, 1975, the Church of the Virgin Mary was placed on its new foundation 841,1 m (just over ½ mile) from its original site. And after performing 8,791,790 measurements, approximately 55,000 regulation cycles and 586,000 regulation corrections, the HP 9821 Calculator was turned off after 1,041 hours of continuous performance.

CURRICULUM VITAE

Dipl. Ing. Karel Vrána was born in 1926. He received his degree in Electrical Engineering at the Technical University in 1958. At the Communications Research Institute and later at the Communications Computing and Checking Centre, his work was related to optical character recognition. In 1966, he joined the Development Workshop of the Czechoslovak Academy of Sciences, working on interfacing HP calculators with tape readers, tape punches, teleprinters, typewriters, measuring instruments, tape recorders, apparatuses for the blind, etc.

During the three-year study and preparation for the transfer of the Church of the Virgin Mary, Ing. Vrána served as consultant and designer of interfacings and measuring equipment for the automatic leveling system.

Dipl. Ing. Tomás Cáp, CSc. was born in Hradec Králové in 1935. Following studies at the Academic Grammar School in Prague, he attended and received his Electrical Engineering degree at the Technical University. At the Research Institute for Telecommunications, Ing. Cáp was involved with pulse code modulation, and in 1969 he was awarded the Degree of Candidate of Sciences for his efforts in that field.

He is presently employed by INOVA Research and Development Works where, for the past several years, he has worked on hardware and software to be used in the Most church relocation.

Dipl. Ing. Jiří Souček was born in 1932 in Benešov and attended grammar school in that community. From 1951 to 1955 he studied at the Mechanical Engineering Faculty of the Technical University in Prague. After graduation he worked as a lecturer of mechanics and elasticity at the Agricultural Mechanisation Faculty. He joined the CKD National Corporation in 1960, where he was engaged in work with power semiconductors. He continued his studies at the Electrical Engineering Faculty at the Technical University and attended Charles University to study solid-phase physics. He was awarded the State Prize in 1968 for the development and introduction of power thyristors.

Employed by INOVA Research and Development Works since 1971, Ing. Souček headed the group that developed and constructed the automatic leveling equipment for the church transfer in Most.

Photo credits: Kilian and V. Zitny

Simplifying Optical Design

by Douglas C. Sinclair

Special programming turns the 9815A into an important tool for optical system layout.

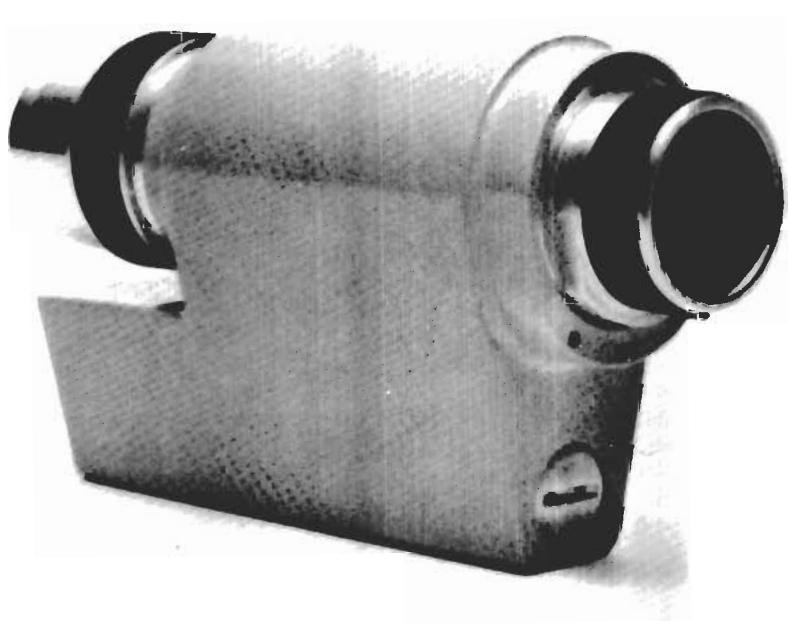
Optics, one of the oldest branches of physics, has been enjoying a renaissance and growth for the past several years that makes it one of today's most active areas for the development of new technology. The development, during the late 1950's and early 1960's of digital computers, solid-state photodetectors, and lasers created so many new areas for optical applications that there are still several that have not been explored.

An optical system may be virtually a complete instrument, as in a pair of binoculars, or it may be a subsystem within an instrument requiring the interfacing of a number of systems, as in distance-measuring equipment.

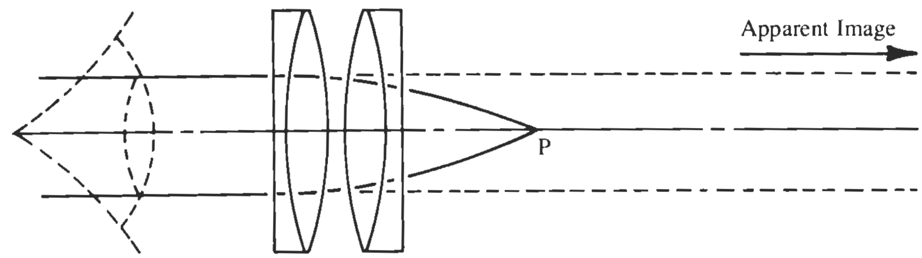
In designing an optical system, designers choose values for the system variables, such as the lens curvatures, to meet performance requirements. Performance is evaluated by computing the trajectories of rays that pass through the system. Typically, optical systems utilize lenses that form images of prescribed objects. If the rays from a point on the object miss the desired image point, the lens is said to have aberration, and the designer attempts to improve the image by choosing different curvatures. The ray trajectories depend on the lens curvatures in a highly non linear fashion, so the process involves a great deal of numerical computation. In view of this, it is not surprising that optical designers have watched developments in the computer industry with a keen interest.

Such familiar products as today's high-quality xerographic copiers employ computer-designed lenses and solid-state photodetectors, and the new point-of-sale check-out systems now beginning to appear in supermarkets around the country are the result of the marriage of laser, optical, and computer technology to create an overall system. Optical techniques are finding increased use in surveying, where optical distance-measuring equipment is revolutionizing field practice, and in construction, where laser alignment systems are rapidly replacing traditional mechanical equipment for tasks such as pipe laying. In medicine, several instruments are under development for use in clinical laboratories to perform analytical tasks automatically. Laser photocoagulators are now used routinely for the repair of detached retinas and computer-controlled automatic refractometers for testing vision are beginning to be used in optometry.

The number and diversity of today's optical applications have created an increased need for a variety of lens systems. Practically every new optical application requires a new lens; when the performance requirements for a lens are changed, the design must be changed to obtain an optimum configuration.



A Medical Pocketscope developed by ITT as a visual aid for people with night blindness. The instrument, developed under the direction of ITT engineer James Burbo, employs an electronic image intensifier tube that increases the apparent illumination of scenes at night. The low-cost eyepiece needed for the instrument was designed for ITT by the author and is a modification of a standard type.



Cross section of the eyepiece used in the instrument shown in Figure 1. According to usual custom, the eye is imagined to be looking through the eyepiece from the left. The eyepiece forms an image of the actual object (at P) at infinity, where the eye can view it comfortably.

An example of a typical requirement for a new lens system is the Medical Pocketscope recently developed by the Electro-Optical Products Division of ITT. Studies by Dr. Eliot Berson of the Department of Opthamology, Harvard Medical School, and Massachusetts Eye and Ear Infirmary, had shown that the Night Vision Pocketscope, a military optical instrument employing a small image intensifier, was a useful prosthetic aid for people with retinitis pigmentosa, a hereditary degenerative disease of the retina whose earliest symptom is night blindness. ITT wanted to modify the military instrument for this medical application, but had to substantially reduce the manufacturing cost. In reviewing the optical design, it was discovered that the eyepiece, while superlative in performance, was more complex than needed for this application. As a result, a design study was undertaken to develop a simple eyepiece for the final instrument.

In recent years the quality of optical systems has improved greatly. One indication of recent progress in the field is illustrated by the copying lenses used for integrated-circuit production. Today's lenses can resolve one hundred times as many image points as ones available ten years ago. In the future, we can expect equally impressive progress, particularly in the development of more and better zoom (variable-magnification) systems, very-high-resolution lenses, ultra-fast lenses, and wide-angle lenses. These advances are largely due to the increased use of computer-aided design.

Serious efforts to develop computer programs for optical design started in the 1950's, when machines such as the IBM card-programmed calculator and the Royal McBee LGP-30 became available. Since that time, it seems that optics programs have been written for just about every machine ever developed; but there has been a tendency to use larger and larger computers and, consequently, larger and more intricate programs.

Thus the designer has been faced with two major problems. He has had to know a great deal about the optical system he is designing, and he has had to understand the operation of the computer program. One of the country's leading optical designers recently commented that it takes a full year's training for a designer to learn how to make optimum use of the complex Grey Code, a program generally regarded as one of the most powerful available today. So although lens design has been greatly expanded in its capabilities by the use of computers, it has remained the province of those designers who work closely with large computers.

