FACTORS AFFECTING THE VALIDITY AND ACCURACY OF INSTRUMENTED IMPACT TESTS WITH SPECIAL REFERENCE TO THE PENDULUM AND DROP TOWER VERSIONS OF THE CHARPY TEST

Ward C. Stevens
B.S., Case Institute of Technology, 1969
M.S., Oregon Graduate Center, 1987

A dissertation submitted to the faculty
of the Oregon Graduate Center
in partial fulfillment of the
requirements for the degree
Doctor of Philosophy
in
Materials Science

October, 1987

The dissertation "Factors Affecting the Validity and Accuracy of Instrumented Impact Tests with Special Reference to the Pendulum and Drop Tower Versions of the Charpy Test" by Ward Conley Stevens has been examined and approved by the following Examination Committee:

William E. Wood
Professor and Department Chairman
Thesis Research Advisor

Robert G. Babb II
Associate Professor

David G. Atteridge
Associate Professor

John L. Johnsen
Adjunct Assistant Professor

ACKNOWLEDGMENT

I thank Professor W. E. Wood for serving as my thesis advisor. I also thank Professors D. G. Atteridge, R. G. Babb, and J. L. Johnsen for agreeing to examine this thesis. I appreciate G. H. Boehme and G. A. Frost's work in constructing and repairing the electronic equipment used in this study. I am grateful to Ireland Associates for providing the drop tower Charpy tup used in the experimental part of this work and to R. B. Turpin both for instruction in machining and for assistance in machining crucial parts of the apparatus. I thank D. H. Roberts for assistance in preparing the specimens. I also thank Professors R. A. Elliot, P. R. Davis, and C. H. Palmer Jr., as well as D. R. Ireland, and D. E. Christensen for many helpful discussions. I thank my wife, Helen, for typing and proofreading this document.

QUOTATION

Man's hand assaults the flinty rock; his eyes see all its treasures. He searches the sources of the rivers and brings hidden things to light. But where can wisdom be found? Where does understanding dwell? Man does not comprehend the worth; it cannot be found in the land of the living. (Holy Bible, New International Version, Job 28:9-13.)

God understands the way to it and he alone knows where it dwells, for he views the ends of the earth and sees everything under the heavens. (Holy Bible, New International Version, Job 28:23-24.)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

FACTORS AFFECTING THE VALIDITY AND ACCURACY OF
INSTRUMENTED IMPACT TESTS WITH SPECIAL REFERENCE TO
THE PENDULUM AND DROP TOWER VERSIONS OF THE CHARPY TEST

Ward C. Stevens, Ph.D.
Oregon Graduate Center, 1987

Supervising Professor: William E. Wood

While it is theoretically true that useful information can be calculated from recorded instrumented impact test data, such calculations can be performed only if the correct calibration procedures and numerical methods are used. However, neither the literature nor existing standards contain useful guidance with respect to correct calibration procedures or adequate numerical methods.

Two different homogenous populations of alloy 4340 steel Charpy bars were manufactured and tested by instrumented pendulum and drop tower versions of the Charpy impact tests. The salient dimensions and masses of the drop tower and pendulum machines were directly measured and the initial velocities were calculated from initial drop heights. Data were recorded using the COMPUTERSCOPE system manufactured by RC Electronics. The two instrumented tups were calibrated by applying

known loads statically, by matching energy results to known ASTM E-23 (Charpy impact test) results and by matching calculated general yield loads to assumed values of general yield loads. Using the calibration data, absorbed energies, general yield loads, and total system compliance were calculated from the recorded tup output information for both populations and test machines using ASTIR, a computer program specifically designed and constructed for this work. The energy, load, and compliance data were compared statistically to one another and, in the case of absorbed energies, to standard ASTM E-23 values.

It was found that the response of instrumented tups varies from almost totally strain rate insensitive to highly strain rate sensitive and that dynamic tup calibrations using energy standards can be dangerously misleading. It was shown that quite simple numerical methods are adequate for load and energy calculation and that approximately 80 data points are adequate for correct energy calculations.

Methods for investigating the discrepancies uncovered in this study and for obtaining first principles dynamic tup calibrations are outlined.

1. INTRODUCTION

1.1 Significance

1.1.1 Use

There are two factors which should motivate materials scientists to use instrumented impact testing.

First, instrumented impact testing permits accurate and precise measurement of absorbed energies greater than those which can be measured with current commercially available pendulum machine designs. For example, very tough materials exist which exceed the capacity of the largest Charpy pendulum impact machines. It is not adequate simply to increase pendulum weight since then dial precision would be coarsened unacceptably. A Charpy impact machine design combining very high pendulum weight and an instrumented tup would permit measurement of high absorbed energies with good precision and would have the desirable feature of nearly constant strain rate.

Second, instrumented impact testing permits direct measurement of dynamic material constants since loads and not energies are measured. Dynamic material constants are important since few engineering structures experience static conditions; hence, most engineering structures are ultimately limited by worst case situations involving medium or high strain rates. Fracture toughness values can be used as design data and not merely relative ranking numbers.

It would be convenient if all impact tests could be performed on machines permitting simple tup and anvil changes so that varying specimen sizes and configurations could be used. Drop towers are more likely to make this possible than pendulum machines.

1.1.2 Investigation

These two goals are attainable if three things are accomplished:

1. The factors controlling load, displacement, and energy calculations must be accurately measured.
2. The correct numerical methods must be used.
3. The dynamics of impact testing machines must be understood.

Load, displacement, and energy calculations depend on accurate knowledge of effective tup mass, initial velocity, and tup calibration under the conditions prevailing during testing. There is no standard method for measuring tup effective mass. The standard methods given for initial velocity measurement in, for example, the American Society for Testing and Materials (ASTM) method E-23 (the Charpy pendulum test) could possibly be questioned if applied to machine designs other than those approved under ASTM E-23. There is no standard method for 3 point bend tup calibration.

If digital storage techniques are used, results calculated from the digital data will depend on which differential equation solver or numerical integration approach is used. There is no standard method to perform the digital calculations. There is no evidence demonstrating that any particular approach is correct.

compensate for windage and friction losses. Hence, by conservation of energy and the fact that friction and windage as measured by specification are negligible, the kinetic energy just prior to the impact point must be very nearly equal to the initial potential energy of the pendulum. In addition, the kinetic energy at the fiducial point must, by the same logic, very nearly equal the potential energy at the end of a swing after an impact. The difference between these two kinetic energies is thus equal to the 'dial energy.'

Now suppose that the velocities at the impact point and the fiducial point are measured by electro-optical means. Using the velocities, both the initial and final energies can be calculated. With this data, it is possible to calculate an amount of energy equal to the 'dial energy.' Let this energy be called 'flag energy.'

The cause of the energy change measured by the 'dial' or 'flag' method is the interaction of the tup and the specimen. In particular, it is the load applied by the specimen to the tup (striker). By noting both the initial velocity of the tup and the load applied to the tup by the specimen, the change in kinetic energy of the pendulum can later be calculated. For example, the load-time record could be integrated to obtain the change in velocity. If that change were subtracted from the initial velocity, the result should equal the velocity of the tup at the fiducial point from which, with the initial velocity, it has already been shown that the energy change can be calculated. Let the energy change calculated from the load-time record be called the 'tup energy.' The law of conservation of energy again demands that the 'tup energy' equal the 'flag' and 'dial energies.' Therefore, 'dial,' 'flag,' and 'tup energies' are all equally valid means of measuring energy absorbed by the specimen.

1.3.3 Drop Tower Charpy Test

Next, consider a hypothetical [15];[16] drop tower Charpy test. (See Figure 1-1). A rigid, heavy crosshead with a tup attached to the center of its undersurface is used in the test. The tup's dimensions are consistent with the requirements of ASTM E-23. The crosshead is allowed to drop from a predetermined height, constrained only by vertical low-friction guide bars. Just before striking the specimen, the crosshead has developed a certain amount of energy and a vertical velocity which falls within the range allowed for the tangential velocity of the tup in the Pendulum Charpy test (E-23). The specimen is surface-ground and notched with dimensions in accordance with ASTM E-23. It is supported horizontally as a simple beam on an anvil which is dimensioned in accordance with ASTM E-23. The notch on the specimen faces away from the tup. The notch is centered between the

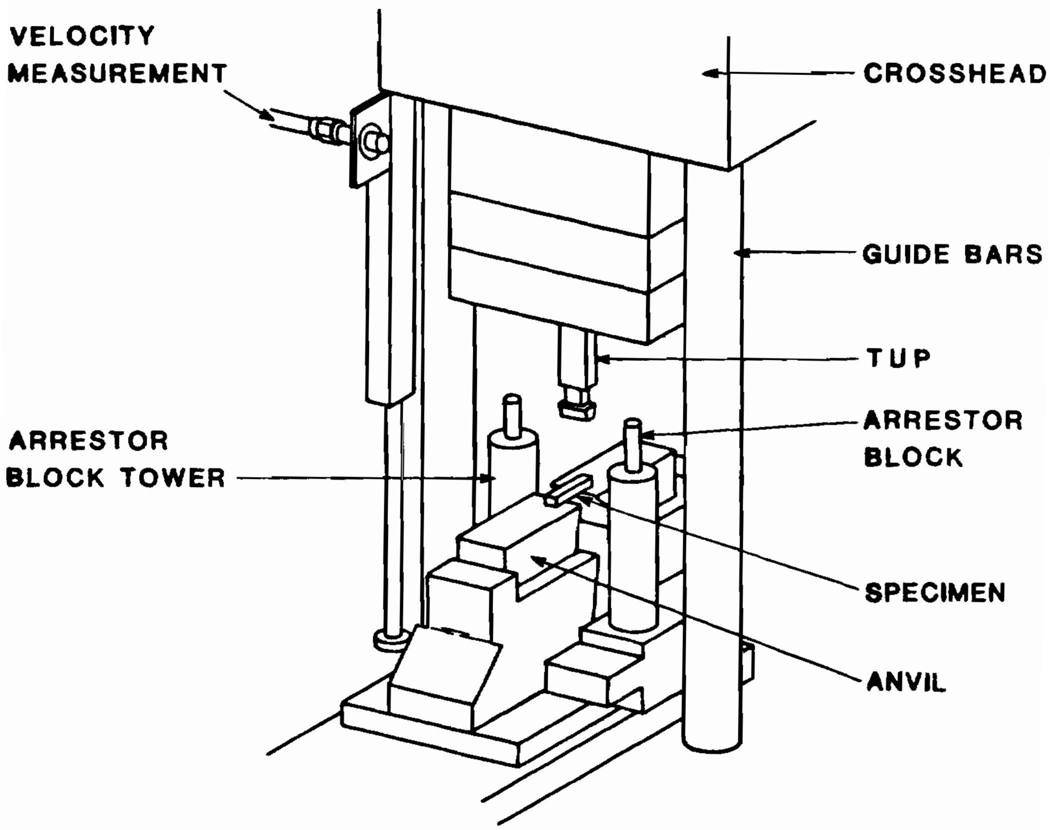


Figure 1-1. Drop Tower Charpy Test

sides of the anvil in the horizontal plane and is parallel to the plane of the crosshead. When the tup strikes the specimen, it breaks the specimen and in so doing the tup loses a portion of its energy. The crosshead then falls another inch to the fiducial point. There, its velocity is measured. After another inch of free fall, the crosshead encounters an arresting device and is stopped.

Since the available kinetic energy is never reconverted into potential energy (as it is in the pendulum Charpy machine), it is not immediately obvious how to measure or calculate the final energy of the crosshead accurately. Apparently the problem was considered insoluble in the early 20th century when Izod devised his test, since that was why he rejected the drop tower version of the Izod test. [17]. Bluhm also did not try to measure energies for individual tests; rather, he varied drop height and reported the break - no break energy for a population. [18]. However at present, the velocity of the crosshead can be measured electro-optically, and the load applied to the tup can be measured electronically.

The energy change of the tup can be calculated as follows: Allow the crosshead to drop freely through the fiducial plane with no specimen in place. Record the velocity and calculate this free drop energy. Repeat the test with a specimen in place, recording velocity and calculating energy. The difference in the two energies should be equal to the 'flag energy' from an ASTM E-23 pendulum machine with an identical specimen.

There is a second method whereby the 'flag energy' for the drop tower test can be calculated: Allow the crosshead to fall freely except for a specimen properly placed on the anvil. Measure the velocity just before impact and at the fiducial plane. Despite the loss of energy caused by the impact, the crosshead must have gained energy in the fall from the plane where the impact began to the fiducial plane. Calculate the energy thus gained by multiplying the distance from the initiation plane to the fiducial plane by the weight of the crosshead. Subtract that amount from the kinetic energy of the crosshead at the fiducial plane calculated from the final velocity. Finally, subtract the remainder from the kinetic energy calculated from the initial velocity. Again, the 'flag energy' calculated in this way should equal both the pendulum 'flag energy' and the ASTM E-23 standard (or 'dial') energy for an identical specimen at an identical temperature.

The crosshead energy loss due to the specimen can also be calculated from a load-time record as it can in the case of the pendulum. Since the effect of gravity can be accounted for in the case of tup energy calculated from a load-time record of a drop tower impact test, the drop tower 'tup energy' must also be equivalent to both the pendulum 'tup energy' and the ASTM E-23 'dial energy' for an identical specimen at an identical temperature.

At this point it is important to consider the motions of the pendulum and the crosshead and the 'tup energy' calculations in greater detail.

1.3.4 Dynamics of Both Methodologies

Consider the motion of the tup in the Charpy pendulum machine. It is nearly the end of a rigid pendulum which swings (*i.e. rotates*) about a low friction bearing whose axis is horizontal. The pendulum must obey Newton's laws of motion. Hence:

$$\tau = I \frac{d\omega}{dt} \tag{1-1}$$

where:	τ	$\equiv$	torque applied to the pendulum	(1-2)
	I	$\equiv$	*rotational inertia of the pendulum*	(1-3)
	ω	$\equiv$	rotational velocity of the pendulum	(1-4)
	t	$\equiv$	time	(1-5)

Let:	V_T	$\equiv$	the tangential velocity of the tup	(1-6)
	A_T	$\equiv$	the tangential acceleration of the tup	(1-7)
	F_T	$\equiv$	the tangential force applied to the tup	(1-8)
	r_s	$\equiv$	the radius from the center of the bearing to the tup	(1-9)

It follows that: $\tau = F * r_s$ (by the definition of torque) (1-10)

$\omega = V_T/r_s$ (by the definition of angular velocity) (1-11)

Substituting:

$F_T * r_s = I(dV_t/dt)(1/r_s)$ (1-12)

(since r_s is constant)

Hence: $F_T = (I/r_s^2)(dv_t/dt)$ (1-13)

But: I/r_s^2 has units of mass. (1-14)

So define: $M_{eff} \equiv I/r_s^2$ (1-15)

(M_{eff} is called 'effective mass.')

Therefore: $F_T = M_{eff}\, dV_T/dt$ (1-16)

But: $dV_T/dt = A_T$ (1-17)

Therefore: $F_T = M_{eff}\, A_T$ (1-18)

Let: $P(t) \equiv$ the load applied to the tup by the specimen as a function of time (1-19)

Now, the impact event in a Charpy machine begins (according to the ASTM-23 requirements) with the pendulum very nearly vertical (i.e., with the motion of the tup nearly horizontal). It usually ends when the horizontal component of the displacement of the tup is no more than a few tenths of an inch and it could not possibly continue horizontally for more than an inch. Hence the entire event occurs while the pendulum moves through a small fraction of a radian from the vertical, and the tangential force applied to the tup and the motion of the tup are very nearly horizontal; force and motion and, more importantly, the gravitational component of F_t are nil. Now frictional and windage contributions are (due to E-23) also negligible. Hence:

$$F_T = P(t) \tag{1-20}$$

Substituting: $$P(t) = M_{eff} A_T \tag{1-21}$$

$$A_T = P(t)/M_{eff} \tag{1-22}$$

Let: $X \equiv$ horizontal Displacement (1-23)

Since it has been seen that horizontal and tangential components of motion are very nearly equal:

$$A_T = d^2x/dt^2 \tag{1-24}$$

Substituting: $d^2x/dt^2 = P(t)/M_{eff}$ (1-25)

Let: $t_o \equiv$ the time when impact starts (1-26)

$t_f \equiv$ the time when impact ends (1-27)

$x_o \equiv$ the horizontal position when impact starts (1-28)

Set: $x_o \equiv 0$ (1-29)

and: $t_o \equiv 0$ (1-30)

$v(t) \equiv$ horizontal velocity of the tup at time "t" (1-31)

$v_o \equiv$ the initial horizontal velocity when impact begins. (1-32)

Since P(t) is in the negative direction, integration of Equation 1-25 yields:

$$\frac{dx}{dt} = v(t) = v_o - \int_{t_o}^{t} \frac{P(t)}{M_{eff}} \, dt \qquad (1\text{-}33)$$

Hence:

$$dx = [v(t)] \, dt = [v_o - \int_{t_o}^{t} \frac{P(t)}{M_{eff}} \, dt] \, dt \qquad (1\text{-}34)$$

Integrating Equation 1-34 from t_o to t:

$$x(t) = x_o + v_o t - v_o t_o - \int_{t_o}^{t} \int_{t_o}^{t} \frac{P(t)}{M_{eff}} \, dt \, dt \qquad (1\text{-}35)$$

But since Equation 1-29 provides that:

$$x_o \equiv 0$$

And Equation 1-30 provides that:

$$t_o \equiv 0$$

Then it follows that:

$$x(t) = v_o t - \int_{t_o}^{t} \int_{t_o}^{t} \frac{P(t)}{M_{eff}} \, dt \, dt \tag{1-36}$$

Let: $E(t) \equiv$ the energy dissipated by the tup due to $P(t)$ (1-37)

$$dE = P(t) \, dx \tag{1-38}$$

Substituting:

$$dE = P(t) \left[v_o - \int_{t_o}^{t} \frac{P(t)}{M_{eff}} \, d(t)\right] dt \tag{1-39}$$

Integrating:

$$E(t) = v_o \int_{t_o}^{t} P(t) \, dt - \int_{t_o}^{t} P(t) \int_{t_o}^{t} \frac{P(t)}{M_{eff}} \, dt \, dt \tag{1-40}$$

It follows from the above that an accurate value of M_{eff} is essential in correctly calculating energy, displacement, and velocity as a function of time.

Having obtained equations for A(t), V(t), X(t), and E(t) for the pendulum, let the equations for the same quantities for the drop tower be calculated.

The tup in the drop tower is rigidly attached to the crosshead which is constrained by the guide bars so as to slide only in a vertical direction. The motion of the drop tower tup due to Newton's second law is then described by:

$$F_V(t) = M_X A_V(t) \tag{1-41}$$

where:

$$F_V \equiv \text{total vertical force on the crosshead} \tag{1-42}$$

$$M_X \equiv \text{mass of the crosshead} \tag{1-43}$$

$$A_v(t) \equiv \frac{d^2Z}{dt^2} \tag{1-44}$$

Let:

$$g \equiv \text{gravitational acceleration} \tag{1-45}$$

$$p(t) \equiv \text{load applied by specimen to tup} \tag{1-46}$$

$$V_V(t) \equiv \text{vertical velocity of tup} \tag{1-47}$$

$$z(t) \equiv \text{vertical displacement of tup} \tag{1-48}$$

$$E(t) \equiv \text{energy dissipated from tup} \tag{1-49}$$

$$V_{VO} \equiv \text{velocity of impact event start} \tag{1-50}$$

$$t_O \equiv \text{time of impact event start} \tag{1-51}$$

$$z_O \equiv 0 \equiv \text{location of impact event start} \tag{1-52}$$

Since the windage and frictional drag are assumed negligible:

$$F_V = P(t) - M_X g \tag{1-53}$$

Substituting Equations 1-44 and 1-53 in Equation 1-41:

$$-\ (\ P(t) - M_X g\ ) = M_X \frac{d^2 z}{d\,t^2} \tag{1-54}$$

(Since $P(t)=F_V$ is in the negative direction)

Solving for acceleration:

$$\frac{d^2 z}{d\,t^2} = -\ (\ \frac{P(t)}{M_X} - g\ ) \tag{1-55}$$

Integrating:

$$\frac{dz}{dt} = V_v(t) = V_{vo} - \int_{t_o}^{t} [\ \frac{P(t)}{M_X} - g\ ]\ dt \tag{1-56}$$

where $V_{VO} \equiv V_v(t_o)$

$$\frac{dz}{dt} = V_{vo} - \int_{t_o}^{t} \frac{P(t)}{M_x}\, dt + gt - gt_o \qquad (1\text{-}57)$$

But t_o has previously been defined as 0 in Equation 1-30.

So:

$$\frac{dz}{dt} = V_{vo} - \int_{t_o}^{t} \frac{P(t)}{M_x}\, dt + gt \qquad (1\text{-}58)$$

Rearranging:

$$dz = V_{vo}\, dt - \int_{t_o}^{t} \frac{P(t)}{M_x}\, dt\, dt + gt\, dt \qquad (1\text{-}59)$$

Remember that $t_o \equiv 0$ and integrate:

$$z(t) = V_{vo}\, t - \int_{t_o}^{t} \int_{t_o}^{t} \frac{P(t)}{M_x}\, dt\, dt + \frac{1}{2}\, gt^2 \qquad (1\text{-}60)$$

The energy dissipation at the tup is:

$$dE = [P(t) - M_x g] * dz \tag{1-61}$$

But the energy dissipation by the tup <u>due to the specimen</u>, dE_S, is:

$$dE_S = P(t)\ dz \tag{1-62}$$

$$dE_S = P(t)\ [V_{vo}\ dt - \int_{t_o}^{t} \frac{P(t)}{M_x}\ dt\ dt + gt\ dt] \tag{1-63}$$

$$E_S(t) = V_{vo} \int P(t)\ dt \tag{1-64}$$

$$- \int_{t_o}^{t} P(t) \int_{t_o}^{t} \frac{P(t)}{M_x}\ dt\ dt + g \int_{t_o}^{t} P(t)\ t\ dt$$

The equations of motion and of energy dissipation for the two forms of the Charpy test are of similar form except for the term containing the gravitational acceleration.

Let: $t_f \equiv$ the time at event termination (1-65)

Then, E (t_f) for the drop tower will equal E(t_f) for the pendulum in spite of the added term.

There are two arguments to support such a claim. First, the gravitational term $g\int_{t_o}^{t} P(t)tdt$ will be negligibly small both due to the short duration of the event and due to the small size of g as compared to P(t)/M. Second, E_a, i.e. the term $V_{vo}\int P(t)dt$, will be slightly smaller in the drop tower test than in the pendulum test since the event will take slightly less time in the drop tower test than in the pendulum test due to gravitational acceleration.

In spite of the argument in the last paragraph, it might be contended that the strain rate in the drop tower test will be higher because of gravitational acceleration than the strain rate in the pendulum test, and the greater strain rate will distort P(t). As a result, P(t) would be significantly different in the drop tower and pendulum tests. Such a conclusion is not valid because:

1. The time for acceleration is too short for gravitation to significantly affect strain rate;
2. The magnitude of gravitational force is not significant compared to the impact load; and

3. For larger scale tests (the DT and DWTT) which have longer event times during which gravitation should be able to exert a greater effect, it has already been shown that the drop-weight and pendulum versions are equivalent. [19];[20].

Energy is dissipated in the following ways: It is used to accelerate the specimen to tup velocity (the initial inertial event [21]), to plastically deform the specimen (crack initiation), and to fracture the specimen (crack propagation). It is stored elastically in the apparatus and specimen (i.e., as "ringing"). [22]. Finally, it is dissipated by the apparatus' bearings. The first of these quantities must be the same in both the test types since it consists of the amount of energy required to accelerate a standard sized specimen to a standardized velocity. The second and third must be the same as they are related to material constants of the same material made into standard sized specimens. It might be argued that the fourth quantity might be different for pendulum and drop tower tests since it depends on the stiffness of the machines. The crosshead of the drop tower is clearly stiffer than the pendulum in the pendulum machine since the crosshead has greater moment of inertia and is shorter than the pendulum. Hence, if the tup and anvil used in the drop tower are of equal or greater stiffness than those used in the pendulum test, there must be less stored elastic energy during the drop tower test than during the pendulum test. However, as machines are made stiffer, stored elastic energy must decrease until it is inconsequential. Only at that point can many machine designs get the same answer on impact tests of the same materials as noted by Bluhm [23], but such is the case with modern pendulum Charpy machines. Therefore, increasing the stiffness by using a drop tower instead of a pendulum should not distort the Charpy test.

The fifth quantity has been carefully made negligible in the pendulum apparatus by setting the radius of the strike equal to the radius of percussion and by requiring extremely low friction bearings. If the drop tower has linear bearings with low friction as well as the ability to orient the crosshead accurately, it will be negligible for the drop tower too.

Several details remain to be treated:

1. Measurement of load applied by the specimen to the tup;
2. Determination of tup velocity;
3. Numerical methods used to trace tup motion and calculate absorbed energy.

1.4 Instrumentation Analysis

The procedures used in this study to measure tup load are the same as those described by earlier researchers. [24];[25];[26];[27];[28]. In essence, the solution involves machining a reduced section into the tup and attaching to it strain gauges connected together in the form of a rosette. Various rosette designs have been used successfully. The rosette, in turn, is included in a resistance bridge circuit to which an input voltage is applied as shown in Figure 1-2. [29];[30];[31];[32]. If a load is applied to the tup, the reduced section undergoes an elastic strain proportional to the load. The bridge becomes unbalanced, and the output voltage of the bridge increases. With an excellent tup design, the increase in voltage is very nearly linearly proportional to the applied load and is independent of strain rate. In this paper the power supply, the portion of the bridge other than the strain gauges, and an amplifier to boost the tup output to usable levels is called the amplifier unit. The instrumented tup output can then be fed to an oscilloscope and the oscilloscope trace photographed. Alternatively, the signal can be digitized and recorded at discrete time intervals by any one of a number of commercial systems. The recorded digitized signal can become the argument of the load voltage calibration curve, thus yielding load information.