A further complication is the exclusive position of lens design within the larger field of optical engineering. The lens designer is responsible for the combination of lenses used by the optical instrument or system; the optical engineer is responsible for the overall instrument or system design. In a typical situation, the engineer is faced with the dilemma of establishing the relationships between the optics and the mechanical or electronic components of the system, with very little knowledge of what can be accomplished by different optical layouts. Lens designers cannot help much. They are so specialized in their own field they have little feel for what can be done with the electronic or mechanical components.

Now, with the advent of calculators such as the Hewlett-Packard 9815A and programs written especially for this purpose, optical engineers can carry out much of the preliminary analysis of an optical design before turning it over to the lens designer for detailed optimization. This approach solves both problems — learning to use complex hardware/software and bridging the gap between lens design and optical engineering.

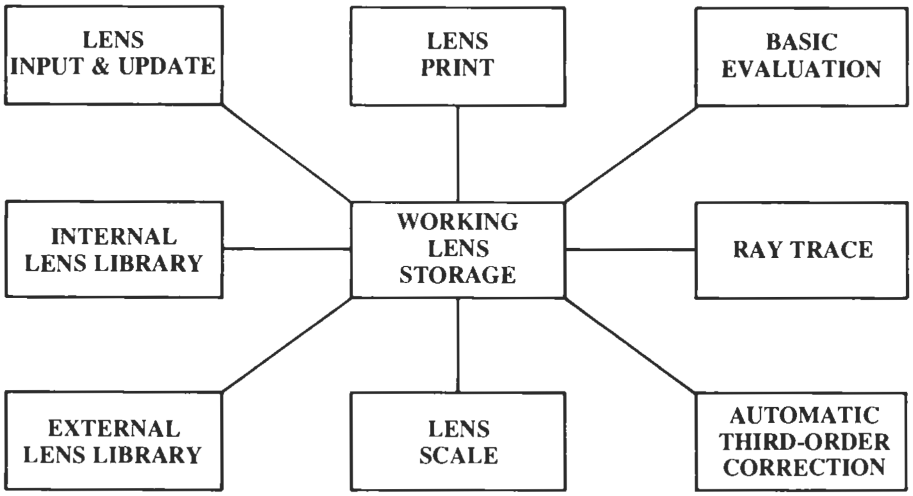
The design of a new optical system usually involves the following steps:

1. Thin-lens layout
2. Paraxial thick-lens layout
3. Third-order solution
4. Optimization
5. Evaluation

At any stage of the design it may be necessary to return to a previous stage, modify the system, and continue.

In the first stage, the basic layout of the system is set up. To simplify the analysis, the lenses are assumed to have negligible thickness and to be completely characterized by their focal lengths and diameters. Some of the considerations in this stage concern the image location magnification, light throughput, and mechanical constraints. It is important, in this stage, to make a number of sketches of the system to make sure that lenses are not required to perform impossible tasks.

Once the basic layout is established, real lenses with appropriate thicknesses are set up



The functional organization of this optical system layout program is typical of many optical design programs. It consists of a data file containing the constructional parameters of the system being considered, which is utilized by various subprograms. Starting from the upper-left corner, the Lens Input and Update program is used to put information into the working lens storage, and to edit and update erroneous entries. The Lens Print program prints the values of the constructional parameters on te line pritner. The Basic Evaluation program is used to calculate the gaussian constants, the paraxial-ray trajectories, the chromatic aberrations, and third-order aberrations. The Ray Trace program traces real Meridionial and Skew rays and ray fans through the system. The Automatic Third-Order Correction program is used to change the constructional parameters so that the system has desired residuals for the third-order aberrations. The Lens Scale program independently scales the lens focal length, aperture, and field angle by user-supplied factors. The External and Internal Library programs allow the user to store lenses on either a separate data cartridge (100 lenses/cartridge) or on the working data cartridge (20 lenses).

in the proper location. In this second stage, the basic lens types needed to satisfy aperture and field requirements are selected, and a paraxial solution for power, spacing, and chromatic correction is determined.

A so-called "third-order" solution is obtained in the next stage. Here, the shapes of the various lenses are chosen for the desired residual contributions to the third-order aberrations: spherical aberration, coma, astigmatism, and distortion. The third-order solution is usually the *sine qua non* of a good final design. The final design involves the detailed balancing of third-order aberrations against various higher-order aberrations. If a third-order solution cannot be obtained, it is unlikely that enough control over the system can be gained to get a good final balance.

Optimization, the fourth stage of optical design, involves the detailed balancing previously described. This is usually carried out by tracing rays through the system, and it is in this and the fifth stage, evaluation, that large computers have made an enormous impact on the quality of optical designs. The number of aberrations and system variables is so large that it is impossible for a human designer to do as good a job of balancing as a computer.

To carry out the balancing, the engineer sets up a group of target values for the system aberrations. A merit function consisting of the sums of the squares of the differences between the actual and target values of the aberrations is minimized by changing the system variables. Inasmuch as the relationship between the variables and the aberrations is highly nonlinear, the mathematical problem is very difficult to solve. The most common approach uses the "damped-least-squares" algorithm, which attempts to cope with system nonlinearity by limiting the size of steps taken to improve the merit function.

In the fifth stage, the perfomance of the optical system is evaluated in ways that relate to other parts of the system. A common method for evaluating the performance of an optical system involves the computation of the modulation transfer function of the system, which describes the relative capability of the system to form images of periodic objects (gratings, for example) having different frequencies.Other performance measures are the relative importance of diffraction and aberrations and the sensitivity of the system to manufacturing errors. If the system passes these tests, the design is done; if not, the designer must return to stage four, or perhaps even an earlier stage, and try again.

In the five stages of optical design, large computers are needed in the final two only. The first two stages require a great deal of human input, and the third is well within the capabilities of the 9815. As a benchmark of what can be done on a calculator, we note that optical systems containing up to sixteen surfaces can be stored, and rays can be traced with a speed of about 0.5 second per ray surface.

An engineer can experiment with several possible optical layouts and can carry the analysis through to the point where he or she has a good idea of whether the lens designer can execute the final design. In addition, by becoming more involved with the details of the optical system, optical engineers will be able to communicate better with lens designers. This should lead to less wasted effort and, ultimately, to better and cheaper optical systems.

CURRICULUM VITAE

Douglas C. Sinclair received his PhD in Optics from the University of Rochester in 1963. He has been professionally active in the fields of lasers and optical design. He was Technical Director of Spectra-Physics and has served on the faculty of the University of Rochester's Institute of Optics since 1969. In addition to his duties as Professor of Optics, Dr. Sinclair is Editor of the Journal of the Optical Society of America.

9830A Helps Sample Martian Soil

by A. B. Sperry, Hewlett-Packard
Calculator Products Division

What is the composition of Mars? Can the Red Planet support life, and, if so, what form of life? Two NASA Viking space vehicles are speeding silently along trajectories that will carry them some 460 million miles (740 million km) to help answer these questions that have puzzled scientists for centuries.

Launched from Cape Canaveral Air Force Station, Florida, on August 20, 1975, Viking I was blasted out of the Earth's atmosphere by a Titan IIIE/Centaur vehicle. It will land its surface module on Mars on July 4 this year, during the height of the U.S. Bicentennial celebration.

Viking II was launched 20 days later, September 9. It is scheduled to dispatch its lander to the Mars surface on September 4, 1976, at a spot about 1000 miles (1610 km) from Viking I. The timing allowed using the same launching pad for both events. Experience gained from the first launch was used to optimize the second one, and experience from landing the first module on Mars will allow any desired programming changes to be made before the second one touches down.

Each Viking spacecraft, consisting of an orbiter and lander, will orbit Mars for at least 10 days before dispatching its lander, gathering data for a precise landing and checking the preselected landing sites, which have not been viewed since the last Mariner flight in 1971-72. The landing sites are in areas of geological interest and are also those most likely to be able to support some form of life. When a landing area has been verified as suitable, the lander module will separate from the oribiter while traveling at an initial velocity of about 10,300 miles (16.600 km) per hour.

LANDING PROCEDURE

Because the Viking spacecraft will reach Mars when it is about 206 million miles (331+ million km) from Earth, it will take a radio signal and response about 37 minutes to make the round trip — far too long to allow emergency corrections during the lander's entry. This necessitates completely automating the landing procedure, which is controlled by a computer in the lander and guided by a radar altimeter and other devices. This makes Viking the most completely automated mission NASA has ever conducted.

During the landing process, the module's velocity is first slowed by friction between it and the Martian atmosphere. An ablative shield on the lander burns away, dispersing the friction-caused heat, and is then discarded. When the module's speed has decreased to about 560 miles (900 km) per hour, a small mortar is fired to eject a parachute. The mortar is used because a pilot chute would not have enough drag to open the main parachute in the thin atmosphere. The parachute is jettisoned at about 4,000 feet (1220 m) from the surface, after it has slowed the lander to around 138 miles (220 km) per hour. Finally, three landing engines are ignited to slow the lander to about 5.5 feet (1,68 m) per second when it reaches the Mars surface. Each lander leg has a switch, so that the engines will be shut off when the first leg touches down, minimizing soil disturbance by the engine plumes. Shock-absorber assemblies in the legs contain honeycomb aluminum cylinders that are crushed to some extent by the landing shock. Each leg has a stroke gauge calibrated to indicate the degree of compression, which is measured when the lander's cameras photograph the gauges.

The landers will conduct investigations during entry, as well as after landing. These will provide data on the concentration of ions in the upper atmosphere, atmospheric pressure, composition, temperature, and density. From previous Mariner missions and other sources, we know that the Martian atmosphere is about one percent the density of Earth's and is chiefly carbon dioxide, with traces of other gases. The Viking measurements and observations will add greatly to our store of knowledge about this and other characteristics of Mars.

SURFACE SAMPLER

The surface sampler boom assembly comprises a 10-foot retractable boom and a powered collector head. The retractable boom is made of two thin, edge-welded stainless steel strips, forming an assembly that rolls up similar to a steel measuring tape when it is retracted. The strips curve laterally when the arm is extended, making it rigid.

The boom can rotate through 302 degrees in azimuth and from 32 degrees above the horizon to 43 degrees below it. The collector head on the end of the boom contains a small, shovel-like mechanism that can dig with a force of 30 pounds (13,6 kg), backhoe with a force of 20 pounds (9 kg), and lift with a force of 5 pounds (2,25 kg). It scoops up about 50 cc of soil at a time. A backhoe mounted beneath the collector head will make trenches in the soil to permit the collection of subsurface soil samples (below 5 cm).

Magnets mounted on the backhoe will collect magnetic particles of soil for study. For examination of the magnets and collector head at close range, the collector head can be positioned in front of a 4x magnifying mirror that reflects into the cameras.

The collector head delivers soil samples to three inlets on top of the lander that process soil for the biology instrument, the gas chromotograph mass spectrometer, and the X-ray fluorescence spectrometer. The collector head has a screen on top, so that when it is turned over the soil samples are sifted into the instrument inlets, culling out particles to large for the instruments to handle.

SOIL INVESTIGATION

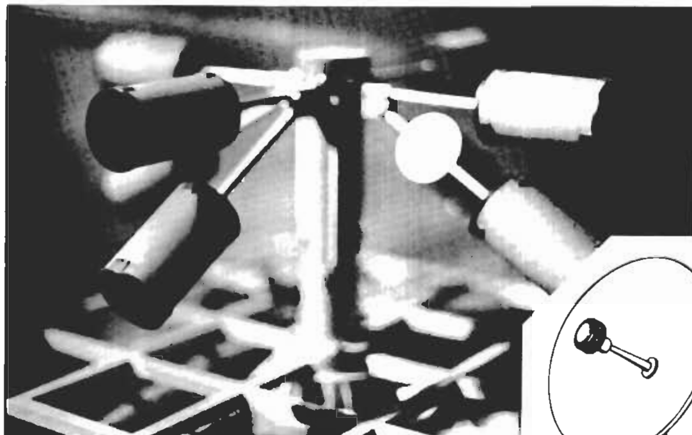
Biology Investigations

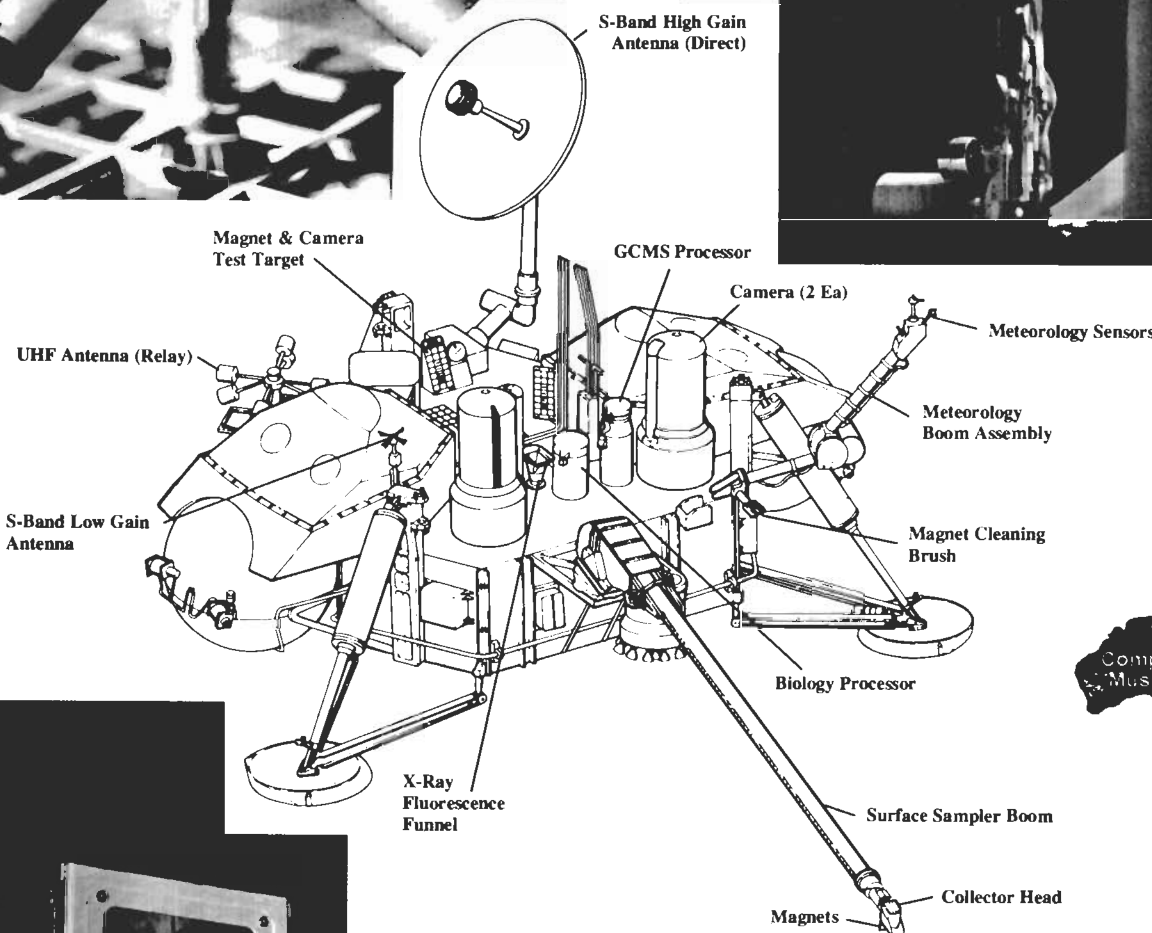
The biology instrument conducts three experiments to search Martian soil samples for living microorganisms. These are a pyrolytic release experiment, a lableled release experiment, and a gas exchange experiment.

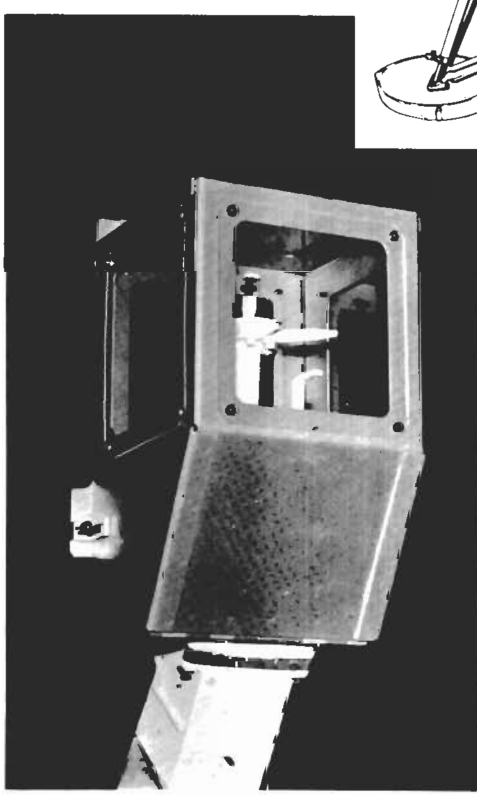
The pyrolytic release experiment looks for microorganisms functioning by photosynthesis or chemotrophy (organic response to chemicals). The labeled release experiment looks for signs of metabolism. The gas exchange experiment detects living organisms by measuring changes in the gases in a closed environment.

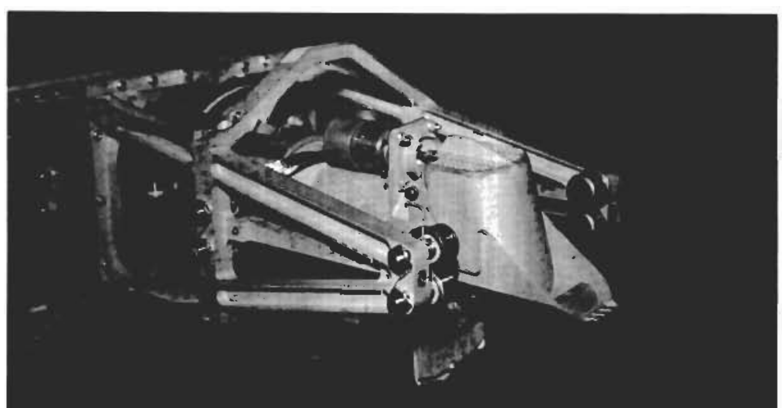
The gas exchange experiment can be performed twice; the pyrolytic release and labeled release experiments can be performed four times.