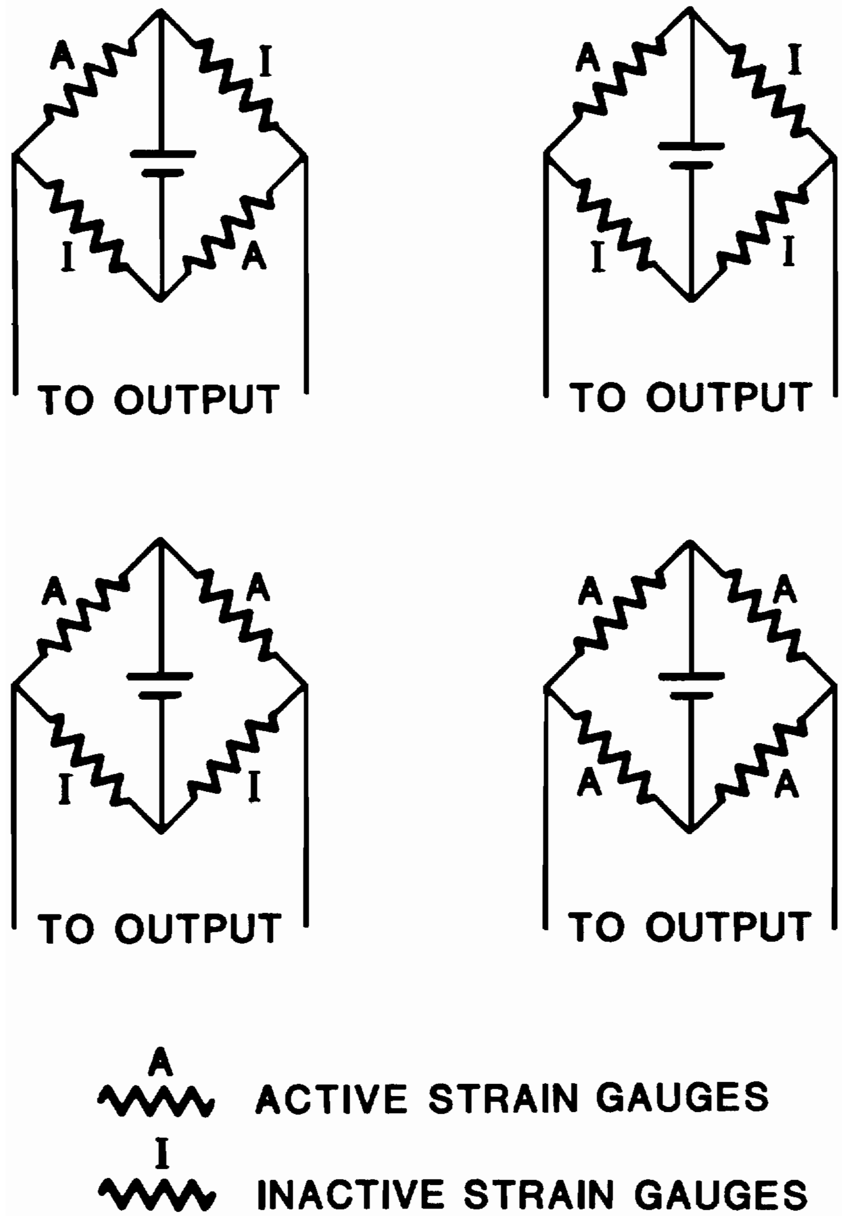


Figure 1-2. Load Cell Resistance Bridge Designs
Reported in the Literature

1.4.1 Calibration

1.4.1.1 Tup Calibration

1.4.1.1.1 Static Tup Calibration

The response of the amplifier unit and tup can be calibrated as follows:

1. Apply known loads to the tup and record the corresponding outputs of the tup-amplifier unit combination.
2. Use a least squares approach to fit a polynomial to the data.
3. Back solve the resulting function and use the result to calculate load from recorded signals.

1.4.1.1.2 Dynamic Tup Calibration

Since the calibration method described in the preceding paragraph is performed under static conditions, Ireland [33] has argued in favor of a dynamic calibration methodology as follows:

1. Insert a very strong elastic object on the anvil;
2. Drop the pendulum or crosshead from a known height on the object while recording the output of the tup;
3. Find a linear tup calibration function which will cause the impulse calculated from the tup output record to equal the value required to reverse the motion of the pendulum or crosshead.

The virtues and pitfalls of the static and dynamic calibration methods will be considered in the Discussion section.

1.4.1.2 Initial Velocity

There are several methods by which initial velocity can be found:

1. Calculation from drop height;
2. Electro-optical methods:
    a. Timing the passage of flags;
    b. Ronchi gratings or shaft encoders.
3. Stroboscopy

1.4.1.2.1 Drop Height

ASTM E-23 suggests that the initial velocity (and hence the free swing and free drop final velocities) could be calculated by $V_0 = (2gh)^{\frac{1}{2}}$ which is satisfactory if, as E-23 requires, the windage and frictional losses are small.

1.4.1.2.2 Electro-Optical Methods

1.4.1.2.2.1 'Flags'

ASTM E-604 suggests that both initial and final velocities can be calculated by electro-optical measurement of the passage of a rectangle of accurately known width, called a 'flag.' There are two basic versions of this technique. In the first, a light source shines on an electronic detector of light and an opaque rectangle attached to the pendulum or crosshead passes between them. Either the output of the detector as a function of time is recorded or the time when the detector output drops and the time when it returns to a high level are recorded. In any event, average velocity is just the width of the flag divided by the time during which the flag interrupted the light. In the second version, both the light and the detector aim toward the pendulum or crosshead. The flag is highly reflective and it is surrounded by highly absorptive (i.e. black) material. When the flag arrives, it causes the detector output to jump; when it departs, it causes the detector output to drop. The rest of the method is exactly the same as that used in the first version of this method.

1.4.1.2.2.2 Ronchi Gratings and Shaft Encoders

There is another method for obtaining pendulum or crosshead velocity which involves the use of Ronchi gratings. A Ronchi grating is an equal-width, equally-spaced parallel series of opaque bars on a plane transparent or reflective substance such that its opaque width is equal to its transparent reflective width. Consider two identical transparent background Ronchi gratings -- one attached to the crosshead or pendulum and one attached to a fixed support with the rulings parallel to one another and perpendicular to the direction of tup motion. (See Figure 1-3. [34].) A well collimated beam of light passes perpendicularly through the stationary Ronchi grating to an electronic light detector. The light, its associated lenses, and the detector are far enough away that the moving Ronchi grating can pass easily between them, but they are close enough to one another that the amplitude of the light gives a strong

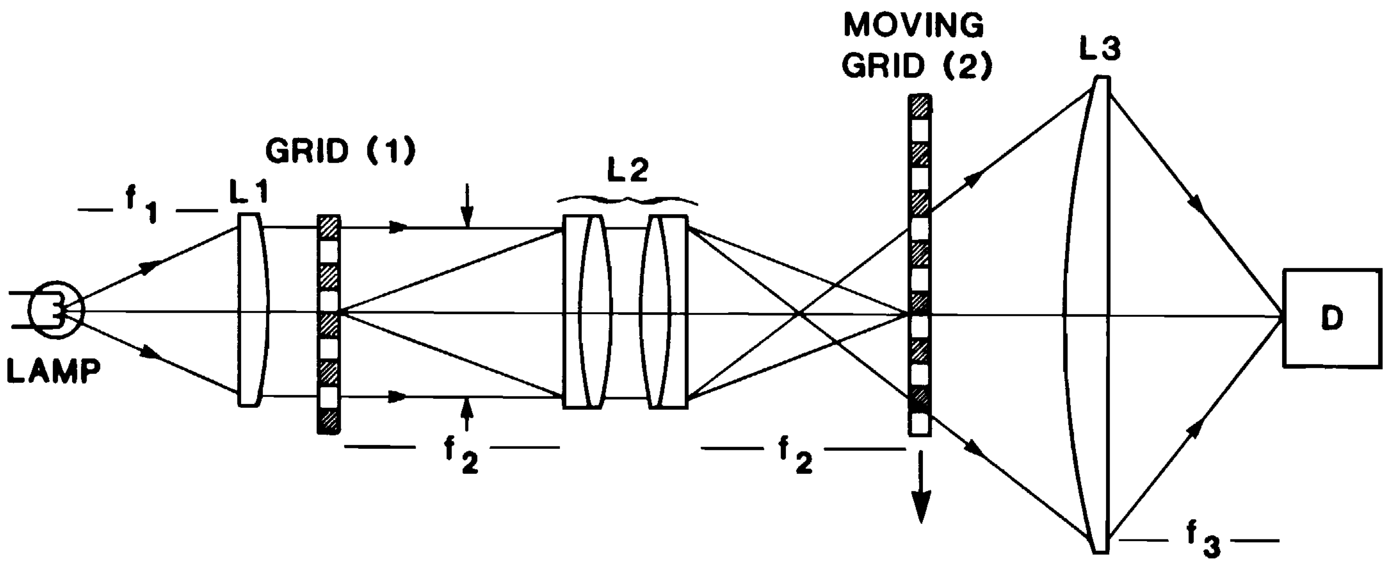


Figure 1-3. A Ronchi Grating System for Velocity Measurement

signal at the detector and the light is still well collimated as it passes through the grating(s). Lens L1 of focal length L1 collimates the light. Lens L2, actually a pair of achromatic doublets each of focal length L2, focuses the image of the stationary grid, grid (1), on the plane of the moving grid, grid (2), and lens L3 which has focal length f_3 focuses the light onto the detector, D. As the moving grating passes through the light beam it will vary from being perfectly aligned to the stationary grating, allowing maximum light to pass through, to perfectly misaligned, allowing essentially no light to pass through, and back again to perfect alignment. Hence the output of the light detector is a triangle wave of variable frequency. The reciprocal of the frequency of the signal is twice the grating spacing divided by the velocity. Hence, the velocity is twice the grating spacing divided by the output period. A similar method could employ a Ronchi grating with reflective bars between the opaque ones.

An exactly analogous method which can be used on a pendulum involves the use of a device known as a shaft encoder. The shaft encoder contains two transparent discs with a pattern of opaque arcs printed on them. The arcs are concentric to the discs. The output of the shaft encoder is a parallel digital signal representing the angular position of the shaft. Shaft encoders exist which are accurate to a small fraction of a degree. If a shaft encoder were attached to the shaft of a pendulum machine, its output could be used to calculate the velocity of the pendulum. For example, the signal corresponding to the least significant binary digit would be a square wave with frequency proportional to shaft angular velocity and hence to tup tangential velocity.

An advantage of both the Ronchi grating and flag velocity measurements is that they measure actual and not theoretical velocities. That is, they do not depend on the assumption that drag sources are negligible.

An advantage of the Ronchi grating method over the flag method is that several valid velocity measurements can be made before the event and many can be made during and after it. Since relatively few signal samples are usually recorded, and since different materials require differing sampling rates, the flag system often requires different flag size and spacing for different materials if both initial and final velocities are desired. The Ronchi grating system has inherently enough flexibility to avoid such a problem.

1.4.1.2.3 Stroboscopy

If a moving object, illuminated by a light flashing at known time intervals, is photographed, the position of the object at each time interval can be obtained from the photographic image, and the speed of the object can be calculated from the time-position data thus obtained.

Stroboscopy is easier to set up, but is not as accurate as Ronchi grating techniques.

1.4.1.3 Effective Mass

1.4.1.3.1 Effective Mass of the Pendulum

Let:

I	$\equiv$	the angular momentum	(1-66)
M_{act}	$\equiv$	pendulum assembly actual mass	(1-67)
M_{eff}	$\equiv$	pendulum assembly effective mass	(1-68)
r_{cg}	$\equiv$	center of gravity radius	(1-69)
r_g	$\equiv$	radius of gyration	(1-70)
r_o	$\equiv$	radius of oscillation	(1-71)
r_p	$\equiv$	radius of percussion	(1-72)
r_s	$\equiv$	radius of strike	(1-73)

The angular inertia of a compound pendulum is the pendulum's total rotating mass multiplied by its radius of gyration squared. [41].

I.e.:

$$I = r_g^2 \, M_{act} \tag{1-74}$$

Due to Equation 1-15:

$$M_{eff} = I / r_s^2 \tag{1-75}$$

Substituting Equation 1-79 into Equation 1-75:

$$M_{eff} = r_p \, r_{c\,g} \, M_{act} / r_s^2 \tag{1-80}$$

1.4.1.3.2 Effective Mass of the Crosshead

The effective mass of the crosshead is simply the total mass of the crosshead and all attached accessories.

1.4.2 Calculations

The 'tup energy' can be calculated from the initial velocity and the load-time curve in several ways. The most obvious are:

1. The Augland equation;
2. The double integral method.

1.4.2.1 The Augland Equation

The Augland equation (actually devised by Grumbach) [35];[36];[37] is:

$$E_{tot} = E_a - \frac{Ea^2}{4E_0} \qquad (1\text{-}81)$$

where:

E_{tot} ≡ the total energy absorbed by the specimen during the event (1-82)

$$E_a \equiv v_o \int_{t_o}^{t_f} P \, dt \quad (1\text{-}83)$$

$$E_o \equiv \tfrac{1}{2} MV_o^2 \quad (1\text{-}84)$$

where:

V_{to} ≡ initial velocity (1-85)

t_o ≡ time at event start (1-86)

t_t ≡ time at event end (1-87)

$P = P(t)$ ≡ load applied to the tup as a function of time (1-88)

M ≡ mass of crosshead or effective mass of pendulum (1-89)

The Augland equation can be shown to be exact for the pendulum, but it requires a small correction for the drop tower. (See derivations in Appendix A.)

1.4.2.2 The Double Integral Method

The double integral method, described in section 1.3.4, yields:

For the Pendulum:

$$E(t_f) = v_o \int_{t_o}^{t_f} P(t)\, dt - \int_{t_o}^{t_f} P(t) \int_{t_o}^{t_f} \frac{P(t)}{M_{eff}}\, dt\, dt \qquad (1\text{-}90)$$

For the Drop Tower:

$$E(t_f) = V_o \int_{t_o}^{t_f} P(t)\, dt \qquad (1\text{-}91)$$

$$- \int_{t_o}^{t_f} P(t) \int_{t_o}^{t_f} \frac{P(t)}{Mx}\, dt\, dt + g \int_{t_o}^{t_f} P(t)\, t\, dt$$

1.4.2.3 Relationship Between the 'Tup Energy' Methods

For the pendulum, the double integral method can be shown to be equivalent to the original Augland equation. For the drop tower, the double integral method can be shown to be equivalent to the corrected Augland equation. It is more straightforward to obtain intermediate values of velocity and displacement using the double integral method.

1.4.2.4 Practical Application of the Methods

1.4.2.4.1 Practical Application of the Augland Equation

In the event that the load-time record is a photograph, the most straightforward option is to carefully obtain the scale of the photograph, graphically integrate the load-time curve and apply the Augland equation.

1.4.2.4.2 Practical Application of the Double Integral Method

If the load-time record is a digitized sampling of load at discrete time intervals, it will be necessary to perform the integration by a numerical method. Since the load samplings are at equal intervals, only a Newton-Cotes method or spline integration would be possible. The Zeroth order Newton-Cotes method (the rectangle rule) is a desirable choice: it permits integration over an arbitrary number of points and it is simple to program. Higher order Newton-Cotes methods such as the trapezoid rule and Simpson's rule could also be used.

1.4.2.5 Integration Methods

1.4.2.5.1 The Rectangle Rule

Acceleration during the i^{th} interval is given by:

$$a_i = P_i/M \tag{1-92}$$

where: $P_i \equiv$ the load sample at the beginning of the i^{th} interval (1-93)

and: $A_i \equiv$ the acceleration during that interval (1-94)

$M \equiv$ the mass of the crosshead or effective mass of the pendulum (1-95)

If velocity at the end of the i^{th} interval is given by:

$$V_{i+1} = V_i + A_i \, \Delta T \tag{1-96}$$

where: $V_{i+1} \equiv$ velocity at the end of the ith interval (1-97)

$V_i \equiv$ velocity at the end of the i-1th interval (1-98)

$\Delta T \equiv$ sampling interval (1-99)

Let: $V_{avg} \equiv$ the average velocity during the i^{th} interval (1-100)

$$V_{avg} = (V_i + V_{i+1}) / 2 \tag{1-101}$$

Therefore, the distance travelled during the i^{th} interval is:

$$\Delta X = [(V_i + V_{i+1}) / 2] \ \Delta T \tag{1-102}$$

Hence, the distance travelled by the end of the i^{th} interval is:

$$X_{i+1} = X_i + [(V_i + V_{i+1}) / 2] \ \Delta T \tag{1-103}$$

where: $X_{i+1} \equiv$ distance travelled up to the end of the i^{th} interval (1-104)

and: $X_i \equiv$ distance travelled up to the end of the $i\text{-}1^{th}$ interval (1-105)

Now the energy absorbed by the tup during the i^{th} interval is:

$$\Delta E_i = P_i \Delta X_i = P_i[(V_i + V_{i+1})/2] \quad \Delta T \tag{1-106}$$

Call the above equation the zeroth order method. According to that method, energy absorbed up to the i^{th} interval is:

$$E_{i+1} = E_i + \Delta E = E_i + P_i \, [(V_i + V_{i+1})/2] \quad \Delta T \tag{1-107}$$

where: $E_i \equiv$ energy absorbed up to the end of the i^{th} interval (1-108)

and: $E_{i-1} \equiv$ energy absorbed up to the i-1th interval (1-109)

A simplified version of a program to use this approach for the pendulum would be (assuming the loads are stored in any P):

For i = 0 to last (whatever last might be!)

Begin

$A_i = P_i/M;$

$V_{i+1} = V_i * A_i * \Delta t;$

$V_{av} = (V_i + V_{i+1})/2;$

$\Delta X = V_{av} * \Delta t;$

$X_{i+1} = X_i + \Delta X;$

$\Delta E = P_i \; \Delta X;$

$E_{i+1} = E_i + \Delta E;$

End; (somehow the program initialized X_i, E_i, V_i, M, P_i and Δt correctly.)

Appendix A contains the ASTIR program which performs the necessary calculations using data collected by the COMPUTERSCOPE program.

1.4.2.5.2 Higher Order Methods

1.4.2.5.2.1 The Trapezoid Rule

Assume instead that during each interval load varies linearly with time and velocity quadratically with time.

We define: $P_i \equiv$ load at the beginning of the i^{th} interval (1-110)

$P_{i+1} \equiv$ load at the end of the i^{th} interval (1-111)

and: $V_{i+2} \equiv$ velocity at the end of the $i+1^{th}$ interval (1-112)

Also:

Allow ΔE_i, E_{i+1}, A_i, V_i and V_{i+1} to be defined as before.

It follows that:

$$E_{i+1} = E_i - \left(\frac{(\Delta t\ P_{i+1} + \Delta t\ P_i)\ V_{i+2}}{24} + \frac{(-10\Delta t\ P_{i+1} - 6\Delta t\ P_i)\ V_{i+1}}{24} - \frac{3\Delta t\ V_i\ P_{i+1}}{24} - \frac{7\Delta t\ P_i\ V_i}{24} \right) \quad (1\text{-}113)$$

Call this the first order method. This method is derived in section A.4.1.1 of Appendix A.

1.4.2.5.2.2 Simpson's Rule

Further, load could vary parabolically during the i^{th} interval and velocity cubically with time.

Let: $P_{i+2} \equiv$ load at the end of the i+1th interval (1-114)

E_i, A_i, M, P_i, P_{i+1}, V_i, V_{i+1}, V_{i+2} be defined as before

and: $V_{i+3} \equiv$ velocity at the end of the i+2th interval (1-115)

It follows that:

$$E_{i+1} = E_i - \left(\frac{(3\Delta t\ P_{i+2} - 20\Delta t\ P_{i+1} - 13\Delta t\ P_i)\ V_{i+3}}{720} \right. \qquad (1\text{-}116)$$

$$+ \frac{(-15\Delta t\ P_{i+2} + 102\Delta t\ P_{i+1} + 63\ \Delta t\ P_i)\ V_{i+2}}{720}$$

$$+ \frac{(51\Delta t\ V_{i+1} + 21\Delta t\ V_i)\ P_{i+2}}{720}$$

$$+ \frac{(-444\Delta t\ P_{i+1} - 177\Delta t\ P_i)\ V_{i+1}}{720}$$

$$\left. - \frac{118\Delta t\ V_i\ P_{i+1}}{720} - \frac{173\Delta t\ P_i\ V_i}{720} \right)$$

Let us call this the second order method. This method is derived in section A.4.2.1 of Appendix A. If still higher methods are considered, the equations will rapidly become more complex.

1.5 Summary

So far, it has been shown that there are three equivalent methods of measuring energy absorption in the pendulum Charpy test and two equivalent methods in the drop tower Charpy test. It has further been shown that all of the Charpy pendulum and Charpy drop tower energy measures should be theoretically equivalent. It would be convenient if there were a third method for the drop tower as well. A possible method for providing such an energy method is suggested by ASTM E-604 in which it is suggested that the arresting mechanism be aluminum blocks and the energy measure be the degree of plastic deformation sustained by the blocks. ASTM E-604 suggests a method which apparently provides an adequate calibration, allowing absorbed energy to be obtained from the block deformation. [38].

Theoretically, all these energy measurements are equal. In particular, the pendulum 'dial' and 'tup energies' should be equal and they should equal the drop tower 'tup energy' for equivalent systems.

It remains to test the hypothesis.

2. PROCEDURE

2.1 Specimens

Five plates of Alloy 4340 Aircraft quality steel which had been blanchard ground to a thickness of approximately 0.400" and which came from a single metal lot were obtained from a commercial source. A sample was cut from each plate and submitted to spark spectrography. The plates were band sawed to 2.160 ± 0.005" lengths and each of the resulting pieces was band saw cut into 0.400" ± 0.005" widths. The resulting Charpy specimen blanks measured 0.400" ± 0.005" x 0.400" ± 0.005" x 2.160" ± 0.005" with the long direction of each blank parallel to the rolling direction. The width of each blank was then abrasively machined to 0.394" ± 0.001" and the thickness was abrasively machined to 0.394" ± 0.003". One hundred and nine suitable specimens were thus prepared. Each specimen was then notched by broaching so that the notches were perpendicular to the rolling direction of the plate. The notches were imaged at low magnification on a Carl Zeiss Standard Universal M microscope using transillumination. (In other words, the metallograph was used as a shadow dimensional comparator.) The dimensions of the notches were checked using a translucent template and found to be in dimensional tolerance to ASTM E-23.

The 109 Charpy bars were then austenitized at 1550°F for one hour, quenched in oil, and tempered at 400°F for one half hour. A random sample had final hardness of 54 on the Rockwell "C" hardness scale and dimensions still in accordance with ASTM E-23.

Forty-five of the Charpy bars were loaded into an evacuated quartz tube and tempered at 1180°F for 12 hours (8 hours were required to reach 1200°F.) A random sample examined after this treatment still had dimensions in accordance with ASTM E-23, but their hardness was now 28 on the Rockwell "C" hardness scale.

2.2 Calibration

2.2.1 Tup Calibration

2.2.1.1 Pendulum Tup Calibration

The Effects Technology Inc. tup used in the pendulum experiments was calibrated as follows: An Ireland Associates 20,000 lb. drop tower tup was attached to a warmed-up amplifier unit whose output was attached to an accurate and sensitive voltmeter.

Compressive load was applied to it using a Tinius-Olsen tensile testing machine. The amplifier unit's output was recorded as a function of load during several runs. The results were examined statistically and a least squares best fit was made using a Fortran statistical subroutine called PANOVA which is described in section B.3.3.8 of Appendix B.

A special tip was installed on the Ireland Associates 20,000 lb. tup and it was mounted on the pendulum machine using a specially constructed frame so that the Ireland Associates 20,000 lb. tup could be pushed against the Effects Technology Charpy tup with an Enerpac hydraulic device. Both the Ireland Associates 20,000 lb. tup and the Effects Technology's pendulum Charpy tup were attached to warmed-up amplifier units whose outputs were the X and Y inputs of an X-Y plotter.

Several plots were made. In each, the load on the Effects Technology's pendulum Charpy tup was raised to approximately 10,000 lb. and reduced to zero.

The data thus obtained was input to a fortran program using the PANOVA subroutine, and thus a least squares curve fit was made.

Substitution of the first curve fit into the second curve fit resulted in the calibration curve for the pendulum Charpy tup. Furthermore, the standard errors of the calibration constants were calculated from the standard errors of the constants in the first and second curve fits.

2.2.1.2 Drop Tower Tup Calibration

The Ireland Associates drop tower Charpy tup was calibrated in a simpler, but essentially similar, way. First using calibration data provided by Instron Corporation, a calibration curve was found for an Instron universal testing machine by using the PANOVA subroutine. Then the drop tower Charpy tup was attached to a warmed-up amplifier unit whose output was, in turn, attached to an accurate and sensitive voltmeter. It was compressed several times using the Instron universal testing machine in load mode to slightly less than 10,000 lbs. and then unloaded. The output of the amplifier unit was recorded as a function of load and the data were submitted to statistical analysis using PANOVA. The result was a least squares curve fit. Finally, the two curve fits and the associated standard errors were combined, resulting in a calibration curve and associated standard error.