Depending on the results of these experiments, scientists hope to determine the range of conditions under which life can form and evolve and whether carbon-based life as it exists on Earth is the only form of life that can exist.




S-Band High Gain Antenna (Direct)
Magnet & Camera Test Target
GCMS Processor
Camera (2 Ea)
UHF Antenna (Relay)
Meteorology Sensors
Meteorology Boom Assembly
S-Band Low Gain Antenna
Magnet Cleaning Brush
Biology Processor
X-Ray Fluorescence Funnel
Surface Sampler Boom
Collector Head
Magnets





Molecular Analysis (Organic) Investigation

The gas chromatograph mass spectrometer (GCMS) analyzes the soil for organic chemicals and analyzes the atmosphere at the surface. This investigation may reveal any existence of past life on the planet. Soil samples are pulverized and heated in steps to vaporize different organic compounds. The GCMS analyzes these vapors, and results are digitized and transmitted to Earth for decoding and interpretation.

These experiments may provide an understanding of the quantity and complexity of organic compounds at the surface of Mars, enabling scientists to determine their origin and whether they are capable of evolving into living systems. This knowledge will also help scientists develop future biology experiments based on Martian rather than terrestrial conditions.

Inorganic Chemical Investigation

The X-ray fluorescence spectrometer is the detection instrument for the inorganic chemistry investigation. It was chosen to analyze the inorganic chemical elements in the Martian soil because it is able to analyze most elements known to exist in the solar system. X-ray bombardment of the soil sample causes electron transfer and release of energy, which is different for each element. This is detected and recorded for transmission and interpretation.

With the results of the inorganic chemistry experiment, the scientists hope to:

- Gain insights into past, present, and future planetary evolution processes,
- Learn more about planetary differentiation, the process that moves massive areas of the planets to create landforms,
- Learn more about erosion and deposit processes on other planets, and
- Identify the interaction between the atmosphere and the solid surface on Mars.

COMPUTER CONTROL

A general-purpose digital computer processes control information aboard the lander. It contains two identical 18k-word channels, which are command-selectable, to permit operation even if one channel fails.

The computer can operate the lander without commands from Earth, as it will for the entire landing sequence. The computer has enough commands stored in its memory to operate the lander and its instruments for 60 days after it lands. Once communication with Earth is established, these commands can be modified and updated if necessary.

THE 9830A'S ROLE

The Viking Mission Control Center is located at the Jet Propulsion Laboratory in Pasadena, California. A full-scale model of the Viking lander has been set up to provide a ground simulation facility for scientists and engineers to use to develop flight command sequences for the lander cameras and surface sampler. The surface sampler subsystem on the Viking Science Test Lander is operated by an HP 9830A in lieu of the flight lander computer.

James Kaehler, design team leader for the surface sampler control assembly electronics at Martin Marietta Aerospace (the lander prime contractor) Denver, Colorado, chose the HP 9830A Calculator as the encoder for the surface sampler commands. He designed a simple interface and multiplexer to connect the 9830 with the control console. As he puts it, "The 9830 was versatile enough that it essentially plugged right into the existing console, which had never been planned to be compatible with it. And the thing that was so impressive was the first time we tried to interface it with the surface sampler system, it took just two hours from scratch to be able to talk to the sampler and get it to execute commands."

The commands are encoded by inputs through the 9830 keyboard, multiplexed as 16-bit binary command strings into the console, and transmitted to the Viking using microwave signals. The 9830's interfacing capability permits receipt of engineering data from the surface sampler in usable form, with no manual conversion from binary coding. As a backup in case of problems, the control console can be used to operate the surface sampler by manual commands independent of the 9830.

OTHER SURFACE INVESTIGATIONS

Imaging Investigation

Two cameras mounted on top of the lander produce electrical images of a broad range of subjects, from landscape panoramas to closeups of items on the lander. They are equipped with variable focus and depth of field and are capable of viewing the entire circumference of the landing site from the lander footpads, 60 degrees below the horizon to 40 degrees above the horizon. Pictures can be taken in color, black and white, and infrared with excellent resolution.

The imaging investigation will assist in all of the other investigations. It will give us the first close-up pictures of Mars. Additionally, it may obtain evidence that either confirms or denies the hypothesis that life exists or has existed on the planet, gain a clearer understanding of the Martian atmosphere and environment, and obtain data providing clues to the origin of Mars and its age to correlate with similar data now available about the Moon and Earth.

Meteorology Investigation

The meteorology instruments mounted on the lander and suitably protected from the environment will periodically measure pressure, temperature, and winds during each Martian day.

The data from this investigation will give insight into how Martian wind varies with time, how wind transports dust that covers and uncovers surface markings, and how temperatures near the surface vary daily. These measurements will help scientists understand Martian weather and use this information to make inferences about weather on Earth.

Seismology Investigation

The lander will rest solidly on the Mars surface on its three legs, so that any seismic motion will be transmitted to the seismometer. This investigation will analyze data on volcanic activity, planet structural shift, and meteorite impacts on the planet's surface.

Using the analysis results, scientists may define sources of continuous background seismic activity, determine whether Mars is tectonically active, and develop a model of the planet's internal structure and composition. It will answer the question of whether Mars has a crest and a core and if its mantle is similar to Earth's.

Magnetic Properties Investigation

An array of small permanent magnets mounted on the surface sampler collector head backhoe will collect magnetic particles when the head is pulled through the soil. It will be viewed by the cameras, either directly or by using magnifying mirrors on top of the lander. The pressure of various iron-bearing materials will help scientists study the separation, or differentiation, of minerals that occurred on the planet.

Physical Properties Investigation

Various engineering measurements, the leg stroke gauges, images of retrorocket-soil interaction using boom-mounted mirrors, and calculations from other experiments will all contribute toward knowledge of the character of the Martian soil and increased understanding regarding the presence or absence of life on the planet. Bearing strengths, cohesion, porosity, and grain size are some of the characteristics to be estimated.

COMMUNICATIONS

Two-way radio links are provided between the Viking orbiter and Earth and between the lander and Earth to carry the encoded commands, ranging data, telemetry, and science data. Both systems operate on S-band frequencies, and the orbiter-Earth link also operates on X-band. A third link operating in X-band is used to transmit telemetry and data from the lander to the orbiter, which relays this information to Earth when direct communication from the lander to Earth is not possible.

The lander has a tape recorder to permit continuous data acquisition when transmission is not possible because of capsule ionic shielding during descent, when data is received faster than it can be transmitted, or whenever communication is not possible between the lander and either the orbiter or Earth. Recorded lander data can be transmitted later to the orbiter via S-band signals at speeds up to 16,000 bits per second.

The lander UHF radio has three commandable output power levels from 1 watt to 30 watts to allow high-speed data transmission and yet conserve power. Complex data transmission, such as photographic and telemetric data, requires high data rates and high power.

The lander S-band subsystem can transmit high-volume scientific, photographic, and telemetric data directly to Earth, and it can receive commands from Earth. The system has two identical receivers, one of which is connected to a low-gain, crossed-dipole antenna. The other receiver is connected to a 30-inch parabolic high-gain antenna to detect the Earth-to-Mars ranging signal and to act as a backup command receiver when the antenna is pointed toward the Earth.

The Deep Space Network, a complex system operated for NASA by Jet Propulsion Laboratory (JPL), Pasadena, California, is the Earth link for Viking communications. It has six 85-foot parabolic antennas and three 210-foot dish antennas. Both systems are evenly spaced around the Earth, so that as Earth rotates, part of the system will always have Mars and the spacecraft in view. The 210-foot system must be used to receive the Viking data at high rates; the 85-foot system receives it at lower rates.

The received data is amplified and processed by computer into high-speed data blocks, then transmitted by satellite or cable to the Goddard Space Flight Center, Greenbelt, Maryland. From there it is routed by land lines or microwave to JPL where the mission control flight teams, scientists, and engineers are located. Messages are sent from Earth to Viking using the same system, but with the procedure reversed.