2.2.1.3 Recalibration of the Drop Tower Tup

To confirm the calibration of the Ireland Associates drop tower Charpy tup, it was recalibrated by Tech Science International, Inc. in Seattle, Washington. The tup was connected to the usual power supply amplifier and the amplifier output was connected to Fluke model 8800A voltmeter S/N S10017 N.B.S. Traceability 74689. The tup was placed on top of one of two proving rings and the tup proving ring combination was compressed in a load frame using a hydraulic cylinder. Tinius Olsen's 2000 lb. proving ring S/N 78856 N.B.S. Traceability SJT.01/103371 was used for loads of 0 through 2000 lbs. Tinius Olsen's 20,000 lb. proving ring S/N 68131 N.B.S. Traceability SJT.01/103173 was used for 0 lbs and 2000 through 10,000 lbs. Three replicate tests were performed at each load level.

Since ASTIR subtracts the voltage corresponding to zero load from each voltage in the data, in each replicate test the voltage at zero load was subtracted from the voltages at the other loads. The resulting data were submitted to analysis by a computer program using the PANOVA subroutine.

It was assumed that the load rings' calibration was perfect (i.e., it could be treated as a primary standard), so the calibration curve in this case was calculated using PANOVA in a one-step process.

2.2.2 Determination of Effective Mass

2.2.2.1 Effective Mass of the Pendulum

The effective mass of the pendulum was obtained by the following procedure. First the period of the pendulum was determined as the pendulum swung through a series of swings of less than 15°.

This process was repeated six times. The results from those seven measurements were then averaged. The equation shown in section 5.2.5.2 of ASTM E-23 [39] was used to obtain the radius of percussion. The mass of the pendulum was obtained by demounting the pendulum, weighing it on a shipping scale, and dividing the result by the acceleration of gravity. The center of gravity's radius was found by balancing the demounted pendulum on an angle iron, measuring the distance between the center of gravity and the axis about which the pendulum rotates, and mathematically allowing for the effect of the shaft and bearings on center of gravity position. The radius of strike was directly measured. Equation 1-8 was used to calculate the pendulum's effective mass.

2.2.2.2 Effective Mass of the Drop Tower Crosshead

The effective mass of the drop tower crosshead was obtained by attaching a scale between the release mechanism of the drop tower and the crosshead and reading the weight of the crosshead. The result was then divided by the acceleration of gravity.

2.2.3 Determination of Initial Velocity

2.2.3.1 Velocity of the Pendulum

The velocity of the pendulum just prior to the impact, V_o, was obtained by carefully measuring the difference in height of the tup just before release and the height of the tup just before impact and applying:

$$V_o = (2gh)^{1/2} \tag{2-1}$$

2.2.3.2 Velocity of the Drop Tower Crosshead

Because a strobe light and camera were readily available and the optics needed for Ronchi grating measurements were not, the velocity of the drop tower crosshead just prior to impact was obtained from strobe pictures taken in the vicinity of the impact for a drop from the same height as used for the drop tower Charpy tests. The front of the crosshead was covered with black tape and a thin, white, horizontal tape line was applied to it. After the strobe photos had been taken, a 6" steel scale was taped to the crosshead and photographed to permit measurement of the position of the tape line.

2.2.4 Rise Time of the Amplifier

Performance of the amplifier used to monitor tup output was tested by inputting square waves and storing response using a Tetronix-Sony model 336 recording oscilloscope.

2.3 Impact Tests

2.3.1 Pendulum Impact Tests

A Tinius Olsen model 74 universal impact testing machine (maximum energy 240 ft-lb M_{eff} 60 lb.) configured for instrumented Charpy testing was used for the pendulum experiments. See Figure 2-1. Instead of the usual dynatup instrumentation, a dedicated amplifier unit designed and built by the OGC electronics shop was connected to the tup. The amplifier unit output was received by a COMPUTERSCOPE APL-D2 interface system and tup output data was displayed and recorded by an Apple II+ computer on a 5 1/4" floppy disk.

Five specimens from each heat-treat condition of the 4340 Charpy specimens prepared for this work were tested in the pendulum machine at room temperature. 'Dial energies' were recorded manually and tup output was recorded by instrumentation.

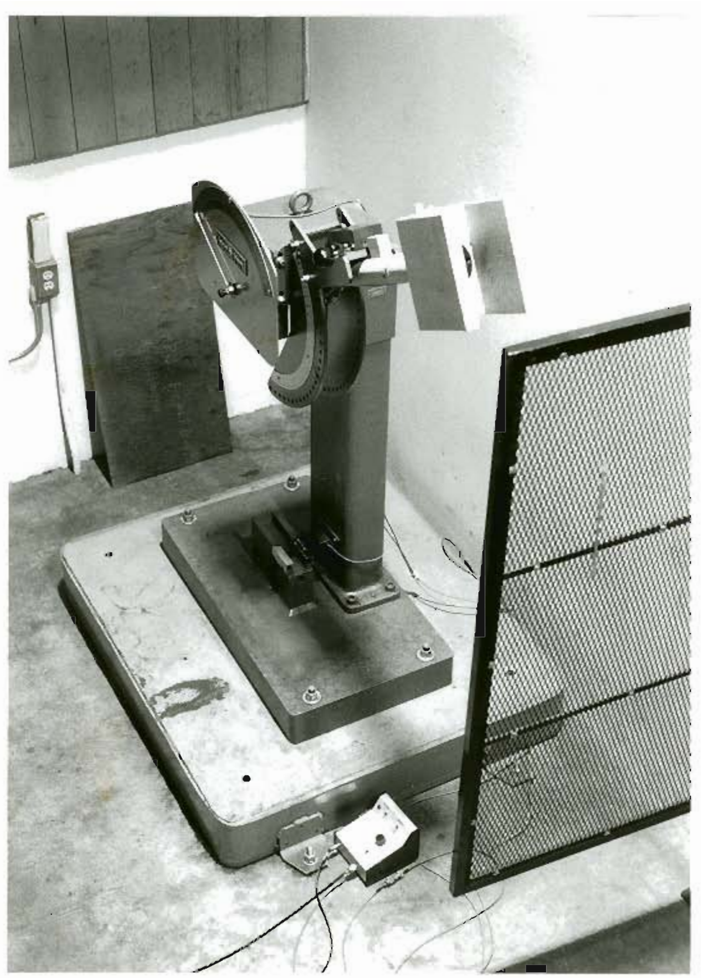


Figure 2-1. Tinius Olsen Model 74 Univeral

Impact Testing Machine

2.3.2 Drop Tower Impact Tests

The apparatus used in the drop tower portion of the experiment is shown in Figure 2-2. It has a 180 lb. crosshead and 10 ft. maximum drop height. For this part of the experiment, an instrumented drop tower Charpy tup designed and built by Ireland Associates (see Figure 2-3) was used in conjunction with the same instrumentation used in the pendulum portion of the experiment.

Arrestor blocks were prepared from Alloy 6061 aluminum and heat treated to place them in the "O" temper. [40].

Five specimens from the softer heat-treat condition and six specimens from the harder heat-treat condition were tested in the drop tower at room temperature.

2.4 Impact Test Calculations

A program called ASTIR utilizing the approach to energy calculation outlined in the Introduction was written and is described in detail in Appendix A. ASTIR is capable of performing calculations using linear or quadratic parabolic tup calibrations and versions of ASTIR exist which perform energy calculations using the zeroth, first,

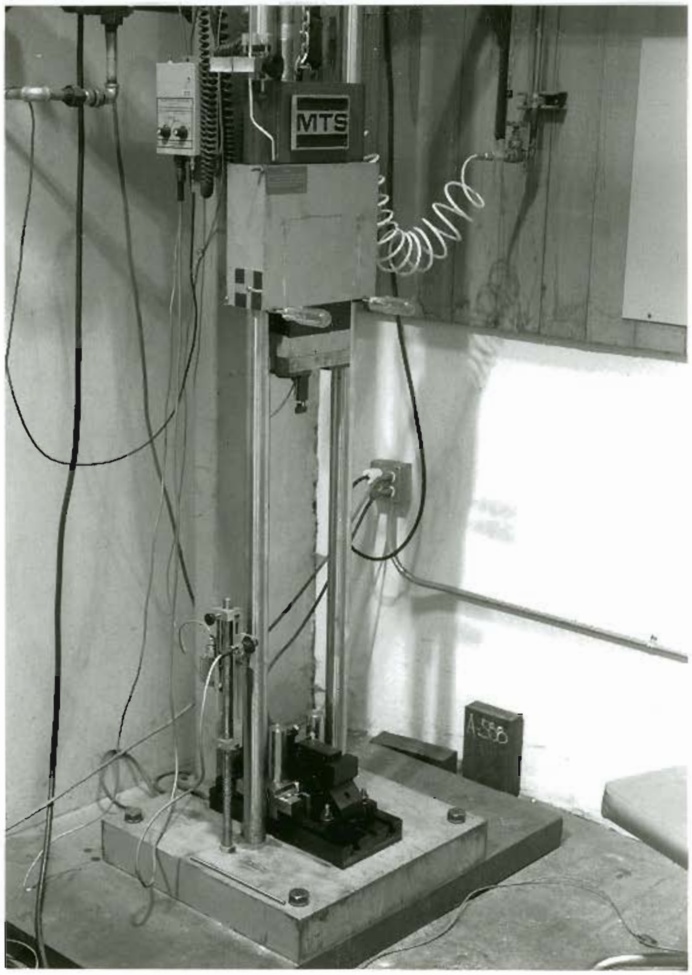


Figure 2-2. Drop Tower

Figure 2-3. Ireland Associates Drop Tower Charpy Tup

and second order equations. All the 'tup energy' results for all four sets of tests were analyzed by ASTIR, and hard copy was made for the pendulum and drop tower test results. System compliance data were calculated using ASTIR. Several combinations of calibration equations and integration methods were used seeking the best technique.

Mean and standard deviations were calculated for all conditions tested, and F-test and T-test comparisons were made between:

1. E-23 'dial energy' and pendulum 'tup energy;'
2. Pendulum 'tup energy' and drop tower 'tup energy;' and
3. E-23 'dial energy' and drop tower 'tup energy.'

3. RESULTS

3.1 Calibration Results

3.1.1 Tup Calibration

Both the Effects Technology and the Ireland Associates tups were calibrated for this experiment. The results of these calibrations are summarized below:

3.1.1.1 Pendulum Tup Calibration

Table 3-1 summarizes the data gathered during calibration of the Ireland Associates 20,000 lb tup (attached to the OGC-designed amplifier unit) against the Tinius Olsen tensile machine. Tup output, shown in the body of the table, is given in volts; tensile machine output (column headings) is in pounds. The data in Table 3-1 are plotted in Figure 3-1.

Table 3-1
Calibration of Ireland Associates 20,000 lb Tup

Rep. #	Load [lb] 0000	1000	2000	3000	4000	5000
1	0.0570	0.0835	0.1065	0.1285	0.1510	0.1735
2	0.0570	0.0840	0.1065	0.1300	0.1510	0.1735
3	0.0570	0.0810	0.1030	0.1280	0.1490	0.1730
4	0.0570	0.0845	0.1065	0.1280	0.1535	0.1750
5	0.0580	0.0775	0.1003	0.1225	0.1425	0.1693
6	0.0580	0.0830	0.1045	0.1285	0.1503	0.1750
7	0.0580	0.0810	0.1043	0.1268	0.1515	0.1740
8	0.0580	0.0850	0.1080	0.1305	0.1535	0.1763
9	0.0580	0.0805	0.1035	0.1250	0.1450	0.1703
10	0.0580	0.0835	0.1055	0.1285	0.1505	0.1738

Rep. #	Load [lb] 6000	7000	8000	9000	10000
1	0.1960	0.2180	0.2415	0.2670	0.2870
2	0.1970	0.2185	0.2450	0.2655	0.2890
3	0.1960	0.2190	0.2405	0.2640	0.2870
4	0.1985	0.2215	0.2430	0.2675	0.2915
5	0.1920	0.2150	0.2350	0.2590	0.2855
6	0.1950	0.2180	0.2403	0.2653	0.2875
7	0.1945	0.2168	0.2405	0.2638	0.2848
8	0.1990	0.2198	0.2433	0.2665	0.2895
9	0.1943	0.2163	0.2390	0.2620	0.2855
10	0.1948	0.2190	0.2405	0.2645	0.2895

Table 3-2 presents the data from the Effects Technology Inc. pendulum Charpy tup which was calibrated against the Ireland Associates 20,000 lb tup. The first line of the table is the Ireland Associates 20,000 lb tup-amplifier unit output. The body of the

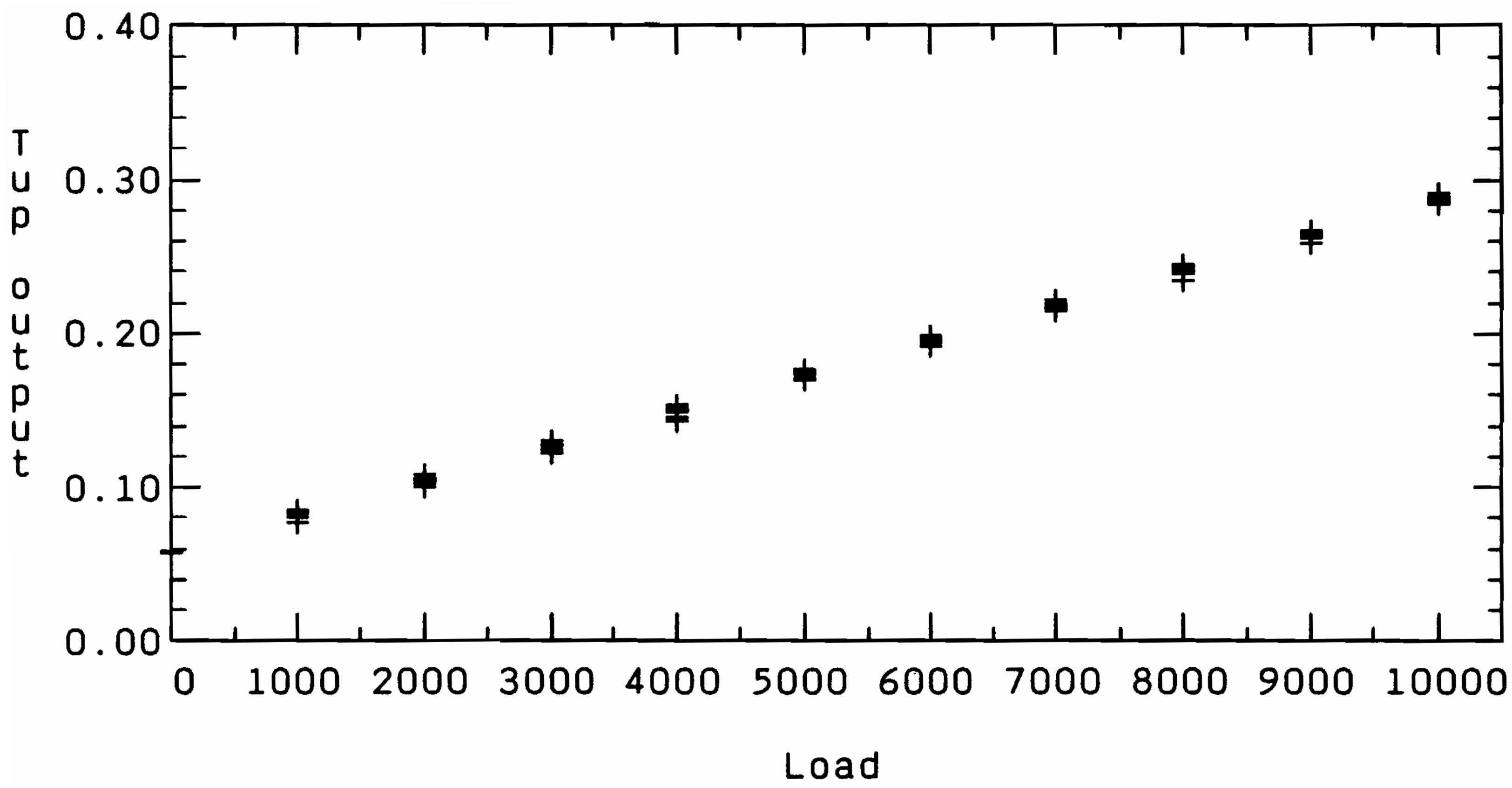


Figure 3-1. Calibration of Ireland Associates 20,000 lb. Tup

table shows the Effects Technology-amplifier unit output. Both outputs are given in volts. The data in Table 3-2 are plotted in Figure 3-2.

Table 3-2
Calibration of Effects Technology Pendulum Charpy Tup
Against the Ireland Associates 20,000 lb Tup

	Tup-Amp Output						
	0.0560	0.0810	0.1060	0.1310	0.1560	0.1810	0.2060
Rep #							
1	-0.0480	0.0260	0.1105	0.1845	0.2560	0.3375	0.4090
2	-0.0480	0.0260	0.1025	0.1805	0.2560	0.3340	0.4080
3	-0.0480	0.0280	0.1055	0.1830	0.2605	0.3355	0.4080
4	-0.0480	0.0280	0.1055	0.1835	0.2570	0.3305	0.4065
5	-0.0480	0.0260	0.1105	0.1855	0.2595	0.3320	0.4070
6	-0.0480	0.0280	0.1070	0.1850	0.2630	0.3370	0.4105
7	-0.0480	0.0355	0.1120	0.1905	0.2655	0.3470	0.4150
8	-0.0480	0.0310	0.1105	0.1845	0.2625	0.3375	0.4110
9	-0.0480	0.0305	0.1100	0.1870	0.2655	0.3375	0.4090
10	-0.0480	0.0280	0.1090	0.1830	0.2605	0.3355	0.4080
11	-0.0480	0.0320	0.1090	0.1835	0.2620	0.3355	0.4065
12	-0.0480	0.0320	0.1105	0.1855	0.2595	0.3370	0.4145
13	-0.0480	0.0330	0.1110	0.1885	0.2670	0.3390	0.4105
14	-0.0480	0.0355	0.1155	0.1905	0.2655	0.3470	0.4150

The output of a statistical program using PANOVA analyzing the data in Tables 3-1 and 3-2 is shown in Tables 3-3(A), 3-3(B), and 3-3(C).

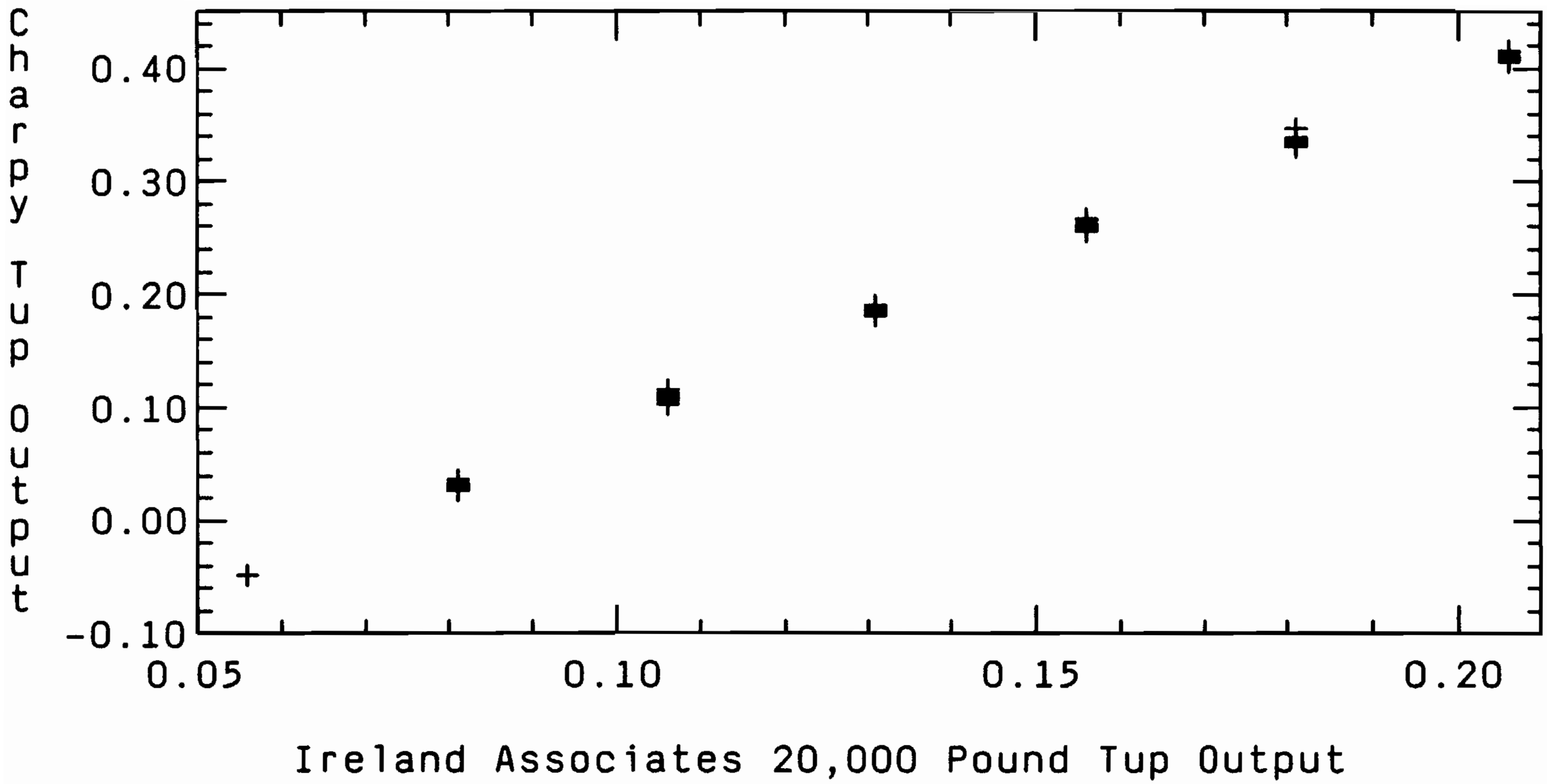


Figure 3-2. Calibration of Effects Technology Pendulum Charpy Tup

Table 3-3(A)(1)
Statistical Analysis of Data in Table 3-1
(Calibration of Ireland Associates Tup)

Calibration Curve:

$$\hat{Y} = b0 + b1 * X$$

where:

$\hat{Y}$	≡	Output of amplifier attached to Ireland Associates 20,000 lb tup in volts
b0, b1	≡	Calibration coefficients
X	≡	Load in pounds

Specific Calibration Curve:

$$\hat{Y} = + .58681e-01 + .22855e-04\ X$$

Source	Sum of Squares	Degrees of Freedom	Mean Square
Due to Regression	0.5746	1	0.5746
Pure Error	0.0005	99	0.0000
Lack of Fit	0.0000	9	0.0000
About Regression	0.0006	108	0.0000
Total	0.5752	109	0.0053
Due to this Order	0.5746	1	0.5746
Degrees of Freedom	1 & 108		

Coefficient	b0	b1
Value	0.58681e-01	0.22855e-04
Standard Error	0.34540e-03	0.69080e-07
Two Sided T Ratio	0.16989e+03	0.33085e+03
Degrees of Freedom	108	

Table 3-3(A)(2)
Statistical Analysis of Data in Table 3-1
(Calibration of Ireland Associates Tup)
(continued)

Calibration Curve:

$$\hat{Y} = b0 + b1 * X + b2 * X^2$$

where:

$\hat{Y}$	≡	Output of amplifier attached to Ireland Associates 20,000 lb tup in volts
b0, b1, b2	≡	Calibration coefficients
X	≡	Load in pounds

Specific Calibration Curve:

$$\hat{Y} = + .58622e{-}01 + .22895e{-}04\ X - .39510e{-}011\ X^2$$

Source	Sum of Squares	Degrees of Freedom	Mean Square
Due to Regression	0.5746	2	0.2873
Pure Error	0.0005	99	0.0000
Lack of Fit	0.0000	8	0.0000
About Regression	0.0006	107	0.0000
Total	0.5752	109	0.0053
Due to this Order	0.0000	1	0.0000
F Ratio for Improvement Due to Order 2:		0.25286e-01	
Degrees of Freedom	1 & 107		

Coefficient	b0	b1	b2
Value	0.58622e-01	0.22895e-04	-.39510e-11
Standard Error	0.50921e-03	0.25798e-06	0.24847e-10
Two Sided T Ratio	0.11512e+03	0.88748e+02	0.15902e+00
Degrees of Freedom	107		

Table 3-3(A)(3)
Statistical Analysis of Data in Table 3-1
(Calibration of Ireland Associates Tup)
(continued)

Calibration Curve:

$$\bar{Y} = b0 + b1 * X + b2 * X^2 + b3 * X^3$$

where:

$\bar{Y}$	≡	Output of amplifier attached to Ireland Associates 20,000 lb tup in volts
b0, b1, b2, b3	≡	Calibration coefficients
X	≡	Load in pounds

Specific Calibration Curve:

$$\bar{Y} = + .57982e\text{-}01 + .23912e\text{-}04\ X - .27068e\text{-}09\ X^2 + .17782e\text{-}13\ X^3$$

Source	Sum of Squares	Degrees of Freedom	Mean Square
Due to Regression	0.5746	3	0.1915
Pure Error	0.0005	99	0.0000
Lack of Fit	0.0000	7	0.0000
About Regression	0.0005	106	0.0000
Total	0.5752	109	0.0053
Due to this Order	0.0000	1	0.0000
F Ratio for Improvement Due to Order 3:		0.37834e+01	
Degrees of Freedom	1 & 106		

Coefficient	b0	b1	b2
Value	0.57982e-01	0.23912e-04	-.27068e-09
Standard Error	0.60086e-03	0.58163e-06	0.13930e-09
Two Sided T Ratio	0.96499e+02	0.41112e+02	0.19431e+01
Degrees of Freedom	106		

Coefficient	b3
Value	0.17782e-04
Standard Error	0.91418e-05
Two Sided T Ratio	0.19451e+01
Degrees of Freedom	106

Table 3-3(B)(1)
Statistical Analysis of Data in Table 3-2
(Calibration of Effects Technology Tup)

Calibration Curve:

$$\hat{Y} = b0 + b1 * X$$

where: $\hat{Y}$ ≡ Output of Effects Technology tup in volts
b0, b1 ≡ Calibration coefficients
X ≡ Output of Ireland Associates 20,000 lb tup in volts

Specific Calibration Curve:

$$\hat{Y} = - .21700e{+}00 + .30580e{+}01\ X$$

Source	Sum of Squares	Degrees of Freedom	Mean Square
Due to Regression	2.2911	1	2.2911
Pure Error	0.0010	91	0.0000
Lack of Fit	0.0003	5	0.0001
About Regression	0.0013	96	0.0000
Total	2.2924	97	0.0236
Due to this Order	2.2911	1	2.2911
Degrees of Freedom	1 & 96		

Coefficient	b0	b1
Value	-.21700e+00	0.30580e+01
Standard Error	0.97873e-03	0.74712e-02
Two Sided T Ratio	0.22172e+03	0.40931e+03
Degrees of Freedom	96	

Table 3-3(B)(2)
Statistical Analysis of Data in Table 3-2
(Calibration of Effects Technology Tup)
(continued)

Calibration Curve:

$$\hat{Y} = b0 + b1 * X + b2 * X^2$$

where: $\hat{Y}$ ≡ Output of Effects Technology tup in volts
b0, b1, b2 ≡ Calibration coefficients
X ≡ Output of Ireland Associates 20,000 lb tup in volts

Specific Calibration Curve:

$$\hat{Y} = - .22926e{+}00 + .32771e{+}01\ X - .83605e{+}00\ X^2$$

Source	Sum of Squares	Degrees of Freedom	Mean Square
Due to Regression	2.2914	2	1.1457
Pure Error	0.0010	91	0.0000
Lack of Fit	0.0000	4	0.0000
About Regression	0.0010	95	0.0000
Total	2.2924	97	0.0236
Due to this Order	0.0003	1	0.0003
F Ratio for Improvement Due to Order 2:		0.30758e+02	
Degrees of Freedom	1 & 95		

Coefficient	b0	b1	b2
Value	-.22926e+00	0.32771e+01	-.83605e+00
Standard Error	0.23698e-02	0.40032e-01	0.15075e+00
Two Sided T Ratio	0.96742e+02	0.81860e+02	0.55460e+01
Degrees of Freedom	95		

Table 3-3(B)(3)
Statistical Analysis of Data in Table 3-2
(Calibration of Effects Technology Tup)
(continued)

Calibration Curve:

$$\hat{Y} = b0 + b1 * X + b2 * X^2 + b3 * X^3$$

where: $\hat{Y}$ ≡ Output of Effects Technology tup in volts
b0, b1, b2, b3 ≡ Calibration coefficients
X ≡ Load in pounds

Specific Calibration Curve:

$$\hat{Y} = -.22426e{+}00 + .31365e{+}01\ X + .33671e{+}00\ X^2 - .29841e{+}01\ X^3$$

Source	Sum of Squares	Degrees of Freedom	Mean Square
Due to Regression	2.2914	3	0.7638
Pure Error	0.0010	91	0.0000
Lack of Fit	0.0000	3	0.0000
About Regression	0.0010	94	0.0000
Total	2.2924	97	0.0236
Due to this Order	0.0000	1	0.0000
F Ratio for Improvement Due to Order 3:		0.62729e+00	
Degrees of Freedom	1 & 94		

Coefficient	b0	b1	b2
Value	-.22426e+00	0.31365e+01	0.33671e+00
Standard Error	0.67428e-02	0.18197e+00	0.14884e+01
Two Sided T Ratio	0.33259e+02	0.17236e+02	0.22622e+00
Degrees of Freedom	94		

Coefficient	b3
Value	-.29841e+01
Standard Error	0.37678e+01
Two Sided T Ratio	0.79201e+00
Degrees of Freedom	94

Table 3-3(B)(4)
Statistical Analysis of Data in Table 3-2
(Calibration of Effects Technology Tup)
(continued)

Calibration Curve:

$$\hat{Y} = b0 + b1 * X + b2 * X^2 + b3 * X^3 + b4 * X^4$$

where: $\hat{Y}$ ≡ Output of Effects Technology tup in volts
b0, b1, b2, b3, b4 ≡ Calibration coefficients
X ≡ Output of Ireland Associates 20,000 lb tup in volts

Specific Calibration Curve:

$$\hat{Y} = -.22908e{+}00 + .33191e{+}01\ X - .20480e{+}01\ X^2 + .99004e{+}01\ X^3 - .24589e{+}02\ X^4$$

Source	Sum of Squares	Degrees of Freedom	Mean Square
Due to Regression	2.2914	4	0.5729
Pure Error	0.0010	91	0.0000
Lack of Fit	0.0000	2	0.0000
About Regression	0.0010	93	0.0000
Total	2.2924	97	0.0236
Due to this Order	0.0000	1	0.0000

F Ratio for Improvement Due to Order 4: 0.55212e-01
Degrees of Freedom 1 & 93

Coefficient	b0	b1	b2
Value	-.22908e+00	0.33191e+01	-.20480e+01
Standard Error	0.21587e-01	0.79823e+00	0.10259e+02
Two Sided T Ratio	0.10612e+02	0.41580e+01	0.19964e+00
Degrees of Freedom	93		

Coefficient	b3	b4
Value	0.99004e+01	-.24589e+02
Standard Error	0.54965e+02	0.10465e+03
Two Sided T Ratio	0.18012e+00	0.23497e+00
Degrees of Freedom	93	

The linear curve fit from Table 3-3(A)(1) was substituted into the quadratic curve fit from Table 3-3(B)(2). The resulting equation was solved and is presented in Table 3-3(C). Statistical data for the equation in Table 3-3(C) were derived from the statistical data in Tables 3-3(A)(1) and 3-3(B)(2). These derived data are included in Table 3-3(C).

Table 3-3(C)
Calibration Curve for Effects Technology Pendulum Charpy Tup

Calibration Curve:

$$\hat{Y} = \text{Zerovolts} + \text{Loadfac1} * X + \text{Loadfac2} * X^2$$

where: $\hat{Y}$ ≡ Output of Effects Technology Pendulum Charpy tup in volts

Zerovolts, Loadfac1, Loadfac2 ≡ Calibration coefficients

X ≡ Load in pounds

Specific Calibration Curve:

$$\hat{Y} = \text{Zerovolts} + .72656\text{e-}04\ X - .43673\text{e-}09\ X^2$$

Coefficient	Zerovolts	Loadfac1	Loadfac2
Value	-.39837e-01	0.72656e-04	-.43673e-09
Standard Error	0.63361e-02	0.15257e-05	0.76107e-10
Two Sided T Ratio	0.62873e+01	0.47621e+02	0.57383e+01
Degrees of Freedom	202		

3.1.1.2 Drop Tower Tup Calibration

The Instron calibration data supplied by Instron Corp. are shown in Table 3-4. Unfortunately, the manufacturer did not provide replicate data so an estimate of pure error and hence goodness of fit is not possible. The data in Table 3-4 are plotted in Figure 3-3.

Table 3-4
Calibration of Voltage Output of
Instron Universal Tensile Testing Machine

Output Voltage [Volts]	Applied Load [lb]
1.000	1261.0
2.000	2517.0
3.000	3717.0
4.000	5020.0
5.000	6267.0
6.000	7512.0
7.000	8762.0
8.000	10015.0
9.000	11257.0
10.000	12502.0

The data gathered during calibration of the Ireland Associates drop tower Charpy tup against the Instron tensile testing machine output is displayed in Table 3-5 in volts. The data in Table 3-5 are plotted in Figure 3-4.

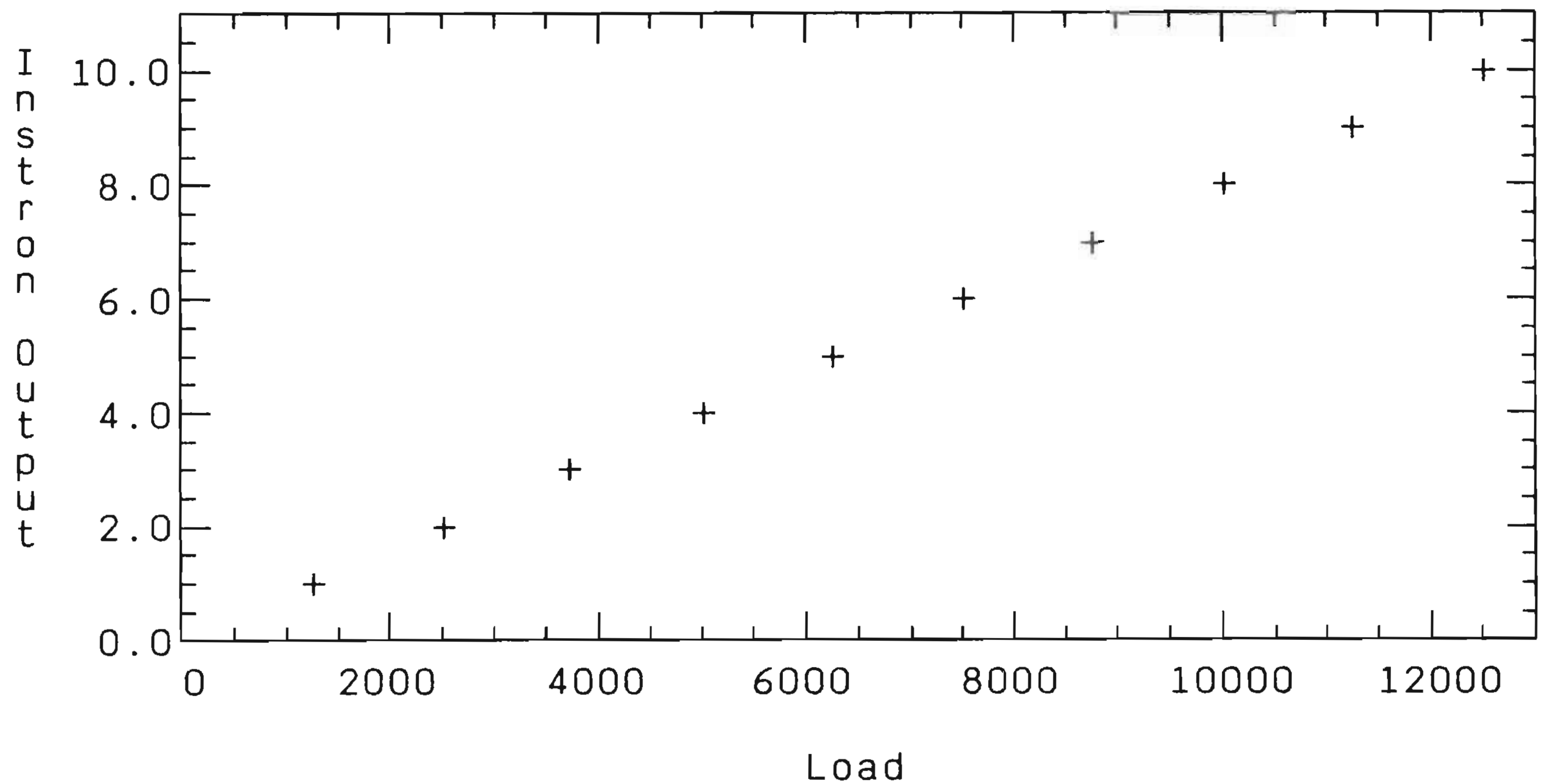


Figure 3-3. Calibration of Voltage Output of Instron Universal Tensile Testing Machine

Table 3-5
Calibration of Ireland Associates Drop Tower Charpy Tup Against the Instron Universal Tensile Testing Machine

Instron Output	Drop Tower Charpy Tup Test Repetitions					
-0.0	-0.2704	-0.2708	-0.2706	-0.2699	-0.2694	-0.2698
-0.5	-0.2344	-0.2362	-0.2344	-0.2353	-0.2346	-0.2343
-1.0	-0.1963	-0.1991	-0.1966	-0.1986	-0.1977	-0.1964
-1.5	-0.1598	-0.1620	-0.1601	-0.1612	-0.1605	-0.1596
-2.0	-0.1234	-0.1256	-0.1238	-0.1255	-0.1244	-0.1230
-2.5	-0.0872	-0.0895	-0.0873	-0.0891	-0.0886	-0.0869
-3.0	-0.0514	-0.0536	-0.0508	-0.0535	-0.0531	-0.0511
-3.5	-0.0154	-0.0175	-0.0154	-0.0178	-0.0171	-0.0149
-4.0	+0.0198	+0.0180	+0.0198	+0.0181	+0.0187	+0.0202
-4.5	+0.0559	+0.0539	+0.0557	+0.0538	+0.0543	+0.0565
-5.0	+0.0915	+0.0890	+0.0912	+0.0895	+0.0897	+0.0915
-5.5	+0.1275	+0.1251	+0.1268	+0.1250	+0.1253	+0.1272
-6.0	+0.1625	+0.1605	+0.1622	+0.1598	+0.1606	+0.1628
-6.5	+0.1978	+0.1966	+0.1976	+0.1957	+0.1964	+0.1976
-7.0	+0.2333	+0.2320	+0.2332	+0.2332	+0.2320	+0.2332
-7.5	+0.2689	+0.2686	+0.2686	+0.2673	+0.2677	+0.2685

The output of a statistics program using PANOVA to analyze the data in Tables 3-4 and 3-5 is shown in Tables 3-6(A), 3-6(B) and 3-6(C).

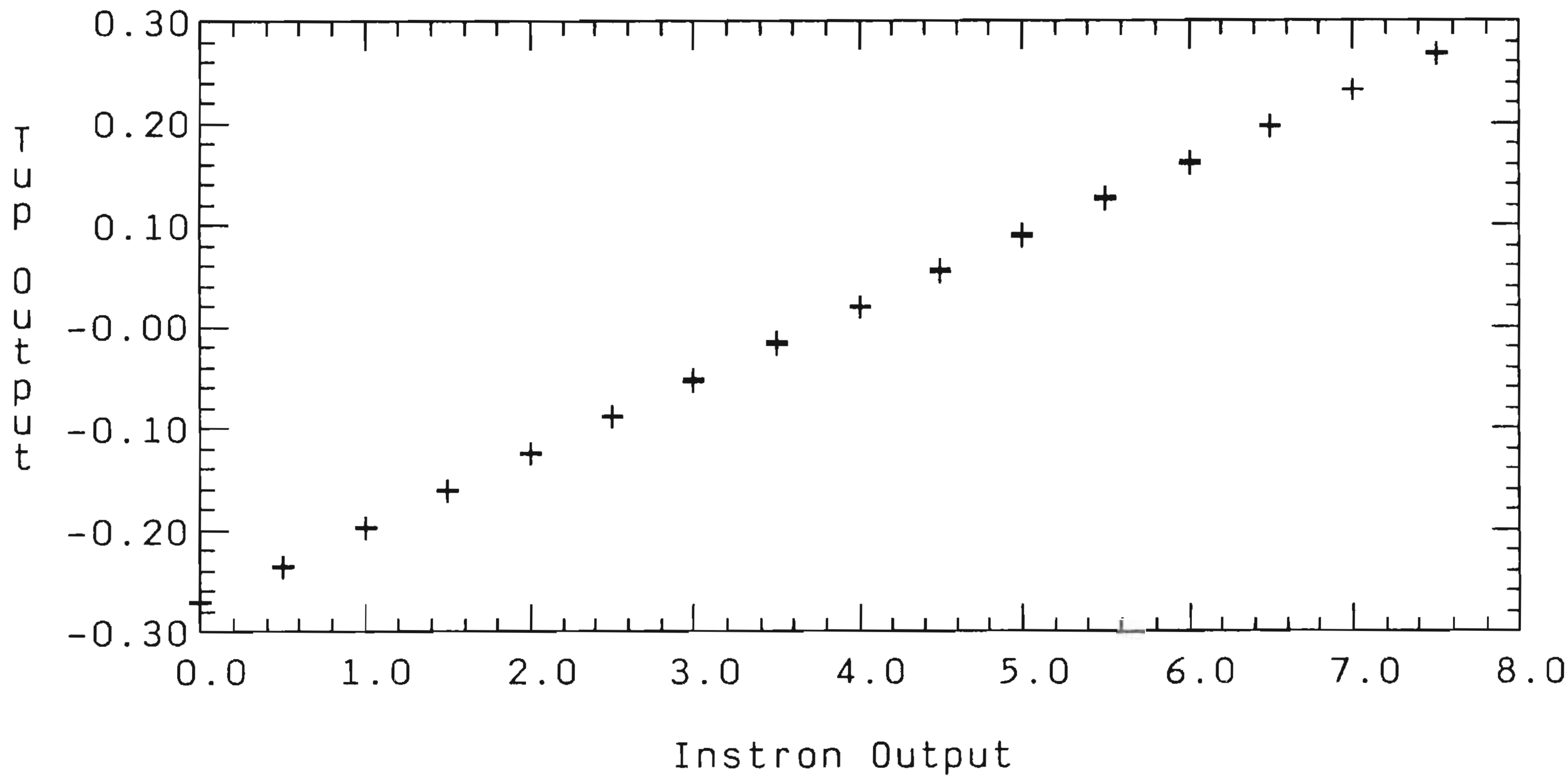


Figure 3-4. Calibration of Ireland Associates Drop Tower Charpy Tup

Table 3-6(A)(1)
Statistical Analysis of Data in Table 3-4
(Calibration of Instron Universal Tensile Testing Machine)

Calibration Curve:

$$\hat{Y} = b0 + b1 * X$$

where:	$\hat{Y}$	≡	Output of Instron Universal Tensile Testing Machine in volts
	b0, b1	≡	Calibration coefficients
	X	≡	Load in pounds

Specific Calibration Curve:

$$\hat{Y} = - .47088e-02 + .79975e-03 X$$

Source	Sum of Squares	Degrees of Freedom	Mean Square
Due to Regression	82.4987	1	82.4987
About Regression	0.0013	8	0.0002
Total	82.5000	9	9.1667
Due to this Order	82.4987	1	82.4987
Degrees of Freedom	1 & 8		

Coefficient	b0	b1
Value	-.47088e-02	0.79975e-03
Standard Error	0.78588e-02	0.11418e-05
Two Sided T Ratio	0.59918e+00	0.70045e+03
Degrees of Freedom	8	

Table 3-6(A)(2)
Statistical Analysis of Data in Table 3-4
(Calibration of Instron Universal Tensile Testing Machine)
(continued)

Calibration Curve:

$$\bar{Y} = b0 + b1 * X + b2 * X^2$$

where: $\hat{Y}$ ≡ Output of Instron Universal Tensile Testing Machine in volts
b0, b1, b2 ≡ Calibration coefficients
X ≡ Load in pounds

Specific Calibration Curve:

$$\bar{Y} = + .91250e-04 + .79784e-03 X + .13937e-09 X^2$$

Source	Sum of Squares	Degrees of Freedom	Mean Square
Due to Regression	82.4987	2	41.2493
About Regression	0.0013	7	0.0002
Total	82.5000	9	9.1667
Due to this Order	0.0000	1	0.0000

F Ratio for Improvement Due to Order 2: 0.13213e+00
Degrees of Freedom 1 & 7

Coefficient	b0	b1	b2
Value	0.91250e-04	0.79784e-03	0.13937e-09
Standard Error	0.15609e-01	0.54127e-05	0.38340e-09
Two Sided T Ratio	0.58458e-02	0.14740e+03	0.36350e+00
Degrees of Freedom	7		

Table 3-6(A)(3)
Statistical Analysis of Data in Table 3-4
(Calibration of Instron Universal Tensile Testing Machine)
(continued)

Calibration Curve:

$$\hat{Y} = b0 + b1 * X + b2 * X^2 + b3 * X^3$$

where: $\hat{Y}$ ≡ Output of Instron Universal Tensile Testing Machine in volts
b0, b1, b2, b3 ≡ Calibration coefficients
X ≡ Load in pounds

Specific Calibration Curve:

$$\ddot{Y} = - .21525e-01 + .81311e-03 X - .25030e-08 X^2 + .12793e-12 X^3$$

Source	Sum of Squares	Degrees of Freedom	Mean Square
Due to Regression	82.4989	3	27.4996
About Regression	0.0011	6	0.0002
Total	82.5000	9	9.1667
Due to this Order	0.0002	1	0.0002
F Ratio for Improvement Due to Order 3:		0.10172e+01	
Degrees of Freedom	1 & 6		

Coefficient	b0	b1	b2
Value	-.21525e-01	0.81311e-03	-.25030e-08
Standard Error	0.26503e-01	0.16076e-04	0.26478e-08
Two Sided T Ratio	0.81217e+00	0.50578e+02	0.94533e+00
Degrees of Freedom	6		

Coefficient	b3
Value	0.12793e-12
Standard Error	0.12684e-12
Two Sided T Ratio	0.10086e+01
Degrees of Freedom	6

Table 3-6(B)(1)
Statistical Analysis of Data in Table 3-5
(Calibration of Ireland Associates Drop Tower Charpy Tup)

Calibration Curve:

$$\hat{Y} = b0 + b1 * X$$

where: $\hat{Y}$ ≡ Output of Ireland Associates drop tower Charpy tup in volts
b0, b1 ≡ Calibration coefficients
X ≡ Output of Instron Universal Tensile Testing Machine in volts

Specific Calibration Curve:

$$\hat{Y} = -\ .26868e+00 + .71749e-01\ X$$

Source	Sum of Squares	Degrees of Freedom	Mean Square
Due to Regression	2.6255	1	2.6255
Pure Error	0.0001	80	0.0000
Lack of Fit	0.0001	14	0.0000
About Regression	0.0002	94	0.0000
Total	2.6256	95	0.0276
Due to this Order	2.6255	1	2.6255

F Ratio for Improvement Due to Order 1: 0.13692e+07
Degrees of Freedom 1 & 94

Coefficient	b0	b1
Value	-.26868e+00	0.71749e-01
Standard Error	0.22994e-03	0.61318e-04
Two Sided T Ratio	0.11684e+04	0.11701e+04
Degrees of Freedom	94	

Table 3-6(B)(2)
Statistical Analysis of Data in Table 3-5
(Calibration of Ireland Associates Drop Tower Charpy Tup)
(continued)

Calibration Curve:

$$\hat{Y} = b0 + b1 * X + b2 * X^2$$

where: $\hat{Y}$ ≡ Output of Ireland Associates drop tower Charpy tup in volts

b0, b1, b2 ≡ Calibration coefficients

X ≡ Output of Instron Universal Tensile Testing Machine in volts

Specific Calibration Curve:

$$\hat{Y} = -.27037e{+}00 + .73198e{-}01\ X - .19324e{-}03\ X^2$$

Source	Sum of Squares	Degrees of Freedom	Mean Square
Due to Regression	2.6255	2	1.3128
Pure Error	0.0001	80	0.0000
Lack of Fit	0.0000	13	0.0000
About Regression	0.0001	93	0.0000
Total	2.6256	95	0.0276
Due to this Order	0.0001	1	0.0001
F Ratio for Improvement Due to Order 2:		0.74193e+02	
Degrees of Freedom	1 & 93		

Coefficient	b0	b1	b2
Value	-.27037e+00	0.73198e-01	-.19324e-03
Standard Error	0.26127e-03	0.17443e-03	0.22435e-04
Two Sided T Ratio	0.10348e+04	0.41965e+03	0.86135e+01
Degrees of Freedom	93		

Table 3-6(B)(3)
Statistical Analysis of Data in Table 3-5
(Calibration of Ireland Associates Drop Tower Charpy Tup)
(continued)

Calibration Curve:

$$\hat{Y} = b0 + b1 * X + b2 * X^2 + b3 * X^3$$

where: $\hat{Y}$ ≡ Output of Ireland Associates drop tower Charpy tup in volts
b0, b1, b2, b3 ≡ Calibration coefficients
X ≡ Output of Instron Universal Tensile Testing Machine in volts

Specific Calibration Curve:

$$\hat{Y} = -.27085e{+}00 + .74120e{-}01\ X - .51068e{-}03\ X^2 + .28217e{-}04\ X^3$$

Source	Sum of Squares	Degrees of Freedom	Mean Square
Due to Regression	2.6255	3	0.8752
Pure Error	0.0001	80	0.0000
Lack of Fit	0.0000	12	0.0000
About Regression	0.0001	92	0.0000
Total	2.6256	95	0.0276
Due to this Order	0.0000	1	0.0000
F Ratio for Improvement Due to Order 3:		0.66619e+01	
Degrees of Freedom	1 & 92		

Coefficient	b0	b1	b2
Value	-.27085e+00	0.74120e-01	-.51068e-03
Standard Error	0.31486e-03	0.39532e-03	0.12490e-03
Two Sided T Ratio	0.86021e+03	0.18749e+03	0.40887e+01
Degrees of Freedom	92		

Coefficient	b3
Value	0.28217e-04
Standard Error	0.10932e-04
Two Sided T Ratio	0.25811e+01
Degrees of Freedom	92

The linear curve fit from Table 3-6(A)(1) was substituted into the linear curve fit from Table 3-6(B)(1). The resulting equation was solved and is presented in Table 3-6(C)(1). The linear curve fit from Table 3-6(A)(1) was substituted into the quadratic curve fit from Table 3-6(B)(2). The resulting equation was solved and is presented in Table 3-6(C)(2). Statistical data for the equations in Tables 3-6(C) were derived from the appropriate data in Table 3-6(A)(1) and Tables 3-6(B). These derived data are included in Tables 3-6(C).