CONCLUSION

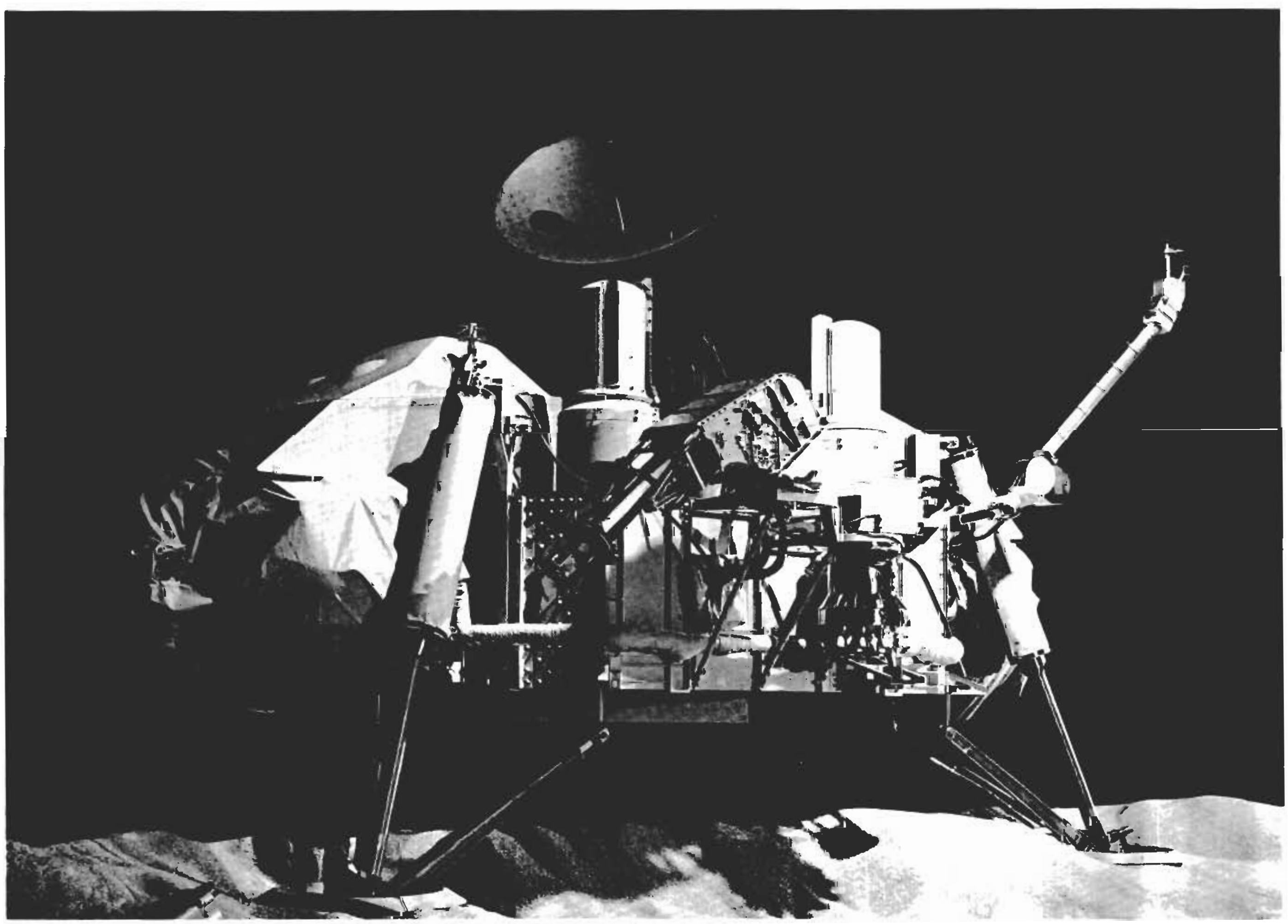
The Viking mission is expected to contribute more toward understanding the characteristics of Mars in the first two weeks of surface operation than has all of the previously acquired knowledge about the planet. Each lander will operate on the surface for a minimum of 90 days, and each orbiter is designed to operate for at least 120 days. The mission could remain operative well into 1977. Adjacent to the Von Karman Auditorium at the Jet Propulsion Laboratory, the public can examine a working model of the lander and visit a comprehensive space museum.

Acknowledgements

The author acknowledges with thanks NASA's permission to take photographs at its Science Test Lander Facility at Jet Propulsion Laboratory and permission from Martin Marietta Aerospace to use and copy in part the information from its book, *The Viking Mission to Mars*.

Editor's note: For further information on the Viking mission to Mars, please contact:

Public Relations Department
Martin Marietta Aerospace
P.O. Box 179
Denver, Colorado 80201
U.S.A.



Forum

We are writing in regard to a user group for program exchange of 9815A material.

If such a group is presently being formed by other *KEYBOARD* readers, we would like to be added to their list. If such a group hasn't been started, we would like to hear from any 9815A users who might be interested in helping to start a program exchange.

Garrison P. Smith, P.E.
Chas. P. Smith Associates, Inc.
P.O. Box 428
Orange, Texas 77630

Some responses to Professor Maeder's question on "WAIT 3000 INPUT X" in last issue's Forum that came to the editor have posed interesting and clever methods to cope with this problem. Since Professor Maeder's original question has caught the interest of a number of our readers, three of these responses are printed below.

Dear Editor:

In reply to the Forum article, Vol. 8, No. 1, Prof. D. G. Maeder posed a question in regard to waiting a certain time for a data entry reply by the user.

Some time back you published a programming tip from me that concerned using the STAT function of the Extended I/O ROM. If Prof. Maeder is fortunate enough to have the ROM, a simple solution can be achieved. The following is a hypothetical case in which the user has a certain time to reply to a display question or the variable A will retain its current value.

```
10 A = 1
20 DISP "ENTER DEGREE OF OFFSET ANGLE"
30 WAIT 3000
40 IF (STAT10 # 15) THEN 60
50 INPUTA
60 . . . . . . .
```

With this simple sample the status of the internal tape cassette is checked and if a certain condition is not met, i.e., door open, the program skips the input statement. The user is able to respond by opening the cassette door and then entering the value for A.

Hope this solves his problem.

David Ripley
New Mexico State Highway Dept.
Engineering Computer Section
P.O. Box 1149
Santa Fe, New Mexico 87503
U.S.A.

Dear Editor:

. . .The problem of reducing data entry for multiparameter computations when only a subset of the parameters require re-entry can be solved in two similar ways. The first method is used if the parameters to be changed are different each time the calculation is performed; the second is used if the same parameters are to be changed after every calculation.

First of all, I find it most convenient to store parameters in matrix form. This allows for easy storage on tape or mass memory, avoids accidental double usage of variables and simplifies writing of data entry and output programs.

Suppose we have a program utilizing 10 parameters with parameter names NAME1, NAME2,...etc. The first program shows how I would enter the data. Although the parameter names and values could have been printed after the input statement, I have left the printout to a separate loop. Keeping the parameter names in data lists simplifies multiple outputs such as on the printer and plotter, since parameter names need only be typed in once.

The second program is a simplified version to show the basic technique. Lines 50 - 100 are used to enter all parameters. Following Line 100, one would branch to the computational part of the program.

In Line 110, the user keys in the names of those parameters he wishes to change. The entry loop in Lines 150 - 210 is used to enter the specified parameters, skipping those not specified in the parameter list.

Program 3 is a more sophisticated version that avoids complications with the POS function. Entries in the parameter list are separated by commas. Each entry is considered in turn in the input loop in Lines 200 - 260. Programs 2 and 3 can be used if the parameters to be changed are different each time.

A slight modification of Program 3 can be used if the parameters to be changed are the same each time. In Line 10 an integer matrix (e.g. A) with the game dimensions as P is specified. This matrix is initialized to ones. Lines 230 - 250 are deleted and replaced by:

```
230 A(I) = 0
```

Following Line 290 another data entry loop is written:

```
300 RESTORE 20
310 FOR I = 1 to 10
320 READ A$
330 IF A(I) THEN 360
340 DISPA$;
350 INPUT P(I)
360 NEXT I
370 END
```

By introducing an auxiliary input, these solutions do not quite satisfy the problem; however, the overall reduction in the total number of data entry steps I think more than justifies the inclusion of the extra entry.

Klaus Schaedlich, P. Eng.
Special Projects Engineer
Air Quality & Meteorology Section, Air Resources Branch
Ontario Ministry of the Environment
135 St. Clair Avenue West, Suite 100
Toronto, Ontario M4V 1P5
CANADA

```
10 DIM P[10],A$[10]
20 DATA "NAME1","NAME2","NAME3","NAME4",
        "NAME5","NAME6","NAME7","NAME8"
30 DATA "NAME9","NAME10"
40 RESTORE 20
50 FOR I=1 TO 10
60 READ A$
70 DISP A$
80 INPUT P[I]
90 NEXT I
100 RESTORE 10
110 FOR I=1 TO 10
120 READ A$
130 PRINT A$;P[I]
140 NEXT I
150 END
```

```
10 DIM P[10],A$[10],B$[100]
20 DATA "NAME1","NAME2","NAME3","NAME4",
        "NAME5","NAME6","NAME7","NAME8"
30 DATA "NAME9","NAME10"
40 RESTORE 20
50 FOR I=1 TO 10
60 READ A$
70 DISP A$;
80 INPUT P[I]
90 PRINT A$;P[I]
100 NEXT I
110 DISP "PARAMETER LIST";
120 INPUT B$
130 IF B$="STOP" THEN 220
140 RESTORE 20
150 FOR I=1 TO 10
160 READ A$
170 IF  NOT POS(B$,A$) THEN 200
180 DISP A$;
190 INPUT P[I]
200 PRINT A$;P[I]
210 NEXT I
220 END
```

```
10 DIM P[10],A$[10],B$[100]
20 DATA "X","X1","NAME3","NAME4","NAME5",
        "NAME6","NAME7","NAME8"
30 DATA "NAME9","NAME10"
40 RESTORE 20
50 FOR I=1 TO 10
60 READ A$
70 DISP A$;
80 INPUT P[I]
90 PRINT A$;P[I]
100 NEXT I
110 DISP "PARAMETER LIST (P1,P2,...)";
120 INPUT B$
130 P1=0
140 IF B$="STOP" THEN 300
150 P=POS(B$,",")-1
160 IF P>0 THEN 190
170 P=LEN(B$)
180 P1=1
190 RESTORE 20
200 FOR I=1 TO 10
210 READ A$
220 IF B$[1,P]#A$ THEN 260
230 DISP A$;
240 INPUT P[I]
250 PRINT A$;P[I]
260 NEXT I
270 IF P1 THEN 300
280 B$=B$[P+2]
290 GOTO 150
300 END
```

Dear Editor:

In response to Prof. D.G. Maeder's question. . . .