Table 3-6(C)(1)
Calibration Curve for Ireland Associates Drop Tower Charpy Tup
Due to Instron Calibration

Calibration Curve:

$$\hat{Y} = \text{Zerovolts} + \text{Loadfac1} * X$$

where:	$\hat{Y}$	≡	Output of Ireland Associates drop tower Charpy tup in volts
	Zerovolts, Loadfac1	≡	Calibration coefficients
	X	≡	Load in pounds

Specific Calibration Curve:

$$\hat{Y} = \text{Zerovolts} + .57382e\text{-}04\ X$$

Coefficient	b1
Value	0.57382e-04
Standard Error	0.44005e-05
Two Sided T Ratio	0.13040e+02
Degrees of Freedom	101

Table 3-6(C)(2)
Calibration Curve for Ireland Associates Drop Tower Charpy Tup
Due to Instron Calibration

Calibration Curve:

$$\hat{Y} = \text{Zerovolts} + \text{Loadfac1} * X + \text{Loadfac2} * X^2$$

where:	$\hat{Y}$	≡	Output of Ireland Associates drop tower Charpy tup in volts
	Zerovolts, Loadfac1 Loadfac2	≡	Calibration coefficients
	X	≡	Load in pounds

Specific Calibration Curve:

$$\hat{Y} = \text{Zerovolts} + .58542\text{e-}04\ X - 0.12360\text{e-}09\ X^2$$

Coefficient	b1	b2
Value	0.58542e-04	-.12360e-09
Standard Error	0.22048e-06	0.13997e-10
Two Sided T Ratio	0.26552e+03	0.88307e+01
Degrees of Freedom	100	

3.1.1.3 Drop Tower Tup Recalibration

The load ring calibration data for the Ireland Associates drop tower Charpy tup is shown in Table 3-7. The data in Table 3-7 are plotted in Figure 3-5.

Table 3-7
Load Ring Calibration Data for
Ireland Associates Drop Tower Charpy Tup

Load [lb]	Tup Output [volts]	Tup Output [volts]	Tup Output [volts]
0	0.0000	0.0000	0.0000
2000	0.1065	0.1025	0.1058
2500	0.1339	0.1287	0.1327
3000	0.1615	0.1550	0.1599
3500	0.1888	0.1818	0.1870
4000	0.2169	0.2085	0.2140
4500	0.2447	0.2354	0.2412
5000	0.2722	0.2621	0.2683
5500	0.3003	0.2899	0.2958
6000	0.3280	0.3168	0.3227
6500	0.3557	0.3551	0 3499
7000	0.3834	0.3722	0.3771
7500	0.4128	0.4005	0.4051
8000	0.4411	0.4289	0.4324
8500	0.4692	0.4568	0.4592
9000	0.4971	0.4852	0.4869
9500	0.5259	0.5131	0.5141
10000	0.5531	0.5402	0.5412
0	0.0000	0.0000	0.0000
500	0.0212	0.0223	0.0218
1000	0.0463	0.0475	0.0468
1500	0.0717	0.0732	0.0725
2000	0.0977	0.0995	0.0989

The output of a statistical program using PANOVA to analyze the data in Table 3-7 is shown in Table 3-8(A).

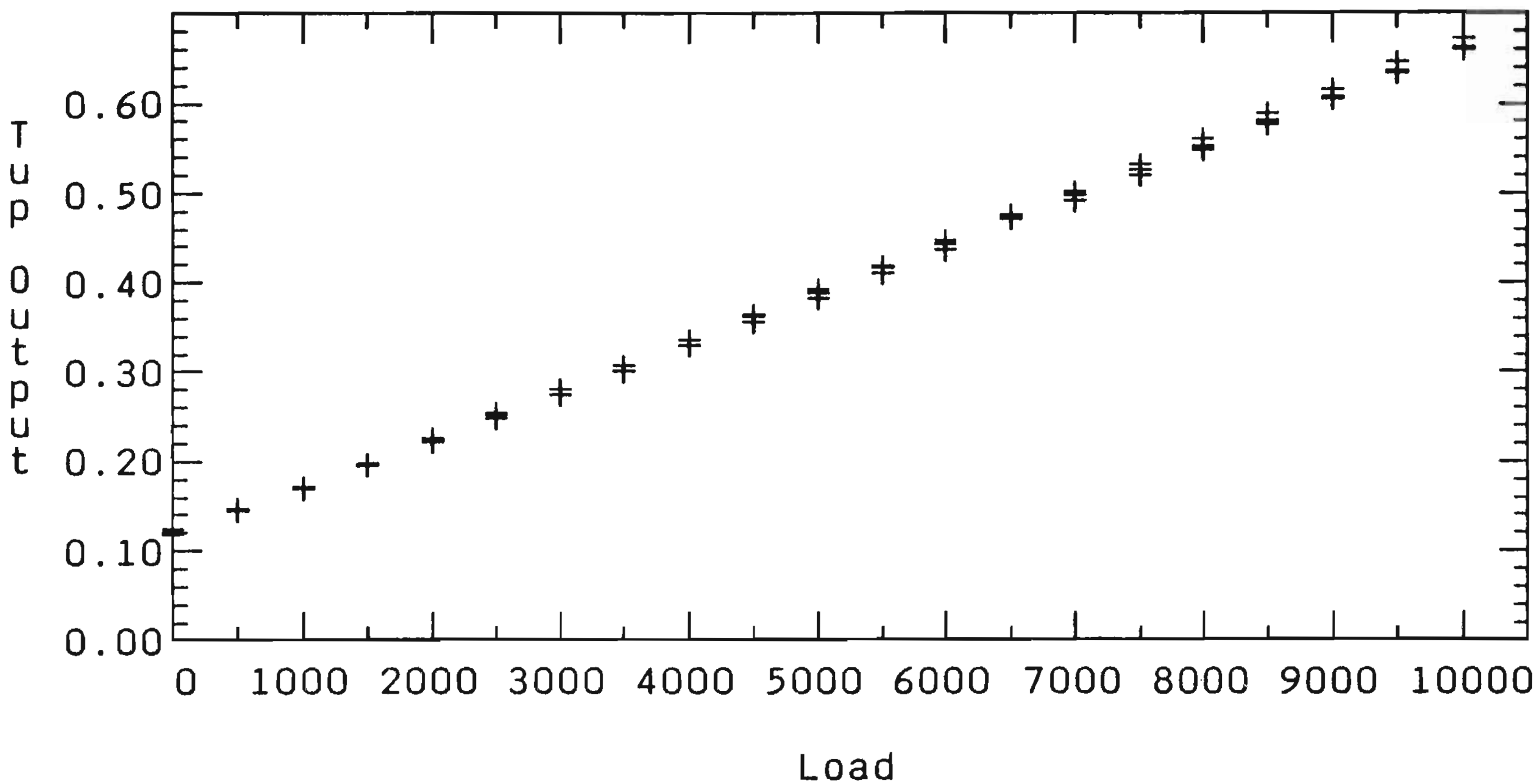


Figure 3-5. Recalibration of Ireland Associates Drop Tower Charpy Tup

Table 3-8(A)(1)
Statistical Analysis of Data in Table 3-7
(Recalibration of Ireland Associates Drop Tower Charpy Tup Against Load Rings)

Calibration Curve:

$$\hat{Y} = b0 + b1 * X$$

where:

$\hat{Y}$ ≡ Output of Ireland Associates drop tower Charpy tup in volts

b0, b1 ≡ Calibration coefficients

X ≡ Load in pounds

Specific Calibration Curve:

$$\hat{Y} = - .57962e-02 + .54942e-04\ X$$

Source	Sum of Squares	Degrees of Freedom	Mean Square
Due to Regression	2.0260	1	2.0260
Pure Error	0.0009	46	0.0000
Lack of Fit	0.0004	21	0.0000
About Regression	0.0014	67	0.0000
Total	2.0273	68	0.0298
Due to this Order	2.0260	1	2.0260
F Ratio for Improvement Due to Order 1:		0.10014e+06	
Degrees of Freedom	1 & 107		

Coefficient	b0	b1
Value	-.57962e-02	0.54942e-04
Standard Error	0.80771e-03	0.17362e-06
Two Sided T Ratio	0.71761e+01	0.31645e+03
Degrees of Freedom	67	

Table 3-8(A)(2)
Statistical Analysis of Data in Table 3-7
(Recalibration of Ireland Associates Drop Tower Charpy Tup Against Load Rings)

Calibration Curve:

$$\hat{Y} = b0 + b1 * X + b2 * X^2$$

where: $\hat{Y}$ ≡ Output of Ireland Associates drop tower Charpy tup in volts
b0, b1, b2 ≡ Calibration coefficients
X ≡ Load in pounds

Specific Calibration Curve:

$$\hat{Y} = - .37143e-02 + .53465e-04\ X + .15275e-09\ X^2$$

Source	Sum of Squares	Degrees of Freedom	Mean Square
Due to Regression	2.0261	2	1.0130
Pure Error	0.0009	46	0.0000
Lack of Fit	0.0003	20	0.0000
About Regression	0.0012	66	0.0000
Total	2.0273	68	0.0298
Due to this Order	0.0001	1	0.0001

F Ratio for Improvement Due to Order 2: 0.60176e+01
Degrees of Freedom 1 & 66

Coefficient	b0	b1	b2
Value	-.37143e-02	0.53465e-04	0.15275e-09
Standard Error	0.11520e-02	0.62514e-06	0.62267e-10
Two Sided T Ratio	0.32241e+01	0.85524e+02	0.24531e+01
Degrees of Freedom	66		

Table 3-8(A)(3)
Statistical Analysis of Data in Table 3-7
(Recalibration of Ireland Associates Drop Tower Charpy Tup Against Load Rings)

Calibration Curve:

$$\hat{Y} = b0 + b1 * X + b2 * X^2 + b3 * X^3$$

where: $\hat{Y}$ ≡ Output of Ireland Associates drop tower Charpy tup in volts
b0, b1, b2, b3 ≡ Calibration coefficients
X ≡ Load in pounds

Specific Calibration Curve:

$$\hat{Y} = -.21935e\text{-}02 + .51020e\text{-}04\ X + .81176e\text{-}09\ X^2 - .45088e\text{-}13\ X^3$$

Source	Sum of Squares	Degrees of Freedom	Mean Square
Due to Regression	2.0261	3	0.6754
Pure Error	0.0009	46	0.0000
Lack of Fit	0.0002	19	0.0000
About Regression	0.0012	65	0.0000
Total	2.0273	68	0.0298
Due to this Order	0.0001	1	0.0001
F Ratio for Improvement Due to Order 3:		0.39676e+01	
Degrees of Freedom	1 & 65		

Coefficient	b0	b1	b2
Value	-.21935e-02	0.51020e-04	0.81176e-09
Standard Error	0.13612e-02	0.13711e-05	0.33641e-09
Two Sided T Ratio	0.16114e+01	0.37211e+02	0.24130e+01
Degrees of Freedom	65		

Coefficient	b3
Value	-.45088e-13
Standard Error	0.22636e-13
Two Sided T Ratio	0.19919e+01
Degrees of Freedom	65

The linear curve fit from Table 3-8(A)(1) is recapitulated in Table 3-8(B).

Table 3-8(B)
Calibration Curve for the Ireland Associates Drop Tower Charpy Tup
Due to Load Ring Calibration

Calibration Curve:

$\hat{Y}$ = Zerovolts + Loadfac1 * X

where: $\hat{Y}$ ≡ Output of Ireland Associates drop tower Charpy tup in volts
Zerovolts, Loadfac1 ≡ Calibration coefficients
X ≡ Load in pounds

Specific Calibration Curve:

$\hat{Y}$ = - .57962e-02 + .54942e-04 X

Coefficient	b0	b1
Value	-.57962e-02	0.54942e-04
Standard Error	0.80771e-03	0.17362e-06
Two Sided T Ratio	0.71761e+01	0.31645e+03
Degrees of Freedom	67	

3.1.2 Effective Mass

3.1.2.1 Effective Mass of the Pendulum

For seven runs of 100 pendulum swings each (maximum angle less than 15°) the total time is shown in Table 3-9:

Table 3-9
Summary of Pendulum Swing Results

190.07 sec	
190.09 sec	
189.84 sec	
190.10 sec	
190.00 sec	
190.09 sec	
189.97 sec	
Mean:	190.02 sec
Standard Deviation:	0.094763 sec

Therefore, each individual swing had a period of 1.9002 sec with a standard deviation of 0.0009 sec.

Tinius Olsen indicates that the time required for 50 swings each less than 15° of the pendulum is 95 sec or, in other words, 100 swings can be accomplished in 190.0 sec. The OGC figure was used in the calculations for this paper since it is in nearly exact agreement with the Tinius Olsen figure.

The measured strike radius of the pendulum was 35.295". Tinius Olsen indicates that the strike radius is 35.437". The OGC figure was presumed to be accurate as the tup had been replaced since manufacture.

The measured pendulum weight was 67.531 lb. The estimated weight of the pendulum shaft (which rotates with the pendulum) was 3.18 lb. Summing the two components of the rotating mass gives 70.71 lb. This figure is in fairly good agreement with Tinius Olsen's estimate of 73.0 lb. Since the OGC pendulum weight is a direct measurement, the OGC total weight is considered more accurate than the Tinius Olsen estimate. However, because of the type of scale used, the OGC figure is accurate only to ± ½ lb.

The pendulum, cap, and screws without the shaft were balanced on an angle iron. The measured center of gravity radius was 2.625 feet or 31.50 inches. It follows that the calculated total pendulum assembly center of gravity was 2.507 feet or 30.08 inches. The calculated M_{eff} was 60.34 lb.

3.1.2.2 Effective Mass of the Drop Tower Crosshead

The measured weight of the drop tower crosshead was 179.62 lb ± 0.5 lb.

3.1.3 Initial Velocities

3.1.3.1 Initial Velocity of the Pendulum

The drop height of the pendulum was measured as 52.63". Tinius Olsen provided drop height dimensions of 53.16". Because of the difficulty of obtaining an accurate empirically derived figure, the Tinius Olsen data was used in calculating the initial velocity of the pendulum.

3.1.3.2 Initial Velocity of the Drop Tower Crosshead

3.1.3.2.1 Strobe Photos

Figures 3-6 and 3-7 are the stroboscopic photographs of the drop tower crosshead. The light horizontal lines are the strobe images of the tape line on the crosshead. The data taken from the stroboscopic photographs are shown in Tables 3-10(A) and 3-10(B).

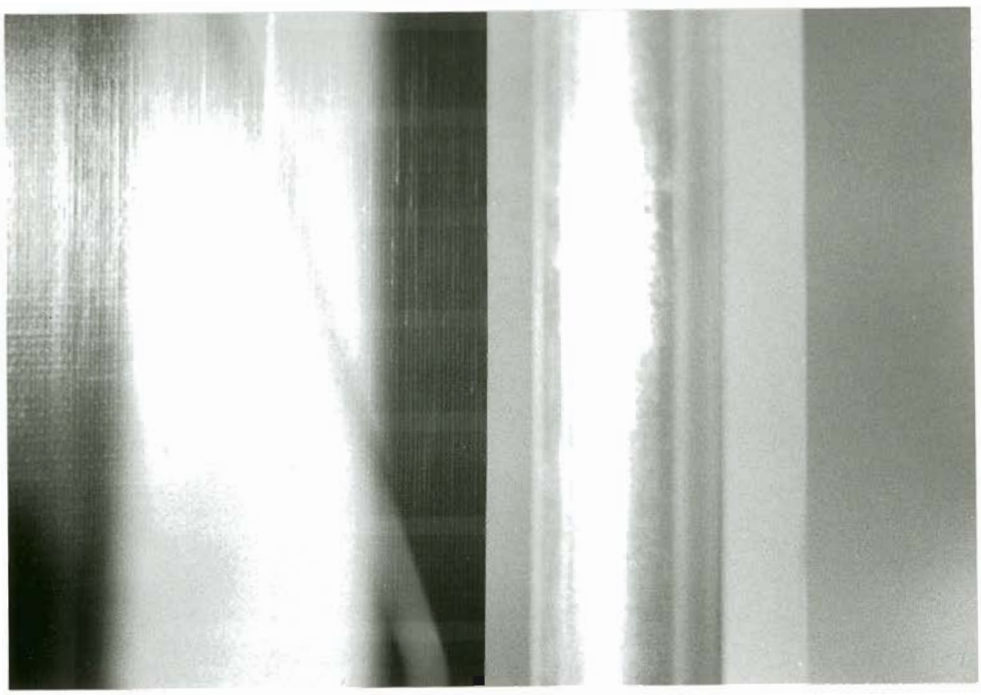


Figure 3-6. Strobe Photo #5

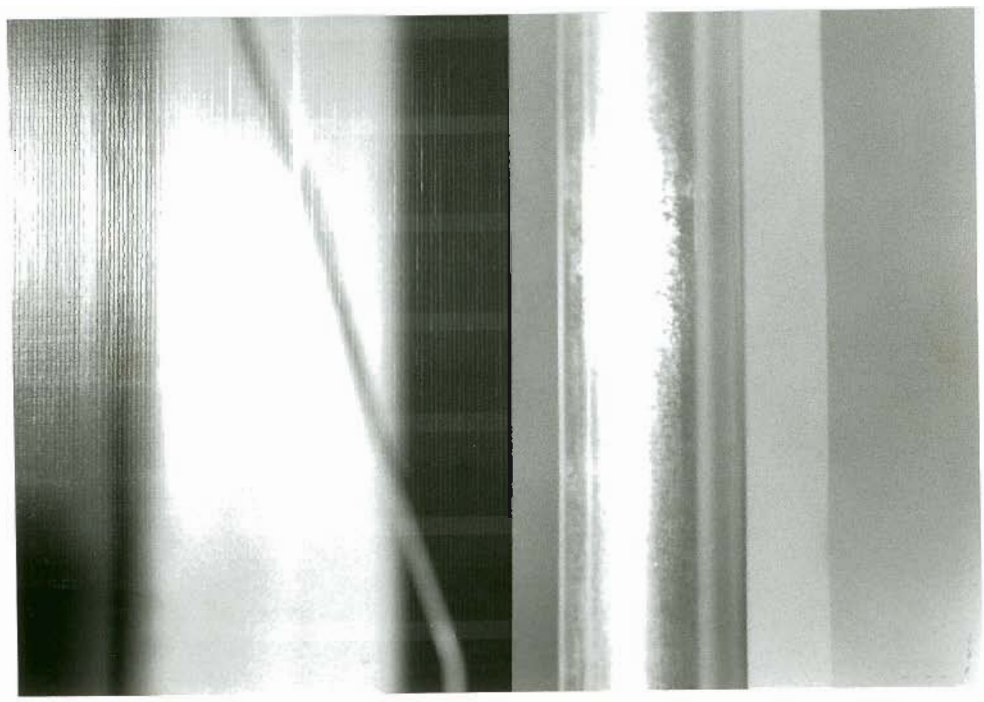


Figure 3-7. Strobe Photo #6

TABLE 3-10(A)
Coordinates of Crosshead in Strobe Photo #5

Time [sec]	Position [ft]
0.0000	0.492
0.0049	0.414
0.0099	0.336
0.0148	0.254
0.0198	0.174
0.0247	0.093
0.0296	0.009

TABLE 3-10(B)
Coordinates of Crosshead in Strobe Photo #6

Time [sec]	Position [ft]
0.0000	0.417
0.0049	0.338
0.0099	0.258
0.0148	0.179
0.0198	0.099
0.0247	0.018

Statistical analysis of Tables 3-10(A) and 3-10(B) by use of the PANOVA subroutine is shown in Tables 3-11(A) and 3-11(B).

Table 3-11(A)(1)
Statistical Analysis of Data in Table 3-10A
(Coordinates of Crosshead in Strobe Photo #5)

Calibration Curve:

$$\hat{Y} = b0 + b1 * X$$

where: $\hat{Y}$ ≡ Crosshead position in feet
b0, b1 ≡ Calibration coefficients
X ≡ Time in seconds

Specific Calibration Curve:

$$\hat{Y} = + .49437e+00 - .16272e+02\ X$$

Source	Sum of Squares	Degrees of Freedom	Mean Square
Due to Regression	0.1809	1	0.1809
About Regression	0.0000	5	0.0000
Total	0.1810	6	0.0302
Due to this Order	0.1809	1	0.1809
Degrees of Freedom	1 & 5		

Coefficient	b0	b1
Value	0.49437e+00	-.16272e+02
Standard Error	0.12468e-02	0.84125e-01
Two Sided T Ratio	0.39651e+03	0.19342e+03
Degrees of Freedom	5	

Table 3-11(A)(2)
Statistical Analysis of Data in Table 3-10A
(Coordinates of Crosshead in Strobe Photo #5)
(continued)

Calibration Curve:

$$\hat{Y} = b0 + b1 * X + b2 * X^2$$

where: $\hat{Y}$ ≡ Crosshead position in feet
b0, b1, b2 ≡ Calibration coefficients
X ≡ Time in Seconds

Specific Calibration Curve:

$$\hat{Y} = + .49185e{+}00 - .15657e{+}02 X - .20730e{+}02 X^2$$

Source	Sum of Squares	Degrees of Freedom	Mean Square
Due to Regression	0.1810	2	0.0905
About Regression	0.0000	4	0.0000
Total	0.1810	6	0.0302
Due to this Order	0.0000	1	0.0000
F Ratio for Improvement Due to Order 2:		0.32111e+02	
Degrees of Freedom	1 & 4		

Coefficient	b0	b1	b2
Value	0.49185e+00	-.15657e+02	-.20730e+02
Standard Error	0.64385e-03	0.11287e+00	0.36583e+01
Two Sided T Ratio	0.76391e+03	0.13872e+03	0.56667e+01
Degrees of Freedom	4		

Table 3-11(A)(3)
Statistical Analysis of Data in Table 3-10A
(Coordinates of Crosshead in Strobe Photo #5)
(continued)

Calibration Curve:

$$\hat{Y} = b0 + b1 * X + b2 * X^2 + b3 * X^3$$

where:		
$\hat{Y}$	≡	Crosshead position in feet
b0, b1, b2, b3	≡	Calibration coefficients
X	≡	Time in seconds

Specific Calibration Curve:

$$\hat{Y} = + .49185e{+}00 - .15657e{+}02\ X - .20730e{+}02\ X^2 + .39501e{-}11\ X^3$$

Source	Sum of Squares	Degrees of Freedom	Mean Square
Due to Regression	0.1810	3	0.0603
About Regression	0.0000	3	0.0000
Total	0.1810	6	0.0302

Coefficient	b0	b1	b2
Value	0.49185e+00	-.15657e+02	-.20730e+02
Standard Error	0.83757e-03	0.29108e+00	0.24082e+02
Two Sided T Ratio	0.58723e+03	0.53789e+02	0.86083e+00
Degrees of Freedom	3		

Coefficient	b3
Value	0.39501e-11
Standard Error	0.53322e+03
Two Sided T Ratio	0.74080e-14
Degrees of Freedom	3

Table 3-11(B)(1)
Statistical Analysis of Data in Table 3-10B
(Coordinates of Crosshead in Strobe Photo #6)

Calibration Curve:

$$\hat{Y} = b0 + b1 * X$$

where: $\hat{Y}$ ≡ Crosshead Position in feet
b0, b1 ≡ Calibration coefficients
X ≡ Time in seconds

Specific Calibration Curve:

$$\hat{Y} = + .41714e+00 - .16131e+02\ X$$

Source	Sum of Squares	Degrees of Freedom	Mean Square
Due to Regression	0.1111	1	0.1111
About Regression	0.0000	4	0.0000
Total	0.1111	5	0.0222
Due to this Order	0.1111	1	0.1111
Degrees of Freedom	1 & 4		

Coefficient	b0	b1
Value	0.41714e+00	-.16131e+02
Standard Error	0.56155e-03	0.45467e-01
Two Sided T Ratio	0.74285e+03	0.35478e+03
Degrees of Freedom	4	

Table 3-11(B)(2)
Statistical Analysis of Data in Table 3-10B
(Coordinates of Crosshead in Strobe Photo #6)
(continued)

Calibration Curve:

$$\hat{Y} = b0 + b1 * X + b2 * X^2$$

where:

$\hat{Y}$	≡	Crosshead position in feet
b0, b1, b2	≡	Calibration coefficients
X	≡	Time in seconds

Specific Calibration Curve:

$$\hat{Y} = + .41640e{+}00 - .15905e{+}02\ X - .91456e{+}01\ X^2$$

Source	Sum of Squares	Degrees of Freedom	Mean Square
Due to Regression	0.1111	2	0.0556
About Regression	0.0000	3	0.0000
Total	0.1111	5	0.0222
Due to this Order	0.0000	1	0.0000
F Ratio for Improvement Due to Order 2:		0.33383e+01	
Degrees of Freedom	1 & 3		

Coefficient	b0	b1	b2
Value	0.41640e+00	-.15905e+02	-.91456e+01
Standard Error	0.60402e-03	0.12881e+00	0.50056e+01
Two Sided T Ratio	0.68938e+03	0.12347e+03	0.18271e+01
Degrees of Freedom	3		

Table 3-11(B)(3)
Statistical Analysis of Data in Table 3-10B
(Coordinates of Crosshead in Strobe Photo #6)
(continued)

Calibration Curve:

$$\hat{Y} = b0 + b1 * X + b2 * X^2 + b3 * X^3$$

where: $\hat{Y}$ ≡ Crosshead position in feet
b0, b1, b2, b3 ≡ Calibration coefficients
X ≡ Time in seconds

Specific Calibration Curve:

$$\hat{Y} = + .41679e{+}00 - .16269e{+}02\ X + .31163e{+}02\ X^2 - .10879e{+}04\ X^3$$

Source	Sum of Squares	Degrees of Freedom	Mean Square
Due to Regression	0.1111	3	0.0370
About Regression	0.0000	2	0.0000
Total	0.1111	5	0.0222
Due to this Order	0.0000	1	0.0000
F Ratio for Improvement Due to Order 3:		0.40059e+01	
Degrees of Freedom	1 & 2		

Coefficient	b0	b1	b2
Value	0.41679e+00	-.16269e+02	0.31163e+02
Standard Error	0.47000e-03	0.20327e+00	0.20448e+02
Two Sided T Ratio	0.88680e+03	0.80034e+02	0.15240e+01
Degrees of Freedom	2		

Coefficient	b3
Value	-.10879e+04
Standard Error	0.54354e+03
Two Sided T Ratio	0.20015e+01
Degrees of Freedom	2

A different procedure using linear algebra to analyze the data in Tables 3-10(A) and 3-10(B) was used to generate Tables 3-12(A) and 3-12(B).

Table 3-12(A)
Estimate of Crosshead Velocity and Acceleration
Based on Data in Table 3-10(A)
(Coordinates of Crosshead in Strobe Photo #5)

	Vo [ft/sec]		Ao [ft/sec/sec]
	-15.51861111		-136.57502315
	-15.68729167		- 68.28751157
	-15.72946181		- 51.21563368
	-15.72102778		- 54.63000926
	-16.19333333		0.00000000
	-16.15116319		- 8.53593895
	-16.05838889		- 27.31500463
	-16.06682292		- 17.07187789
	-15.88970833		- 40.97250694
	-15.59451736		- 64.87313600
V_{avg}:	-15.86103264	A_{avg}:	- 46.94766235

Table 3-12(B)
Estimate of Crosshead Velocity and Acceleration
Based on Data in Table 3-10(B)
(Coordinates of Crosshead in Strobe Photo #6)

	Vo [ft/sec]		Ao [ft/sec/sec]
	-15.94031250		- 34.14375579
	-16.02465278		- 0.00000000
	-16.01059606		- 5.69062596
	-15.99091667		- 13.65750231
	-16.02465276		- 0.00000611
	-16.27767361		34.14375579
	-16.15116319		8.53593895
	-16.07525694		- 6.82875116
	-16.10899299		- 0.00001381
	-15.89814236		- 17.07187789
	-15.82223611		- 27.31500463
	-16.02465268		- 0.00001373
	-15.69572569		- 37.55813137
	************		*************
	-16.15960695		0.00078757
V_{avg}:	-16.01461295	A_{avg}:	- 7.11322860

******** Calculation was impossible for this data set.

3.1.3.2.2 Drop Tower Crosshead Drop Height

The drop height of the drop tower crosshead was measured and found to be 52.125".

3.1.4 Amplifier Rise Time

Figure 3-8 shows how the amplifier used to monitor tup output responded to a square wave. Amplifier rise time can be calculated from this graph.

3.1.5 Summary of Calibration Information

A summary of the calibration data is shown in Table 3-13. Since these calibration data are intermediate results used in further calculations, they are expressed with up to 8 significant figures to avoid round-off errors.

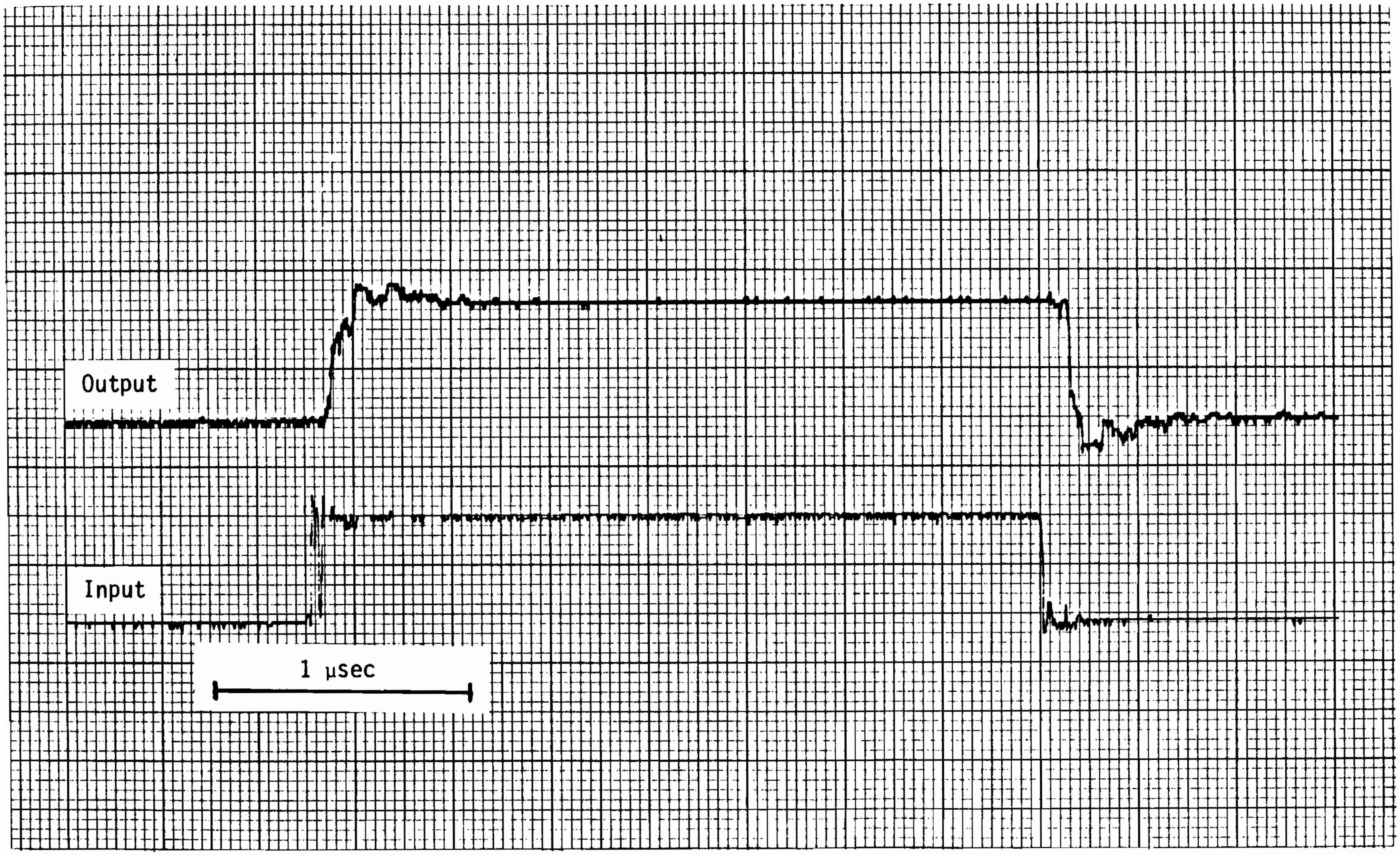


Figure 3-8. Amplifier Square Wave Rise Time Response

Table 3-13
Summary of Calibration Data

For the Drop Tower

Calibration Curve:

$$\hat{Y} = \text{Zerovolts} + \text{Loadfac1} * X$$

where:

$\hat{Y}$	≡	Output of Ireland Associates drop tower Charpy tup in volts
Zerovolts	≡	Tup output corresponding to zero load (determined separately from stored data for each test)
Loadfac1	≡	Calibration coefficient
X	≡	Load in pounds

V_o = 16.717562 ft/sec
M_{eff} = 179.62 lb/1 gravity

$$\hat{Y} = \text{Zerovolts} + (0.54942e\text{-}04) * (\text{Load in pounds})$$

For the Pendulum

Calibration Curve:

$$\hat{Y} = \text{Zerovolts} + \text{Loadfac1} * X$$

where:

$\hat{Y}$	≡	Output of Effects Technology pendulum Charpy tup in volts
Zerovolts	≡	Tup output corresponding to zero load (determined separately from stored data for each test)
Loadfac1	≡	Calibration coefficient
X	≡	Load in pounds

V_o = 16.882123 ft/sec
M_{eff} = 60.34 lb/1 gravity

$$\hat{Y} = \text{Zerovolts} + (0.72656e\text{-}04) * (\text{Load in pounds}) - (0.43673e\text{-}09) * (\text{Load in pounds})^2$$

3.2 Specimens

3.2.1 Specimen Chemistry

The results of spark spectrography of the plates from which the Charpy bars were cut are shown in Table 3-14. The designation, plate of origin, hardness, testing type, and dimensions of the Charpy specimens actually considered in this study are displayed in the following table (Table 3-15).

Table 3-14
Chemical Composition of Specimens
(in percentages)

Element	Plate B	Plate C	Plate D	Plate F	Plate G
Carbon	0.380	0.383	0.385	0.382	0.380
Manganese	0.74	0.73	0.74	0.73	0.74
Silicon	0.34	0.34	0.34	0.34	0.34
Chromium	0.84	0.83	0.83	0.83	0.84
Nickel	1.83	1.82	1.84	1.82	1.83
Molybdenum	0.22	0.22	0.21	0.21	0.22
Copper	0.21	0.21	0.21	0.21	0.21
Sulfur	0.018	0.014	0.013	0.016	0.018
Phosphorous	0.005	0.005	0.005	0.004	0.005
Aluminum	0.020	0.020	0.021	0.021	0.020
Lead	0.005	0.004	0.004	0.005	0.005
Titanium	-	-	-	-	-
Vanadium	0.049	0.048	0.048	0.047	0.05
Boron	0.0009	0.0009	0.0010	0.0009	0.0009
Cobalt	0.03	0.025	0.03	0.03	0.03
Tungsten	0.01	0.013	0.01	0.01	0.01
Zirconium	0.001	-	-	0.001	0.001
Tin	0.007	0.006	0.006	0.005	0.007

3.2.2 Physical Description of Specimens

Table 3-15
Specimen Descriptions

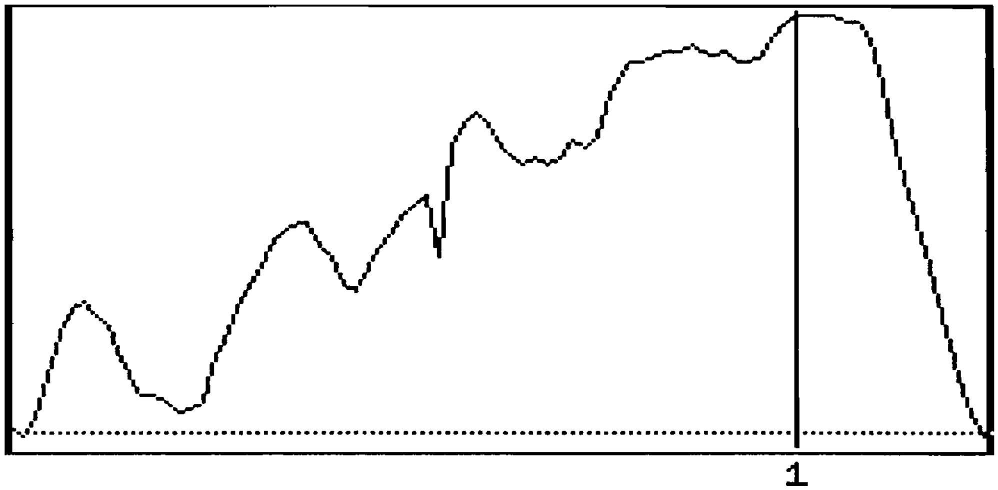
ID	Width [inch]	Depth [inch]	Length [inch]	Rockwell Hardness C Scale	Plate	'Dial Energy' [ft-lb]	Test Temp [°C]	Drop Pend
H1	0.3943	0.3940	2.195	27, 27, 27	C		19.5	Drop
H3	0.3941	0.3940	2.175	28, 28, 28	C		19.0	Drop
H4	0.3940	0.3940	2.174	26, 26, 26	D		19.0	Drop
H5	0.3938	0.3930	2.186	28, 28, 27	D		19.0	Drop
H6	0.3945	0.3935	2.177	28, 28, 28	C	55.5	19.0	Pend
H7	0.3944	0.3942	2.178	27, 27, 26	C	53.0	19.0	Pend
H8	0.3926	0.3937	2.172	29, 30, 29	D	49.5	19.0	Pend
H10	0.3946	0.3948	2.189	29, 29, 29	C	52.0	19.0	Pend
H11	0.3939	0.3944	2.165	27, 27, 27	D	54.2	19.0	Pend
S2	0.3943	0.3940	2.183	24, 23, 24	C		19.5	Drop
L2	0.3924	0.3943	2.212	54, 54, 53	C	12.5	19.0	Pend
L3	0.3945	0.3946	2.192	54, 54, 54	C	12.0	19.0	Pend
L4	0.3954	0.3926	2.196	54, 54, 54	D	12.5	19.0	Pend
L5	0.3929	0.3940	2.156	53, 53, 54	D	12.0	19.0	Pend
L6	0.3953	0.3942	2.181	54, 54, 54	D	12.5	19.0	Pend
L7	0.3953	0.3940	2.154	53, 53, 53	D		18.5	Drop
L8	0.3953	0.3943	2.179	54, 54, 54	D		18.5	Drop
L9	0.3939	0.3943	2.179	55, 54, 54	C		18.5	Drop
L10	0.3947	0.3946	2.204	54, 54, 54	C		18.5	Drop
L11	0.3948	0.3939	2.165	54, 54, 54	D		18.5	Drop
L12	0.3943	0.3938	2.181	54, 54, 54	C		18.5	Drop

3.3 Impact Tests

3.3.1 'Tup Energies' for All Specimens

Figures 3-9(A) through 3-9(U) are the ASTIR reports for all the specimens used in this study. Table 3-16 contains 'tup energy' results for all the specimens used in this study. For those specimens tested in the pendulum machine, pendulum 'dial energies' are listed as well. In this case, Astir was configured to perform second order integration, to use a second order tup calibration, to use the measured value of M_{eff} and to use an initial velocity V_0 consistent with $V_0=(2gh_0)^{1/2}$ where g is the local acceleration of gravity and h_0 is the drop height.

DESIGNATION: L2
TEST METHOD: INSTRUMENTED PENDULUM IMPACT



USEFUL POINTS: 81
DATA AVERAGED OVER 1 POINT

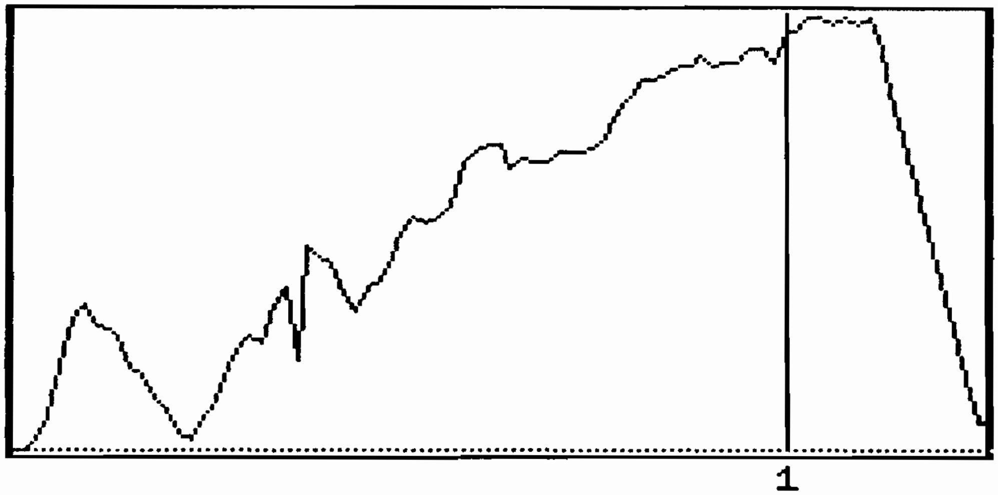
	FEATURE	LOAD [Lb]	TIME [mSec]	DEFLECTION [In]	ENERGY [Ft-Lb]
1	GENERAL YIELD	7446.	0.148	0.0299	9.7
4	END OF EVENT	-----	0.182	0.0367	12.5

TOTAL ENERGY

DIAL: ______ [Ft-Lb]
FLAG: ______ [Ft-Lb]
TUP: 12.5 [Ft-Lb]

Figure 3-9(A). L2 ASTIR Report

DESIGNATION: L3
TEST METHOD: INSTRUMENTED PENDULUM IMPACT



USEFUL POINTS: 81
DATA AVERAGED OVER 1 POINT

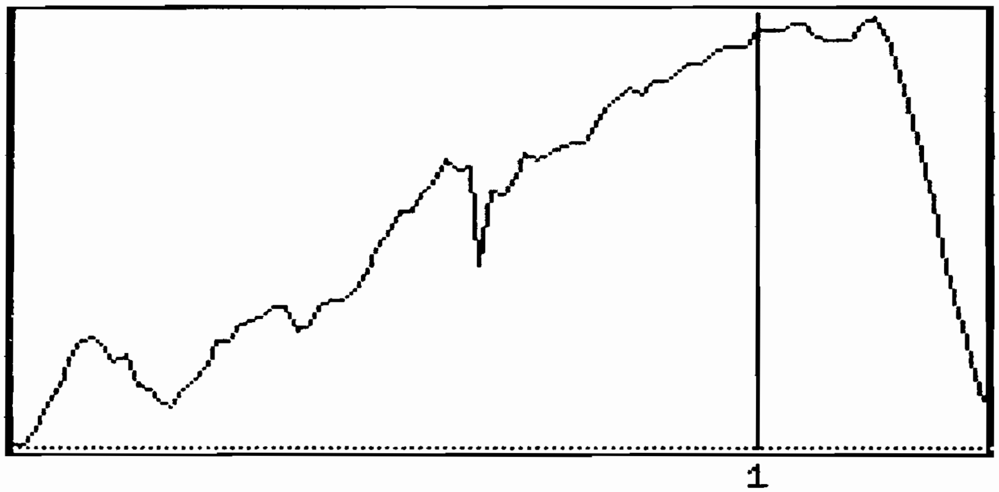
FEATURE	LOAD [Lb]	TIME [mSec]	DEFLECTION [In]	ENERGY [Ft-Lb]
1 GENERAL YIELD	7111.	0.146	0.0294	8.8
4 END OF EVENT	-----	0.182	0.0367	12.1

TOTAL ENERGY

DIAL: ______ [Ft-Lb]
FLAG: ______ [Ft-Lb]
TUP: 12.1 [Ft-Lb]

Figure 3-9(B). L3 ASTIR Report

DESIGNATION: L4
TEST METHOD: INSTRUMENTED PENDULUM IMPACT



USEFUL POINTS: 84
DATA AVERAGED OVER 1 POINT

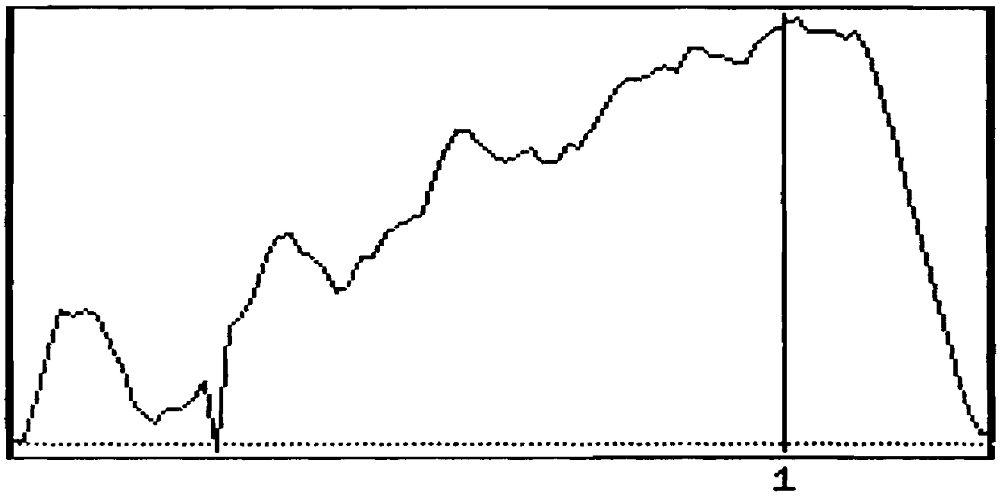
FEATURE	LOAD [Lb]	TIME [mSec]	DEFLECTION [In]	ENERGY [Ft-Lb]
1 GENERAL YIELD	7224.	0.146	0.0294	8.7
4 END OF EVENT	-----	0.189	0.0380	12.9

TOTAL ENERGY

DIAL: ______ [Ft-Lb]
FLAG: ______ [Ft-Lb]
TUP: 12.9 [Ft-Lb]

Figure 3-9(C). L4 ASTIR Report

DESIGNATION: L5
TEST METHOD: INSTRUMENTED PENDULUM IMPACT



USEFUL POINTS: 80
DATA AVERAGED OVER 1 POINT

FEATURE	LOAD [Lb]	TIME [mSec]	DEFLECTION [In]	ENERGY [Ft-Lb]
1 GENERAL YIELD	7248.	0.143	0.0289	9.3
4 END OF EVENT	-----	0.180	0.0362	12.3

TOTAL ENERGY

DIAL: ______ [Ft-Lb]
FLAG: ______ [Ft-Lb]
TUP: 12.3 [Ft-Lb]

Figure 3-9(D). L5 ASTIR Report

DESIGNATION: L6
TEST METHOD: INSTRUMENTED PENDULUM IMPACT

USEFUL POINTS: 90
DATA AVERAGED OVER 1 POINT

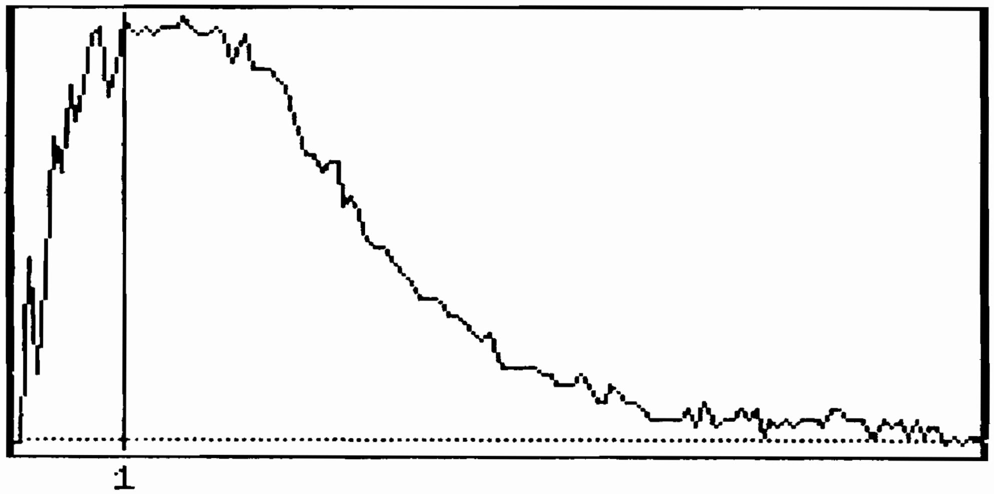
FEATURE	LOAD [Lb]	TIME [mSec]	DEFLECTION [In]	ENERGY [Ft-Lb]
1 GENERAL YIELD	7425.	0.155	0.0313	8.1
4 END OF EVENT	-----	0.203	0.0408	12.6

TOTAL ENERGY

DIAL: ______ [Ft-Lb]
FLAG: ______ [Ft-Lb]
TUP: 12.6 [Ft-Lb]

Figure 3-9(E). L6 ASTIR Report

DESIGNATION: H6
TEST METHOD: INSTRUMENTED PENDULUM IMPACT



USEFUL POINTS: 126
DATA AVERAGED OVER 1 POINT

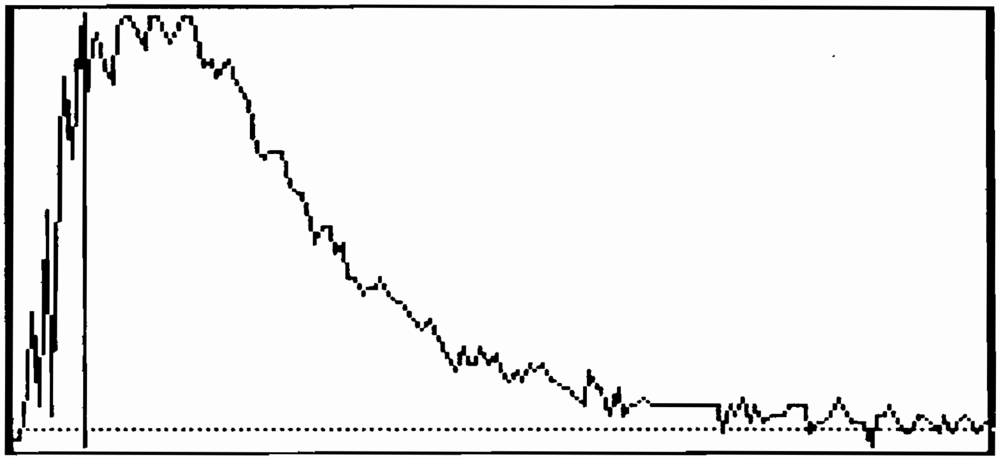
	FEATURE	LOAD [Lb]	TIME [mSec]	DEFLECTION [In]	ENERGY [Ft-Lb]
1	GENERAL YIELD	5032.	0.185	0.0373	10.7
4	END OF EVENT	-----	1.785	0.3335	53.3

TOTAL ENERGY

DIAL: ______ [Ft-Lb]
FLAG: ______ [Ft-Lb]
TUP: 53.3 [Ft-Lb]