A statement of this type may develop inefficiency in machine operation and possible improper program execution. Many applications here at Torrington consist of 40 or more inputs and, needless to say, require considerable time to input. If a wait time is added to several statements (for example, 3 sec x 20 = 60 seconds wait time), additional time will be required to complete inputs, resulting in undesirable loss of time.

Second, we feel it most undesirable to establish a condition where the possibility of lack of response or hesitancy on the operator's part may allow improper program input and logic flow.

The Torrington Company has handled this problem on our 9830's with a function statement using the method as outlined on the attached sheet. You will notice the variable A, B, C need not be initialized to a value (example, A in FNTA). Simply enter the first value, and on repetitive statements enter 0 to retain the original value. No extra inputs are required, and actual input time may decrease in some instances.

If 0 is to be used as an input value, Y in Line 120 can be changed to be equal to any other value not used in the program (example, Y = 100 or Y = -1, etc.).

I agree wholeheartedly with Prof. Maeder's comments on the "Forum" feature. How else can we analyze and be more informed of new ideas than through *en rapport* with our fellow 9830 users?

Joseph Brown
Sr. Methods Time Analyst
The Torrington Company
59 Field Street
Torrington, Connecticut 06790
U.S.A.

```
10 REM    DEMO PROGRAM
20 DISP "INPUT VALUE #1     ";
30 A=FNTA
40 DISP "INPUT VALUE #2     ";
50 B=FNTB
60 DISP "INPUT VALUE #3     ";
70 C=FNTC
80 PRINT
85 PRINT "A=";A;"B=";B;"C=";C
90 GOTO 20

100 DEF FNT(X)
110 INPUT Y
120 GOTO Y=0 OF 140
130 RETURN Y
140 RETURN X
150 END

RUN
INPUT VALUE #1      ?1.23789564
INPUT VALUE #2      ?8.25478621
INPUT VALUE #3      ?-7.25781467

A= 1.23789564   B= 8.25478621   C=-7.25781467

INPUT VALUE #1      ?0
INPUT VALUE #2      ?2.58711442
INPUT VALUE #3      ?0

A= 1.23789564   B= 2.58711442   C=-7.25781467
```

THE Crossroads

Newton-Raphson Iteration

by John Nairn, PhD Hewlett-Packard Calculator Products Division

"It is the mark of an instructed mind to rest satisfied with the degree of precision which the nature of the subject admits, and not to seek exactness where only an approximation of the truth is possible."

Aristotle

It is not a rule of mathematics (or even a sound principle) that the difficulty of solving an algebraic equation is related to how "complex looking" that equation might be. Consider the following three equations:

$$\sqrt{x+1} + \frac{2-\sqrt{x}}{\sqrt{x+1}} = \sqrt{x-2\sqrt{x}}$$

$$\cos(x) = x$$

$$e^x + x = 0$$

Despite the formidable appearance of the first equation, its solution is obtained by merely applying the elementary rules of algebraic manipulation. The latter two equations require the use of far more advanced tools of algebra to obtain their solutions. Or to take a simpler example, many persons who might remember the pencil-and-paper method for taking the square root of a number would have to resort to a set of tables or a calculator to find the cube root of that number.

And even these problems are relatively simple compared to those that have a way of cropping up in many everyday programming jobs. Obtaining a programmed solution to a real-world problem usually breaks down into four major steps: (1) analyze the problem, (2) describe the problem by a set of equations and procedures, (3) solve the equations, and (4) program the solution. (We will be more optimistic than we have a right and leave out the step of debugging the program!)

Each step in the above process has had countless books and articles devoted to it, and a complete understanding of any of these steps could demand years of study. The everyday programmer needs a working knowledge of the fundamentals of all of them and is always glad to pick up tools in each area from time to time. An extremely useful tool for step (3), and one that does not appear to be widely known among programmers, is a method known as Newton-Raphson iteration. One might object that an iterative solution is only an approximate solution, and indeed this is true. But when a computer represents each value to a certain number of digits only and the approximation is correct to that number of digits, the use of the analytic result (if it could be found at all) would not provide more accuracy. At this point the pure mathematician might object that a knowledge of the analytic solution might cast some light on a means of gaining more insight into the problem being programmed, and I cannot deny that. But the immediate task of many programmers is to generate a program to solve the given problem as quickly and efficiently as possible, and any tools that will expedite that task can be used in clear conscience.

So, what is this Newton-Raphson technique? I will first present the method in its general form, followed by some simple examples of its use. Then a few words of caution concerning its range of applicability will be given. The algebraic equation to be solved is first put in the form

$$f(x) = 0 \,. \qquad (1)$$

That is, all terms are collected on one side of the equation, and that side expresses the general function of the variable x for which we are trying to solve. Next a guess (x_0) of the solution is made. The value of x_0 is merely the starting point for the iteration process, and in many cases it does not matter how bad the guess is. Once having exercised your clairvoyant powers and made the initial guess, the next approximation to the true value of x is obtained by use of the formula:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \qquad (2)$$

where f'(x) is the derivative of the original function, f(x). That is, if we take the current "best guess" of the solution and subtract the function evaluated at that point divided by the derivative of the function evaluated at that point, we get the "next best guess" to the solution. This process is repeated until the next guess (iteration) and the last guess are the same value to the number of digits of accuracy required for the problem. You probably have noticed by now that the method does require that you be able to differentiate the function f(x). But often this is easier to do than to solve the original function, f(x), analytically. A few examples should make clear how the method is used.

As a first example, we will use the method to find the square root of a given number, N. We are looking for a solution to the equation $x^2 = N$, or to put it in the standard form of equation (1),

$$f(x) = x^2 - N = 0 \,. \qquad (3)$$

The derivative of f(x) is found to be f'(x) = 2x, so that the iteration equation (2) for this problem becomes

$$x' = x - \frac{x^2 - N}{2x} = \tfrac{1}{2}(x + N/x) \qquad (4)$$

where for simplicity we have adopted the notation x for the current guess and x' for the next guess. Now let's use the result of equation (4) to find the square root of 10. Let's also assume that your ability to guess is no better than mine, and we choose 1 as the first guess. We then have

$$\begin{aligned} x1 &= \tfrac{1}{2}(1+10/1) = 5.50000\ 00000 \\ x2 &= \tfrac{1}{2}(x1 + 10/x1) = 3.65909\ 09091 \\ x3 &= \tfrac{1}{2}(x2 + 10/x2) = 3.19600\ 50819 \\ x4 &= \tfrac{1}{2}(x3 + 10/x3) = 3.16245\ 56228 \\ x5 &= \tfrac{1}{2}(x4 + 10/x4) = 3.16227\ 76652\,. \end{aligned} \qquad (5)$$

Each successive iteration gets closer to the true answer of 3.1622776602 for the square root of 10. In fact, the next pass would have given x6 as the exact root to the number of decimal places we have carried in this example.

We could just as easily find the cube root of a number by starting with the function $f(x) = x^3 - N$ for equation (3). Can the reader find the general form of the iterative equation (4) for the k_{th} root of a number before it is given next time?

Square-root evaluation serves as an example for the Newton-Raphson iteration method, but it could be considered almost trivial since most calculators have a square-root function built in. And higher roots are easily obtained through the relationship $x^{1/n} = \exp(\ln(x)/n)$. So let's try a more complex example from real-world programming. In fact, this is an actual problem that I was asked to help solve not too long ago.

A production process is composed of a series of steps, each with a yield, y_n, giving a total yield for the entire process of $Y = \prod y_n$. Here the symbol $\prod$ is a shorthand notation for the product of all the individual y_n values. The goal is to increase the overall yield of the process by a factor, p, so that a higher overall yield, $Y' = (1 + p)\,Y$, will be obtained. The question is, how much will the individual yields, y_n, for each step have to be increased to obtain the desired overall yield?