Figure 3-9(F). H6 ASTIR Report

DESIGNATION: H7
TEST METHOD: INSTRUMENTED PENDULUM IMPACT



USEFUL POINTS: 146
DATA AVERAGED OVER 1 POINT

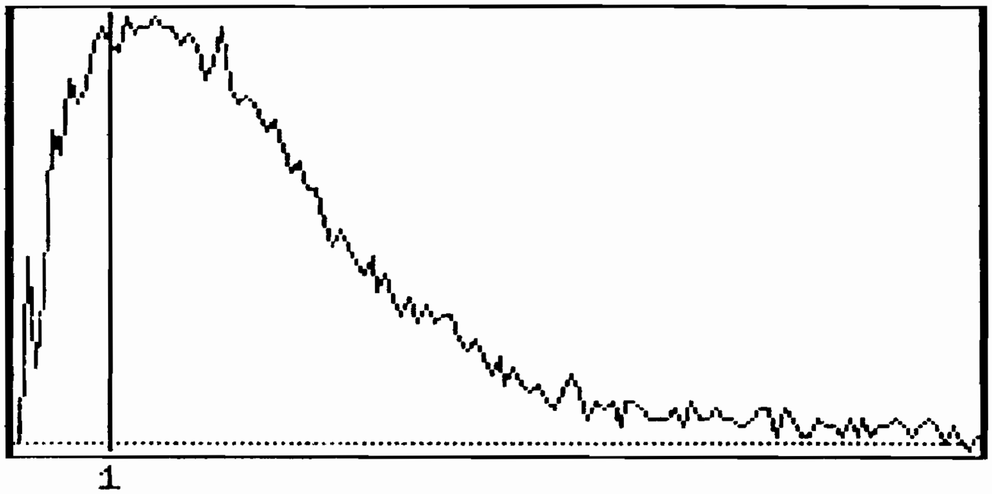
FEATURE	LOAD [Lb]	TIME [mSec]	DEFLECTION [In]	ENERGY [Ft-Lb]
1 GENERAL YIELD	3796.	0.142	0.0288	5.3
4 END OF EVENT	-----	2.071	0.3873	51.6

TOTAL ENERGY

DIAL: ______ [Ft-Lb]
FLAG: ______ [Ft-Lb]
TUP: 51.6 [Ft-Lb]

Figure 3-9(G). H7 ASTIR Report

DESIGNATION: H8
TEST METHOD: INSTRUMENTED PENDULUM IMPACT



USEFUL POINTS: 121
DATA AVERAGED OVER 1 POINT

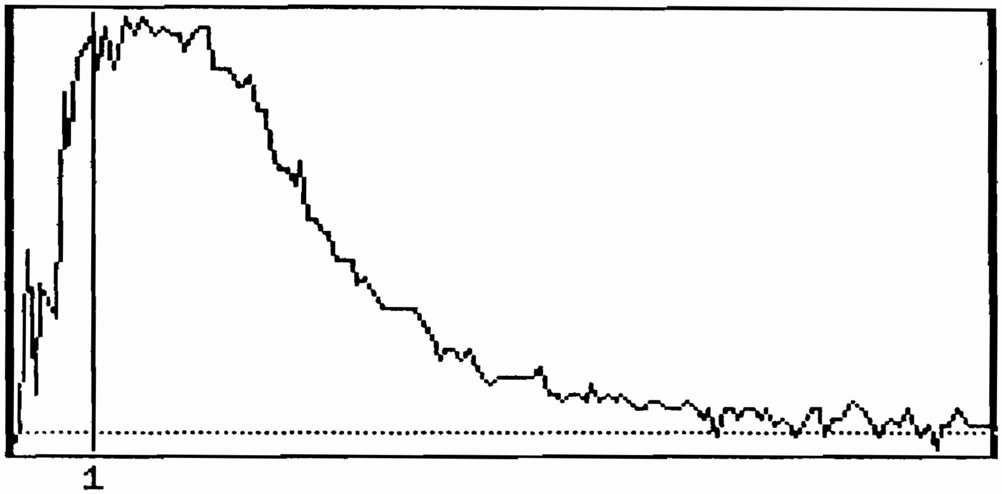
FEATURE	LOAD [Lb]	TIME [mSec]	DEFLECTION [In]	ENERGY [Ft-Lb]
1 GENERAL YIELD	4728.	0.157	0.0316	8.7
4 END OF EVENT	-----	1.714	0.3229	47.6

TOTAL ENERGY

DIAL: ______ [Ft-Lb]
FLAG: ______ [Ft-Lb]
TUP: 47.6 [Ft-Lb]

Figure 3-9(H). H8 ASTIR Report

DESIGNATION: H10
TEST METHOD: INSTRUMENTED PENDULUM IMPACT



USEFUL POINTS: 134
DATA AVERAGED OVER 1 POINT

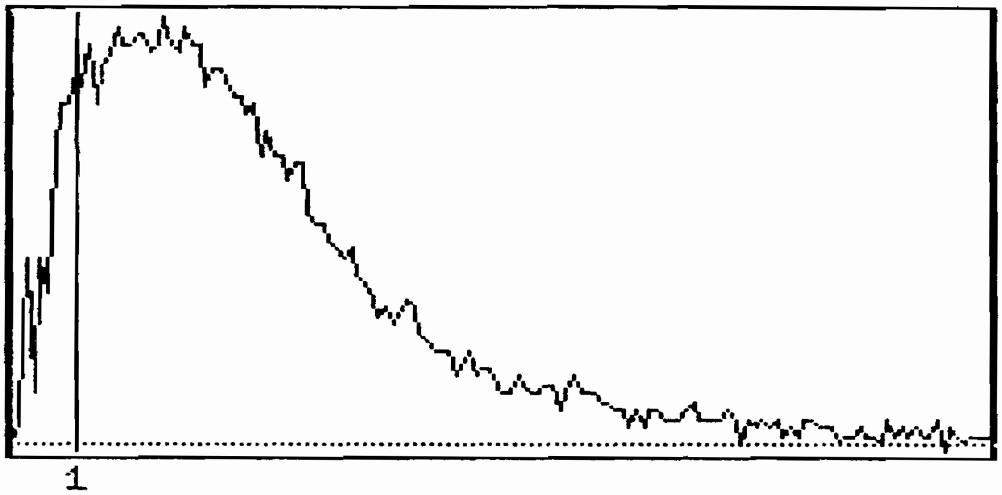
FEATURE	LOAD [Lb]	TIME [mSec]	DEFLECTION [In]	ENERGY [Ft-Lb]
1 GENERAL YIELD	4831.	0.142	0.0288	6.4
4 END OF EVENT	-----	1.899	0.3564	49.7

TOTAL ENERGY

DIAL: ______ [Ft-Lb]
FLAG: ______ [Ft-Lb]
TUP: 49.7 [Ft-Lb]

Figure 3-9(I). H10 ASTIR Report

DESIGNATION: H11
TEST METHOD: INSTRUMENTED PENDULUM IMPACT



USEFUL POINTS: 140
DATA AVERAGED OVER 1 POINT

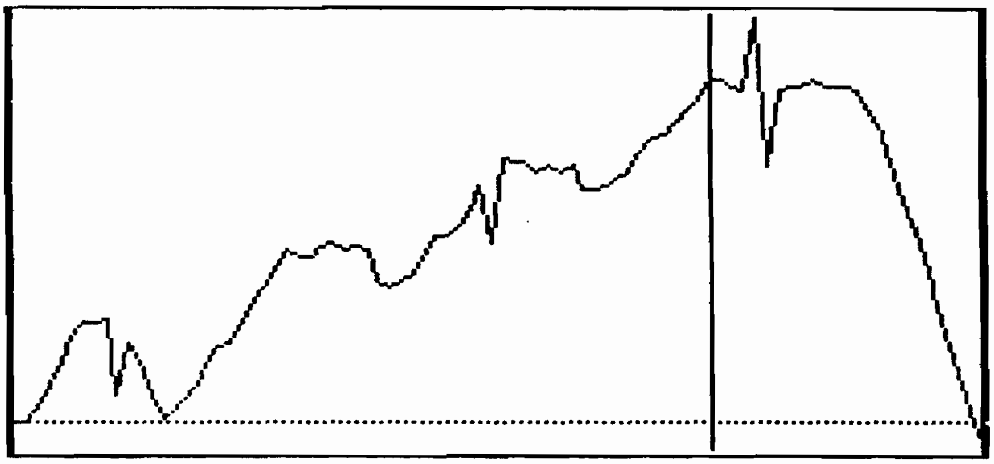
FEATURE	LOAD [Lb]	TIME [mSec]	DEFLECTION [In]	ENERGY [Ft-Lb]
1 GENERAL YIELD	4448.	0.114	0.0230	5.0
4 END OF EVENT	-----	1.985	0.3705	53.2

TOTAL ENERGY

DIAL: ______ [Ft-Lb]
FLAG: ______ [Ft-Lb]
TUP: 53.2 [Ft-Lb]

Figure 3-9(J). H11 ASTIR Report

DESIGNATION: L7
TEST METHOD: INSTRUMENTED DROP TOWER IMPACT



USEFUL POINTS: 84
DATA AVERAGED OVER 1 POINT

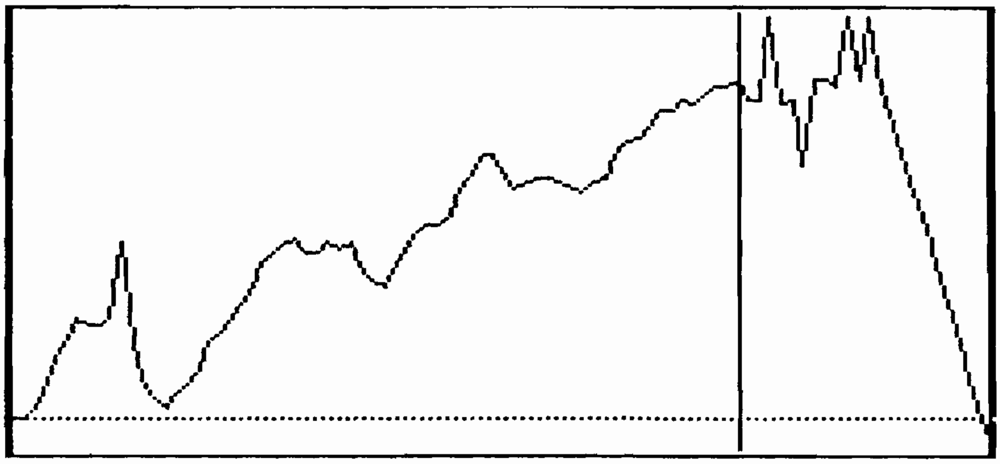
FEATURE	LOAD [Lb]	TIME [mSec]	DEFLECTION [In]	ENERGY [Ft-Lb]
1 GENERAL YIELD	9336.	0.137	0.0274	10.2
4 END OF EVENT	-----	0.189	0.0379	16.6

TOTAL ENERGY

FLAG: ______ [Ft-Lb]
TUP: 16.6 [Ft-Lb]

Figure 3-9(K). L7 ASTIR Report

DESIGNATION: L8
TEST METHOD: INSTRUMENTED DROP TOWER IMPACT



USEFUL POINTS: 88
DATA AVERAGED OVER 1 POINT

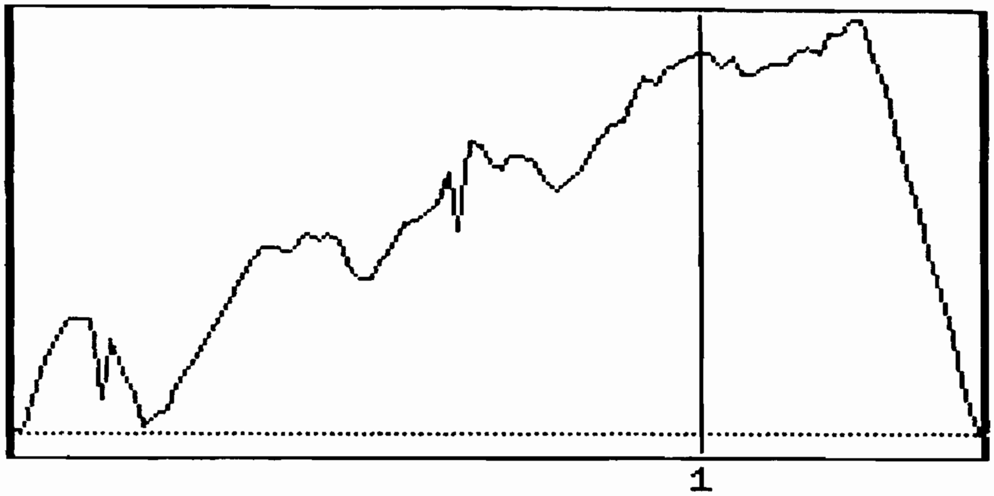
FEATURE	LOAD [Lb]	TIME [mSec]	DEFLECTION [In]	ENERGY [Ft-Lb]
1 GENERAL YIELD	9351.	0.148	0.0297	12.0
4 END OF EVENT	-----	0.198	0.0397	18.2

TOTAL ENERGY

FLAG: ______ [Ft-Lb]
TUP: 18.2 [Ft-Lb]

Figure 3-9(L). L8 ASTIR Report

DESIGNATION: L9
TEST METHOD: INSTRUMENTED DROP TOWER IMPACT



USEFUL POINTS: 88
DATA AVERAGED OVER 1 POINT

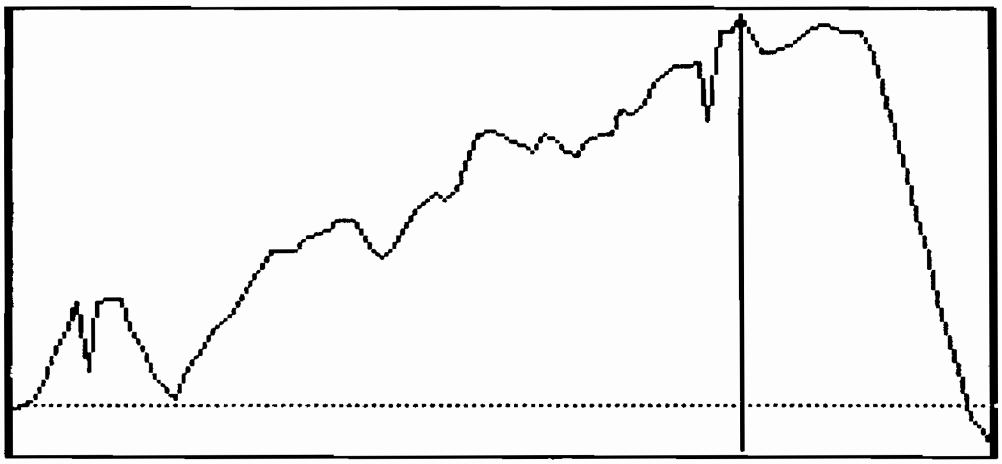
FEATURE	LOAD [Lb]	TIME [mSec]	DEFLECTION [In]	ENERGY [Ft-Lb]
1 GENERAL YIELD	9362.	0.141	0.0283	11.1
4 END OF EVENT	-----	0.198	0.0397	18.4

TOTAL ENERGY

FLAG: ______ [Ft-Lb]
TUP: 18.4 [Ft-Lb]

Figure 3-9(M). L9 ASTIR Report

DESIGNATION: L10
TEST METHOD: INSTRUMENTED DROP TOWER IMPACT



USEFUL POINTS: 89
DATA AVERAGED OVER 1 POINT

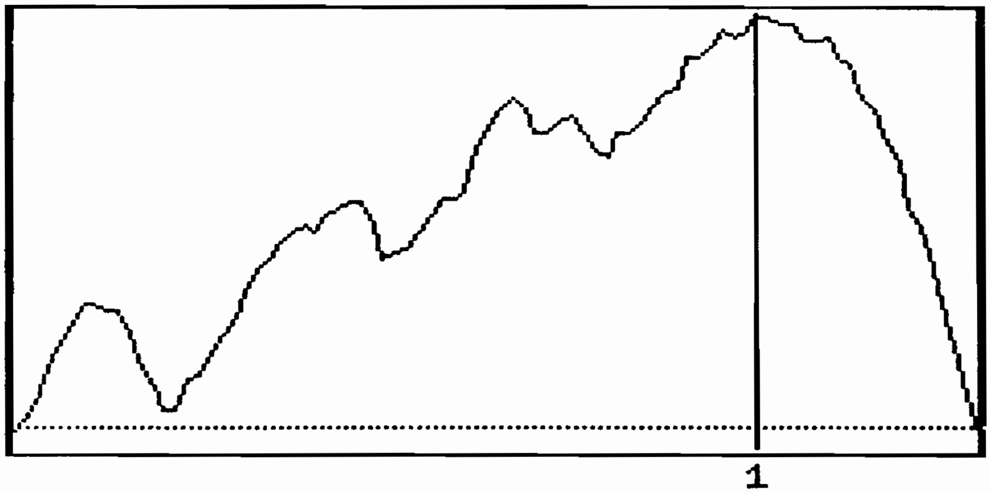
FEATURE	LOAD [Lb]	TIME [mSec]	DEFLECTION [In]	ENERGY [Ft-Lb]
1 GENERAL YIELD	9493.	0.150	0.0301	11.4
4 END OF EVENT	-----	0.201	0.0401	16.9

TOTAL ENERGY

FLAG: ______ [Ft-Lb]
TUP: 16.9 [Ft-Lb]

Figure 3-9(N). L10 ASTIR Report

DESIGNATION: L11
TEST METHOD: INSTRUMENTED DROP TOWER IMPACT



USEFUL POINTS: 82
DATA AVERAGED OVER 1 POINT

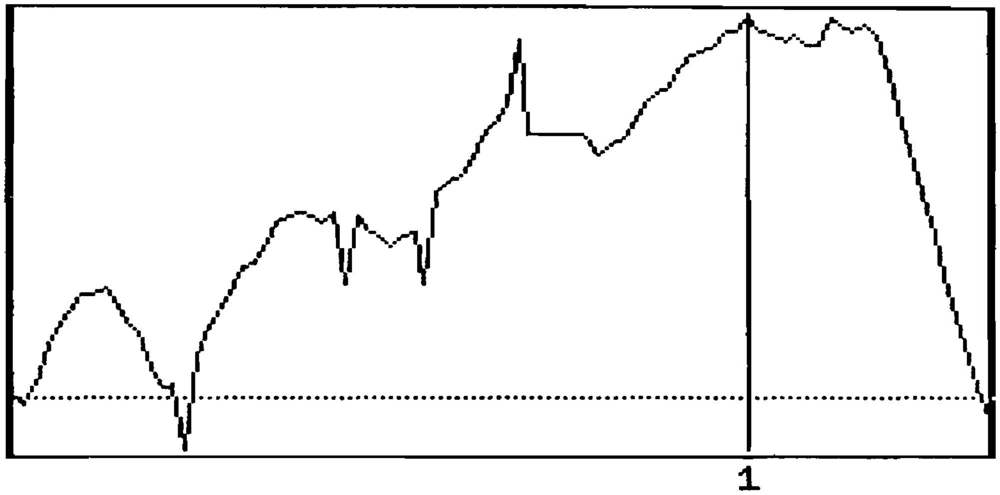
FEATURE	LOAD [Lb]	TIME [mSec]	DEFLECTION [In]	ENERGY [Ft-Lb]
1 GENERAL YIELD	9227.	0.143	0.0288	11.4
4 END OF EVENT	-----	0.185	0.0369	15.9

TOTAL ENERGY

FLAG: ______ [Ft-Lb]
TUP: 15.9 [Ft-Lb]

Figure 3-9(0). L11 ASTIR Report

DESIGNATION: L12
TEST METHOD: INSTRUMENTED DROP TOWER IMPACT



USEFUL POINTS: 84
DATA AVERAGED OVER 1 POINT

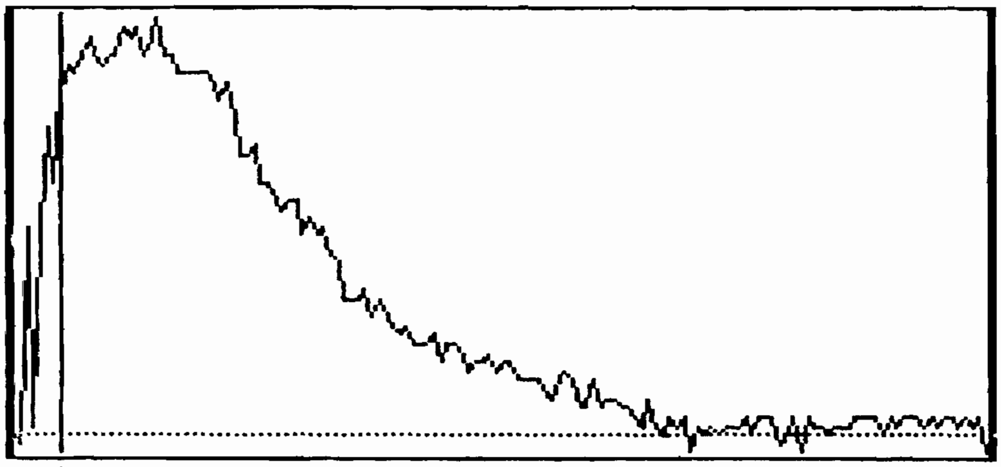
FEATURE	LOAD [Lb]	TIME [mSec]	DEFLECTION [In]	ENERGY [Ft-Lb]
1 GENERAL YIELD	9480.	0.143	0.0288	11.2
4 END OF EVENT	-----	0.189	0.0379	16.6

TOTAL ENERGY

FLAG: ______ [Ft-Lb]
TUP: 16.6 [Ft-Lb]

Figure 3-9(P). L12 ASTIR Report

DESIGNATION: H1
TEST METHOD: INSTRUMENTED DROP TOWER IMPACT



USEFUL POINTS: 141
DATA AVERAGED OVER 1 POINT

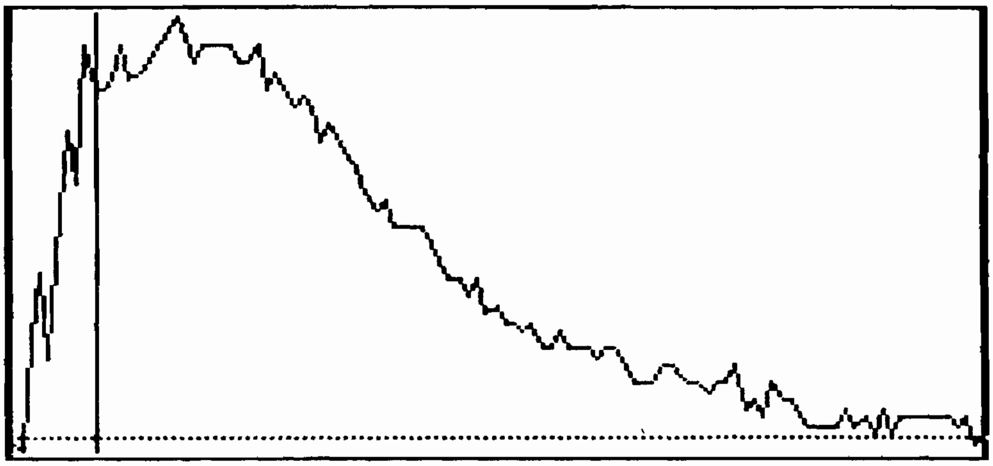
FEATURE	LOAD [Lb]	TIME [mSec]	DEFLECTION [In]	ENERGY [Ft-Lb]
1 GENERAL YIELD	4922.	0.085	0.0171	4.0
4 END OF EVENT	-----	1.999	0.3898	60.3

TOTAL ENERGY

FLAG: ______ [Ft-Lb]
TUP: 60.3 [Ft-Lb]

Figure 3-9(Q). H1 ASTIR Report

DESIGNATION: H3
TEST METHOD: INSTRUMENTED DROP TOWER IMPACT



USEFUL POINTS: 105
DATA AVERAGED OVER 1 POINT

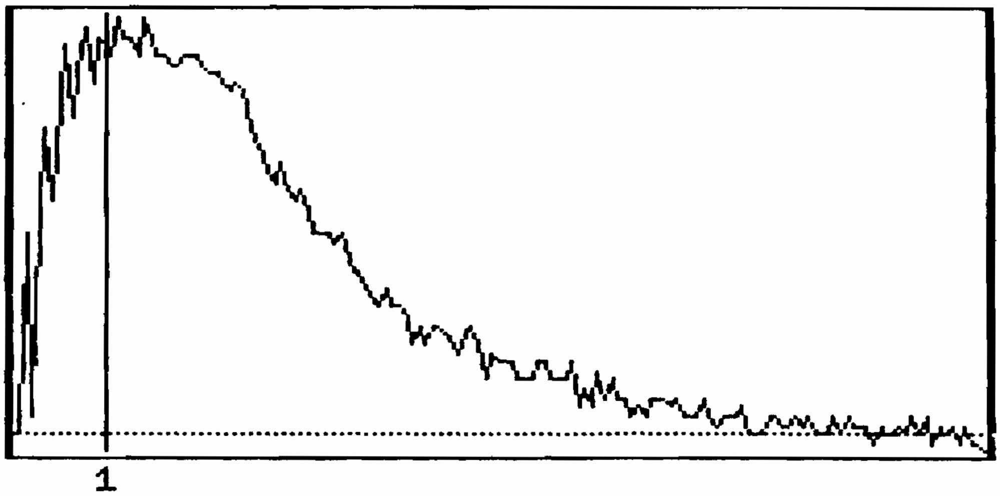
FEATURE	LOAD [Lb]	TIME [mSec]	DEFLECTION [In]	ENERGY [Ft-Lb]
1 GENERAL YIELD	5196.	0.114	0.0228	6.0
4 END OF EVENT	-----	1.485	0.2900	61.4

TOTAL ENERGY

FLAG: ______ [Ft-Lb]
TUP: 61.4 [Ft-Lb]