One approach might be to say that if there are k steps in the process, each step is to increase by a factor of the k^{th} root of $(1 + p)$. Then the new yield, Y', which is the product of all the new individual yields, would be

$$Y' = \prod y'_n = \prod (1 + p)^{1/k} \bullet y_n = (1 + p)Y \quad (6)$$

which is the desired result. This, however, is not a very realistic result. The individual yields will vary in value from almost zero to one (where one means a 100% yield). Obviously, a step in the process that currently has a yield of almost one cannot be asked to carry as large a burden as for smaller-yield steps. Indeed, the increase factor used in equation (6) may cause some yields already close to one to be required to go over 100%. The burden should be more realistically distributed in such a way that the low-yield steps are required to make a large relative increase to obtain the desired overall increase. At this point the problem must be analyzed and a decision made as to how this burden is to be distributed. Let us say for the sake of this example that we choose a linearly distributed burden; that is, the desired increase for each step in the process is larger in portion to its distance from one (a total or 100% yield). This is expressed by the equation

$$p_n = X(1 - y_n) \quad (7)$$

where p_n is the factor by which y_n must increase. Notice that as y_n approaches one, p_n approaches zero, which, as we saw, is realistic. X is some as-yet-undetermined value that is the increase factor required for very small yields. Since we know all of the current yields, y_n, in the process, if we could find the value of X we would know the increase factor, p_n, for each step required to give the desired overall yield, Y'. Combining equation (7) with the two constraints on the total yield, namely $Y' = (1+p)Y$ and $Y' = \prod y'_n$, we find the relation

$$(1 + p) = \prod (1 + X(1 - y_n)) = \prod (1 + X \bullet s_n) \quad (8)$$

where we will use the symbol s_n for the quantity $(1 - y_n)$. Since we know the values of p and the s_n's, equation (8) is merely a polynomial in X when all the factors are multiplied out. The degree of this polynomial, however, is equal to the number of steps in the process and could lead to great difficulty in solving for X analytically if the number of steps is large. (By "large" I mean more than 5, which is small for a real manufacturing process.)

So after all these fancy equations, where do we stand? We have analyzed the problem, written the equations, and would like to move on to writing the program to find the answer -- the increase factor for each step in the process. But we are at a standstill because we cannot solve the equation (8) for the value of X. This is where Newton-Raphson iteration comes to our rescue. If we can write a function for X in the form of equation (1) and differentiate it, we are home free; so let's try.

Equation (8) is simply transformed into the form $\prod (1 + X \bullet s_n) - (1 + p) = 0$. But since this function involves a product with an indefinite number of factors (remember that we are writing a program for a generalized process with any number of steps), finding the derivative could be quite a job. Recalling that the log of a product is the sum of the logs, we can take the log of both sides of equation (8) and then put it into the form of equation (1), giving

$$f(X) = \Sigma \ln(1 + X \bullet s_n) - \ln(1 + p) \,. \quad (9)$$

This gives us for the derivative

$$f'(X) = \Sigma s_n/(1 + X \bullet s_n). \quad (10)$$

We now can write a routine that will make an initial guess for the value of X and iterate to the final value of X, given the current yields and desired yield increase, p. Even though we never were able to obtain an equation that would calculate X directly from the given p and s_n values, we still are able to complete the program and obtain the final answers, which are the y'_n values. The reader might wish to complete the remaining steps to convince himself or herself that we now have all the necessary pieces for the final program.

When adding any new tool to your collection, it is wise to know the limits of that tool. The only thing worse than not being able to get answers is getting bad answers (especially if they look believable). Therefore, a few words of caution are in order concerning the Newton-Raphson method. In essence, we are finding the roots of the equation f(x); that is, the points where f(x) crosses the x-axis. In our example for finding square roots, the function $f(x) = x^2 - N$ crossed the positive x-axis in one place only, so it didn't matter how bad a first guess we made (as long as it was greater than zero). A new situation arises if the function has more than one root; the root found by the method will depend on the initial guess. Even then, however, the method will yield some root that may be extracted from the original function, a new f(x) formed, and the process repeated, to find all the roots. Trouble also arises when one of the roots is a multiple root; that is, the same value of x appears more than once in the function's set of roots.

The references listed below discuss some of the limitations of the method, and Stanton in particular goes into spotting and dealing with functions that are not well-behaved. Even with these limitations, however, the Newton-Raphson iteration technique is a valuable tool for many algebraic roadblocks that arise in programmed solutions to problems and should be kept in a handy place in the programmer's toolkit.

The last *Crossroads* article dealt with permutations and combinations, and I asked any readers who had real applications for these to tell me about them. I would like to thank Robert C. Carter of Atlanta who wrote to me about his application in communications systems. His situation involves transceivers on different frequencies installed in close proximity to each other on the rooftop of a building. When certain linear combinations of these frequencies are within a given bandwidth of the frequency of another of these transceivers, interference may occur. Carter's program uses combinations to look for these interference relations among the existing frequencies and quickly locates problem sources that were virtually impossible to find before the use of his program.

References

Korn and Korn, MATHEMATICAL HANDBOOK FOR SCIENTISTS AND ENGINEERS (McGraw-Hill, 1961), p. 630.

Kuo, Shan S., NUMERICAL METHODS AND COMPUTERS (Addison-Wesley, 1965), p. 90.

Stanton, R.G., NUMERICAL METHODS FOR SCIENTISTS AND ENGINEERS (Prentice-Hall, 1961), p. 84.

programming tips

One-Line X/Y Integration (9820A)

Erik Siwertz of the Institut National de la Recherche Agronomique in Thonon-Les-Bains, France, submits this interesting programming tip and a challenge to other 9820A users.

What do you think about this one-line X/Y integration for the 9820A (either with or without a Math ROM)? I'd like to see it published in Programming Tips and see if someone can match it. The tip works with increasing or decreasing values of X (positive or negative), two flags (0 and 13), and only the alpha register.

```
0:
ENT "X,Y";(Y+B)/2
)*(X-A)+Z)(FLG
0-FLG 13)+C→C;X→
A;Y→B;SFG 0;GTO
0;IF FLG 13;PRT
C⊢
1:
END ⊢
```

Incrementing Logarithmic Scales (9820A)

James Lovell of AIL Division of Cutler-Hammer in Long Island, New York, submits this helpful programming tip.

I have had a number of occasions in which I needed a logarithmic scale. Without thinking too deeply, I simply incremented the independent variable in the usual way, say A + 1→A. Unfortunately, with a logarithmic scale the increments get closer and closer, and one never knows whether to stay and wait or go for a cup of coffee while it plots. Recently I realized that if I simply increment with A + A→A, the independent variable doubles in value with each step, the steps along the independent variable axis are uniform on the plotter, and the plot is soon over. The plots have sufficient detail for my purposes, but variations on this could give more or less detail, such as A + A/2→A and A + 2A→A. Now someone else can have the machine while I have my coffee.

Sorting and Pairing Numbers (9830A)

Andrew Zinn, Scott Wulfe, and Jack Ligon of Robert E. Lee High School in San Antonio, Texas, submit this programming tip, which should be very useful in a number of applications.

We have an idea for sorting four-digit numbers on the 9830 and, if desired, pairing them with alphanumeric data. This could be used in class ranking programs, for example.

First, all data must be in a similar range; i.e., between 0 and 1, 1 and 10, etc. Next, using the first four significant digits, enter the data element as a line number (3.459 would be entered as Line 3459). Type in some dummy statement (the program will never be run, so the statement should be as short as possible to conserve memory), or, if alphanumeric data is to be sorted, the following statement, for example, would suffice:

3459 A$ = "JANE SMITH"

Press END OF LINE and continue entering data in this fashion. When all data are entered, merely LIST the program lines to print out the data elements in order from least to greatest. If alphanumeric data is included, executing REN 1, 1 will produce a listing of the data, in order from least to greatest, with the first line being 1 and all lines consecutive integers, when LIST is executed.

Speeding Cassette Tape Access Time (9830A)

Our thanks to Thomas Krantz of the Bermuda Division of Palisades Geophysical Institute in St. Davids, Bermuda, for the following programming tip.

A decrease in cassette tape access time can be obtained by using a different file marking routine than the method described in the manual.

To illustrate the difference, two tapes were marked with ten usable files of 3000-word length. Figure 1 illustrates the marking method described in the calculator manual, which I will refer to as the "normal" method. The different marking routine, referred to as the "modified" method, is the same as the normal method, except that a minimal size file (4 words) is inserted before each usable file (see Figure 2).

The seventh usable file, File 7 of the normal tape and File 15 of the modified tape, was selected as the reference file. Time measurements with a stop watch were made between the execution of the LOAD 7, 10, 10 command and the appearance of the first line of a dummy program in the display.

Prior to the LOAD 7, 10, 10 (15 for modified tape), the tape was positioned using the LOAD, FIND and REWIND instructions. Two sets of FIND commands were used, since the normal tape was sensitive to the position of the tape prior to the FIND command - rewinding the tape before executing the FIND command, and positioning the tape to the end, File 10 for normal and File 20 for modified, before executing the FIND command.

In all cases except one, access time of the modified tape was faster than the normal tape. For the 31 measurements made, the mean and standard deviation are:

normal tape	M = 83.0 sec	sd = 30.9 sec
modified tape	M = 61.2 sec	sd = 24.7 sec

The modified method does have its drawbacks:

1. It takes about 6 minutes longer to mark the tape.
2. File positions are not in direct order, but this can be adjusted for by using a conversion statement where modified file number = (2 x normal file number) + 1.

Our conclusion is that the modified method of tape marking wins hands down. The methods used to test it are by no means complete, but are good enough to warrant using the modified method for a while to see how it works for you.

```
MARK 10,3000

TLIST

 0    0    3000 0    0    0    0
 1    0    3000 0    0    0    0
 2    0    3000 0    0    0    0
 3    0    3000 0    0    0    0
 4    0    3000 0    0    0    0
 5    0    3000 0    0    0    0
 6    0    3000 0    0    0    0
 7    0    3000 0    0    0    0
 8    0    3000 0    0    0    0
 9    0    3000 0    0    0    0
10    0    3000 0    0    0    0
```

Figure 1

```
10 FOR I=1 TO 10
20 MARK 1,4
30 MARK 1,3000
40 NEXT I
50 MARK 1,4
60 END
```

Figure 2a

TLIST

0	0	4	0	0	0	0
1	0	3000	0	0	0	0
2	0	4	0	0	0	0
3	0	3009	0	0	0	0
4	0	4	0	0	0	0
5	0	3000	0	0	0	0
6	0	4	0	0	0	0
7	0	3000	0	0	0	0
8	0	4	0	0	0	0
9	0	3000	0	0	0	0
10	0	4	0	0	0	0
11	0	3000	0	0	0	0
12	0	4	0	0	0	0
13	0	3000	0	0	0	0
14	0	4	0	0	6	0
15	0	3000	0	0	0	0
16	0	4	0	0	0	0
17	0	3000	0	0	0	0
18	0	4	0	0	0	0
19	0	3000	0	0	0	0
20	0	4	0	0	0	0
21	0	4	0	0	0	0

Figure 2b

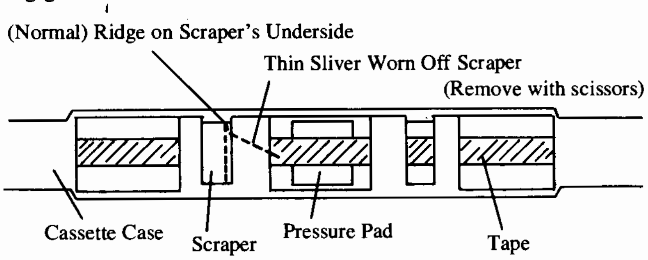
Recoverable Error 59's (9830A)

We received this programming tip via the 9830A User Group in Melbourne, Victoria, Australia. Mr. Ian Bird of I. T. Bird and Associates, Watson, Australian Capital Territory, Australia, submits this very useful tip.

Over 90% of the Error 59's I have experienced are recoverable without loss of data or program. Other users have verified this statement.

Recoverable errors appear to be caused by the tape wearing a very thin sliver of plastic off the tape scraper insert. This hair-like plastic fibre touches the read head and causes errors.

Cut the fibre with a pair of scissors and all is well. A magnifying glass could well be of assistance.



*A Magnifying Glass is Required to See the Fibre

Single-Line Cross Reference (9830A)

This tip is submitted by Dennis Eagle, Hewlett-Packard, Calculator Products Division.

There are times when it is very useful to know where in a program a given line is referenced. In the following example, if you change Line 14 to 16, the program cannot be run because Line 14 is referenced in Lines 10, 12 and 30.

```
1   X=1
10  GOTO 14
12  GOTO X OF 14,20
13  REM
14  FORMAT 6F5.0
20  END
30  WRITE (15,14)
```

In a program of 1000 steps or more, it becomes very difficult to see all the lines referencing a given line. The following procedure permits a user to obtain a cross reference for a given line.

1. Change the line number of the line in question.
2. Type: 9999 GOTO 9998

Press: END OF LINE

3. Be certain that there is not a Line 9998.
4. Type: REN
   Press: EXECUTE
5. ERROR 44 IN LINE XXXX will be displayed, where XXXX is a line number in your program. The line in question in step 1. is referenced in Line XXXX.
6. Change Line XXXX so that it now references the new line number established by 1.
7. Go back to 4. and continue to perform 4. through 6. until ERROR 44 IN LINE 9999 is displayed
8. Type: DEL 9999
   Press:EXECUTE

As an exercise, key in the example program.

1. Change Line 14 to Line 16.

```
1 X=1
10 GOTO 14
12 GOTO X OF 14,20
13 REM
16 FORMAT 6F5.0
20 END
30 WRITE (15,14)
```

2. Enter Line 9999

```
1 x=1
10 GOTO 14
12 GOTO X OF 14,20
13 REM
16 FORMAT 6F5.0
20 END
30 WRITE (15,14)
9999 GOTO 9998
```

3. If there is no Line 9998,
4. Type: REN
   Press: EXECUTE
5. ERROR 44 IN LINE 10 will be displayed
   Type: 10 GOTO 16

   Press: END OF LINE

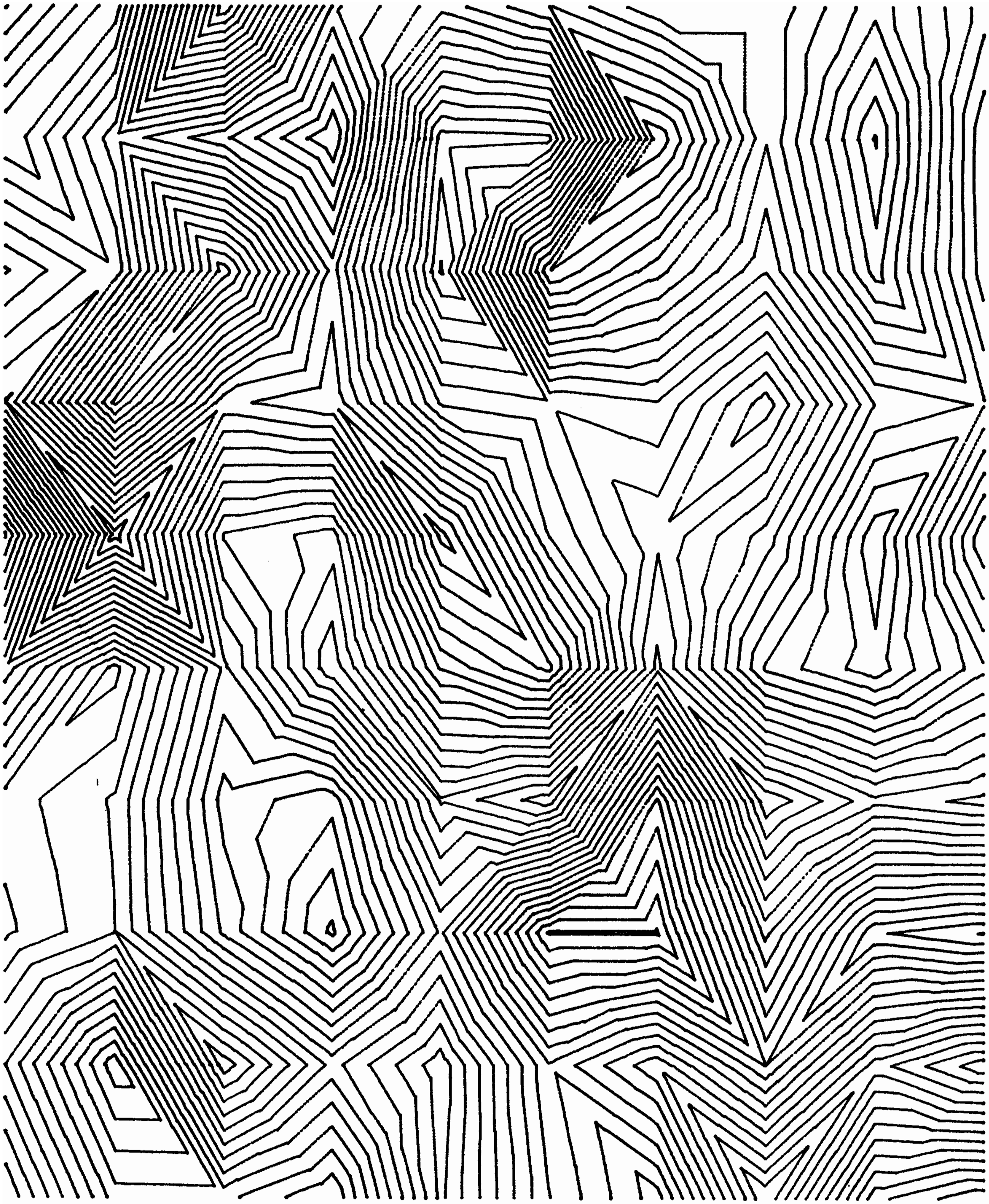
```
1  X=1
10 GOTO 16
12 GOTO X OF 14,20
13 REM
16FORMAT 6F5.0
20 END
30 WRITE (15,14)
9999 GOTO 9998
```

Repeat 4. and ERROR 44 IN LINE 12 will be displayed.
Type: 12 GOTO X of 16,20
Press: END OF LINE

The program will now be listed as:

```
1 X=1
10 GOTO 16
12 GOTO X OF 16,20
16 FORMAT 6F5.0
20 END
30 WRITE (15,14)
9999 GOTO 9998
```

When 4. is repeated again, ERROR 44 IN LINE 30 will be displayed. Change the reference in Line 30 from 14 to 16. Now when you attempt to renumber, ERROR 44 IN LINE 9999 will be displayed. Line 9999 purposely references a nonexistent line so that the program won't actually be renumbered. Delete Line 9999. All references to Line 16 have been found and changed.



Contour Map of Random Elevations Generated on the HP 9825A and 9862A.