Figure 3-9(R). H3 ASTIR Report

DESIGNATION: H4
TEST METHOD: INSTRUMENTED DROP TOWER IMPACT



USEFUL POINTS: 148
DATA AVERAGED OVER 1 POINT

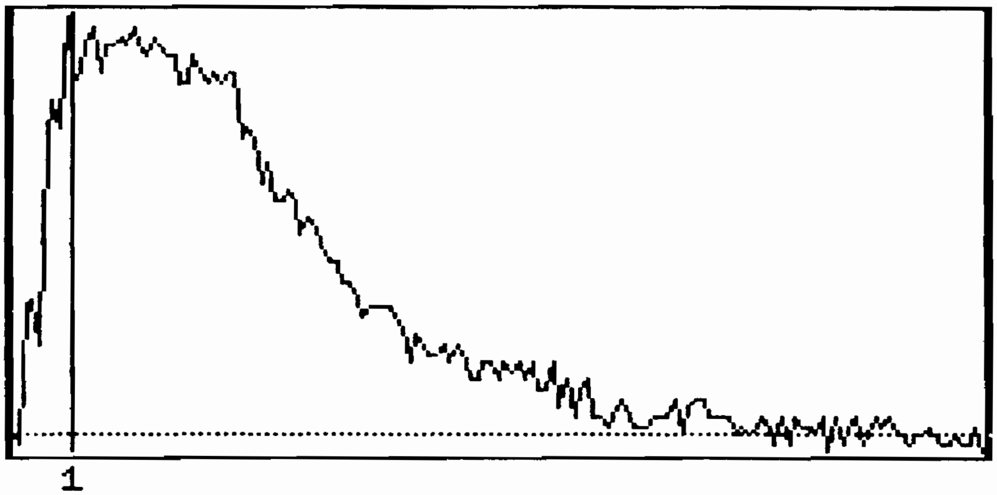
FEATURE	LOAD [Lb]	TIME [mSec]	DEFLECTION [In]	ENERGY [Ft-Lb]
1 GENERAL YIELD	5569.	0.185	0.0371	12.7
4 END OF EVENT	-----	2.099	0.4080	66.7

TOTAL ENERGY

FLAG: ______ [Ft-Lb]
TUP: 66.7 [Ft-Lb]

Figure 3-9(S). H4 ASTIR Report

DESIGNATION: H5
TEST METHOD: INSTRUMENTED DROP TOWER IMPACT



USEFUL POINTS: 144
DATA AVERAGED OVER 1 POINT

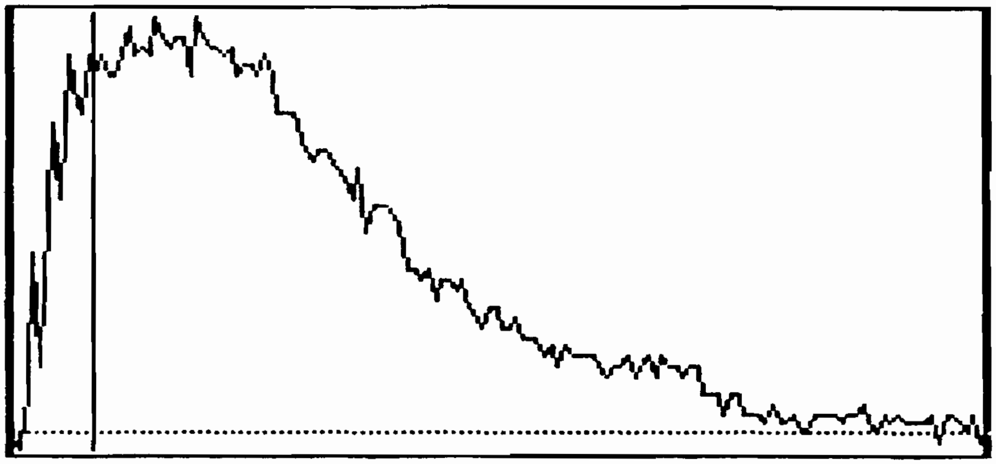
FEATURE	LOAD [Lb]	TIME [mSec]	DEFLECTION [In]	ENERGY [Ft-Lb]
1 GENERAL YIELD	5050.	0.114	0.0228	6.4
4 END OF EVENT	-----	2.042	0.3978	61.4

TOTAL ENERGY

FLAG: ______ [Ft-Lb]
TUP: 61.4 [Ft-Lb]

Figure 3-9(T). H5 ASTIR Report

DESIGNATION: S2
TEST METHOD: INSTRUMENTED DROP TOWER IMPACT



USEFUL POINTS: 141
DATA AVERAGED OVER 1 POINT

FEATURE	LOAD [Lb]	TIME [mSec]	DEFLECTION [In]	ENERGY [Ft-Lb]
1 GENERAL YIELD	4939.	0.157	0.0314	8.7
4 END OF EVENT	------	1.999	0.3881	72.9

TOTAL ENERGY

FLAG: ______ [Ft-Lb]
TUP: 72.9 [Ft-Lb]

Figure 3-9(U). S2 ASTIR Report

Table 3-16
Impact Energy Results for All Specimens in This Study

Specimen ID	Drop Tower Pendulum	'Dial Energy' [ft-lb]	'Tup Energy' [ft-lb]
H6	Pend	55.5	53.3
H7	Pend	53.0	51.6
H8	Pend	49.5	47.6
H10	Pend	52.0	49.7
H11	Pend	54.0	53.2
L2	Pend	12.5	12.5
L3	Pend	12.0	12.1
L4	Pend	12.5	12.9
L5	Pend	12.0	12.3
L6	Pend	12.5	12.6
H1	Drop	-	60.3
H3	Drop	-	61.4
H4	Drop	-	66.7
H5	Drop	-	61.4
S2	Drop	-	72.9
L7	Drop	-	16.6
L8	Drop	-	18.2
L9	Drop	-	18.4
L10	Drop	-	16.9
L11	Drop	-	15.9
L12	Drop	-	16.6

3.3.2 Effects of Integration Order

In Table 3-17 the results of performing the ASTIR calculations using zeroeth order, first order, and second order integration (all other factors being the same) are compared for the pendulum.

Table 3-17
Effect of Integration Order on 'Tup Energy'

Specimen ID	0th Order [ft-lb]	1st Order [ft-lb]	2nd Order [ft-lb]
H6	53.3	53.3	53.3
H7	51.6	51.6	51.6
H8	47.6	47.6	47.6
H10	49.6	49.6	49.7
H11	53.1	53.1	53.2
L2	12.5	12.5	12.5
L3	12.1	12.1	12.1
L4	13.0	12.9	12.9
L5	12.3	12.3	12.3
L6	12.6	12.5	12.6

3.3.3 Effects of Mass

In Tables 3-18(A) and 3-18(B), the results of using several different numbers for M_{eff} are compared with the results for M_{eff} as measured. Figures 3-10(A) and 3-10(B) show this comparison graphically.

3.3.3.1 Pendulum Mass

Table 3-18(A)
Assumed Effects of M_{eff} on Pendulum Test 'Tup Energy'

Assumed Weight of Pendulum [lb]	Specimen L2 [ft-lb]	Specimen H6 [ft-lb]
10	11.7	38.4
20	12.2	47.3
30	12.3	50.3
40	12.4	51.8
50	12.5	52.7
60	12.5	53.3
60.34	12.5	53.3
70	12.5	53.7
80	12.5	54.0
90	12.5	54.3
100	12.5	54.5
200	12.6	55.4
300	12.6	55.7
400	12.6	55.8
500	12.6	55.9
600	12.6	56.0
700	12.6	56.0
800	12.6	56.0
900	12.6	56.1
1000	12.6	56.1
2000	12.6	56.2
10000	12.6	56.2

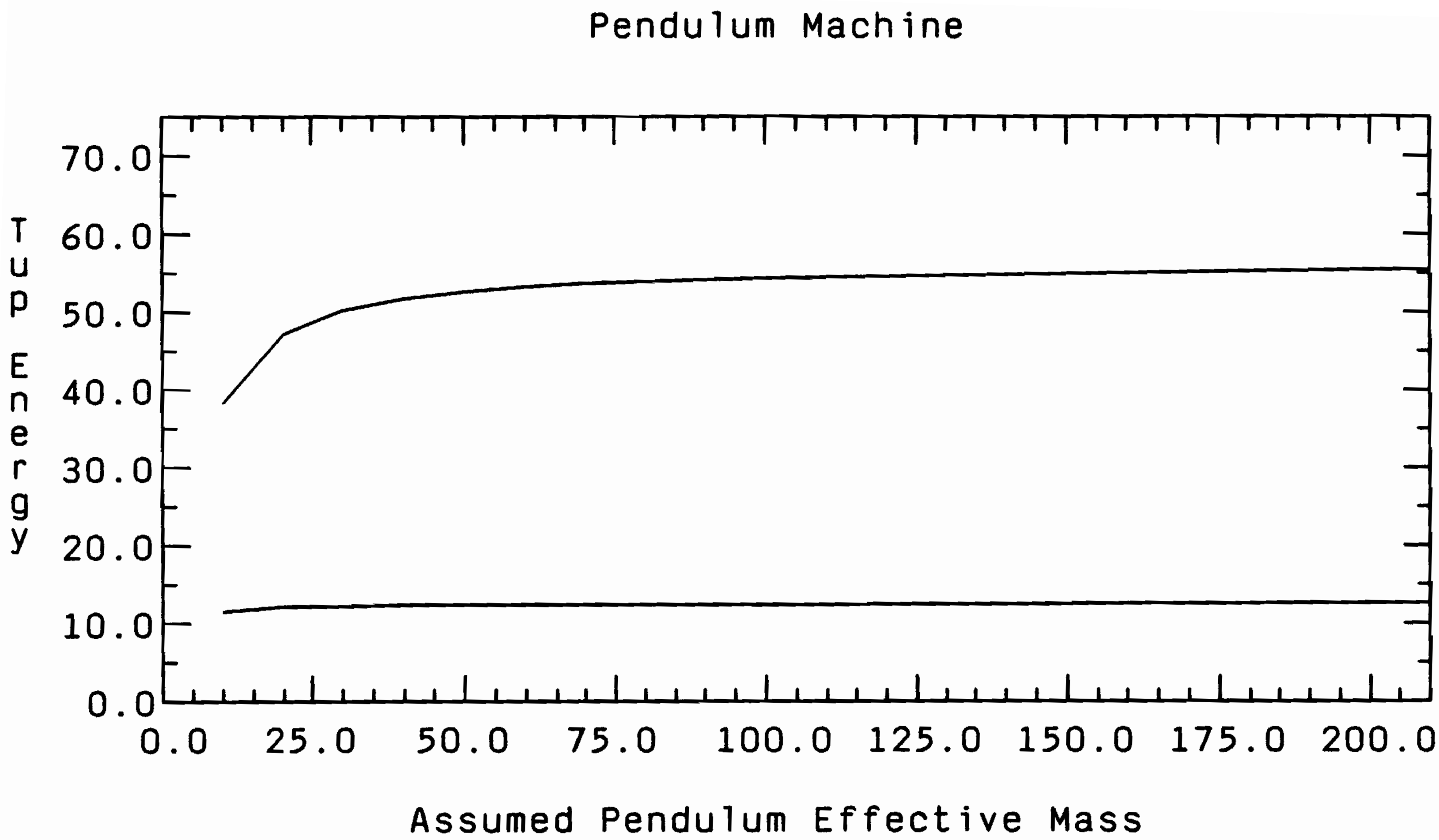


Figure 3-10(A). Effects of Assumed Meff on Pendulum 'Tup Energy'

3.3.3.2 Drop Tower Crosshead Mass

Table 3-18(B)
Assumed Effects of M_{eff} on Drop Tower Test 'Tup Energy'

Assumed Weight of Cross Head [lb]	Specimen L10 [ft-lb]	Specimen H5 [ft-lb]
2	8.6	-50.1
3	11.4	-12.5
4	12.8	6.2
5	13.7	17.5
6	14.2	25.0
7	14.6	30.4
8	14.9	34.4
9	15.2	37.6
10	15.4	40.1
20	16.2	51.3
30	16.5	55.1
40	16.6	57.0
50	16.7	58.1
60	16.7	58.9
70	16.8	59.4
80	16.8	59.8
90	16.8	60.1
100	16.9	60.4
110	16.9	60.6
120	16.9	60.8
130	16.9	60.9
140	16.9	61.0
150	16.9	61.1
160	16.9	61.2
170	16.9	61.3
179.62	16.9	61.4
180	16.9	61.4
190	16.9	61.4
200	16.9	61.5
300	17.0	61.9
400	17.0	62.1
500	17.0	62.2
1000	17.0	62.4
2000	17.0	62.5
10000	17.0	62.6
100000	17.0	62.6
1000000	17.0	62.6

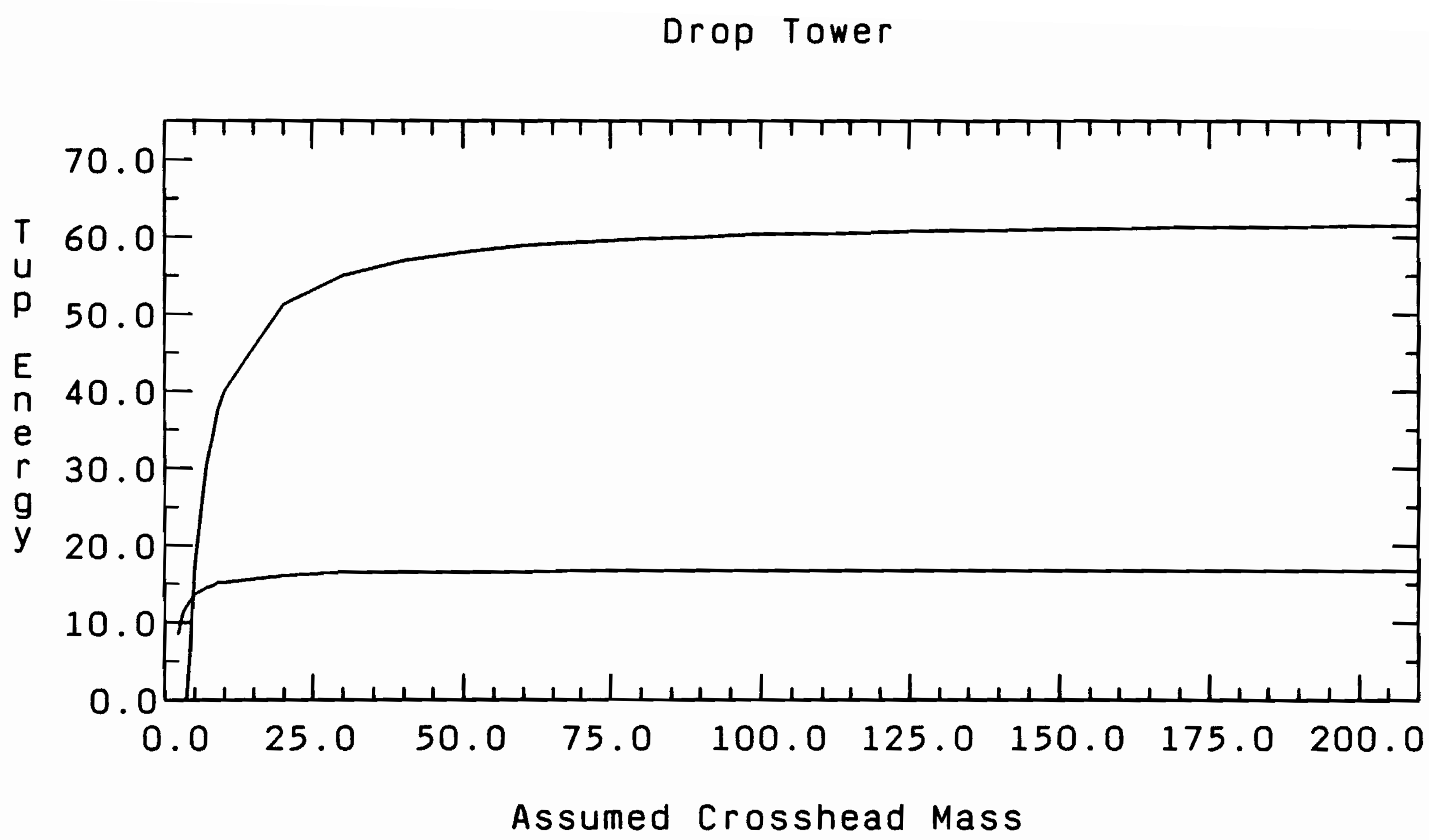


Figure 3-10(B). Effects of Assumed Meff on Drop Tower 'Tup Energy'

3.3.4 Effects of Initial Velocity

In Table 3-19, the results of using several initial velocities are compared with the initial velocity obtained from the measured value of h_0. Figures 3-11(A) and 3-11(B) show the result in graphic form.

3.3.4.1 Pendulum Velocity

Table 3-19(A)
Effects of Assumed Initial Velocity on Pendulum Test 'Tup Energy'

Assumed Initial Velocity of Pendulum [ft/sec]	Specimen Identification L2 [ft-lb]	Specimen Identification H6 [ft-lb]
10	7.3	30.4
11	8.1	33.7
12	8.8	37.0
13	9.6	40.4
14	10.3	43.7
15	11.1	47.0
16	11.9	50.4
16.4	12.2	51.7
16.5	12.2	52.0
16.6	12.3	52.4
16.7	12.4	52.7
16.8	12.5	53.1
16.882123	12.5	53.3
16.9	12.5	53.4
17.0	12.6	53.7
17.1	12.7	54.1
17.2	12.8	54.4
18	13.4	57.1
19	14.1	60.4
20	14.9	63.7
21	15.6	67.1
22	16.4	70.4

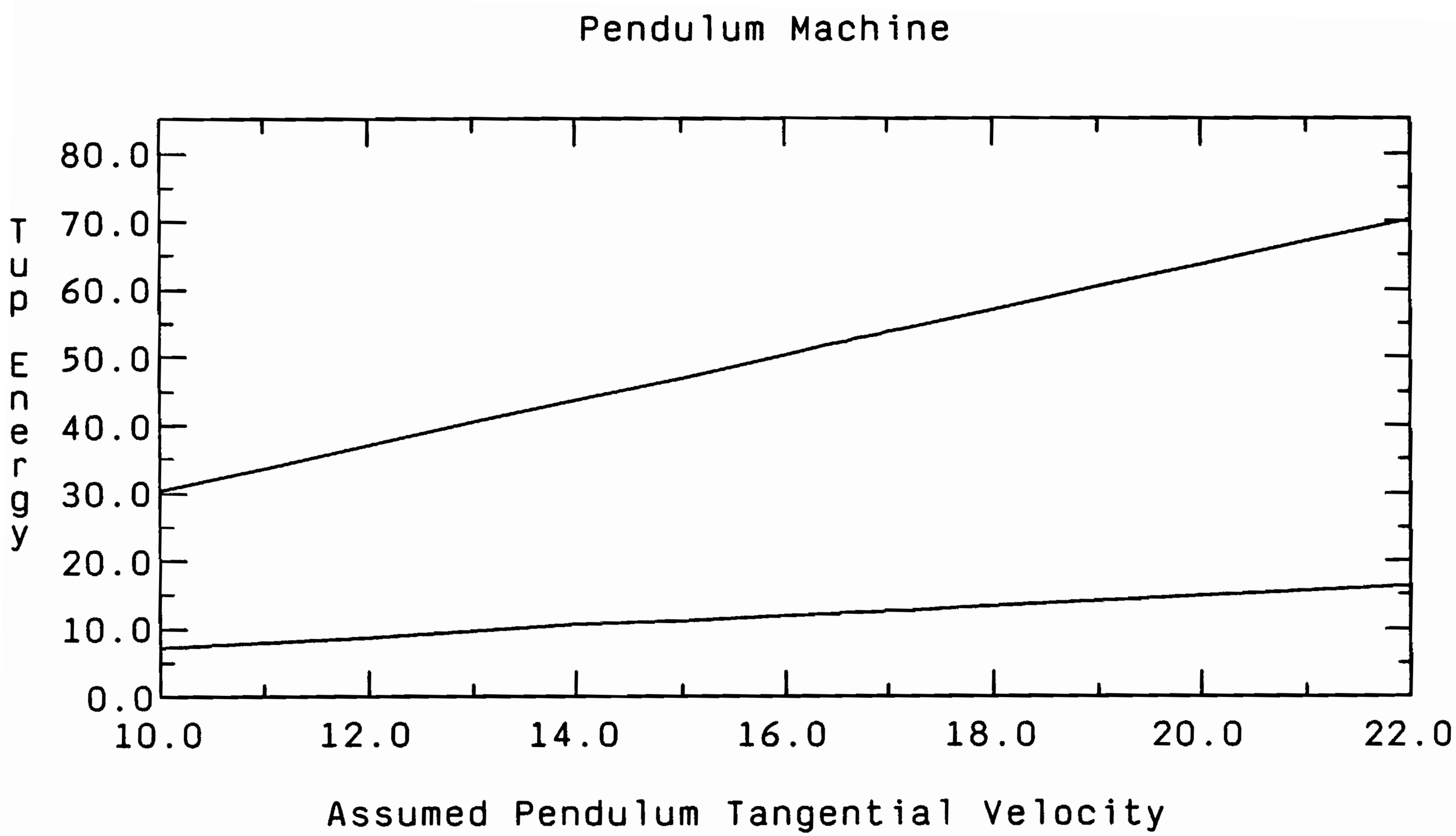


Figure 3-11(A). Effects of Assumed Pendulum Initial Velocity on Pendulum 'Tup Energy'

3.3.4.2 Drop Tower Crosshead Velocity

Table 3-19(B)
Effects of Assumed Initial Velocity on Drop Tower Test 'Tup Energy'

Assumed Initial Velocity of Drop Tower [ft/sec]	Specimen Identification L10 [ft-lb]	Specimen Identification H5 [ft-lb]
10	10.1	36.2
11	11.1	40.0
12	12.1	43.7
13	13.1	47.4
14	14.2	51.2
15	15.2	54.9
16	16.2	58.7
16.4	16.6	60.2
16.5	16.7	60.6
16.6	16.8	60.9
16.7	16.9	61.3
16.7176	16.9	61.4
16.8	17.0	61.7
16.9	17.1	62.1
17.0	17.2	62.4
17.1	17.3	62.8
18	18.2	66.2
19	19.3	69.9
20	20.3	73.7
21	21.3	77.4
22	22.3	81.2

3.3.5 Effects of Calibration Order

In Table 3-20, the results of using first and second order tup calibrations in conjunction with second order integration are compared.

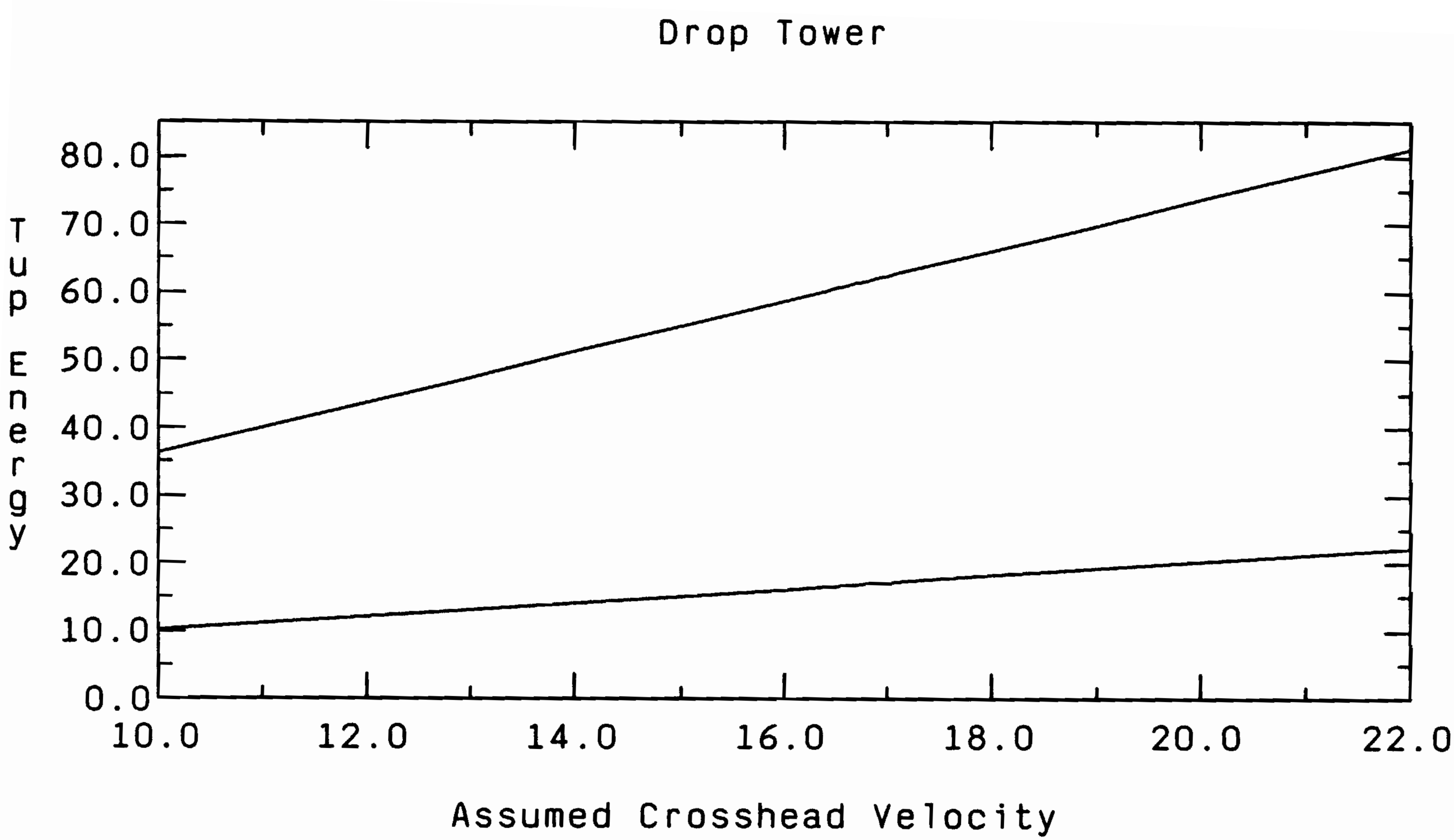


Figure 3-11(B). Effects of Assumed Drop Tower Initial Velocity on Drop Tower 'Tup Energy'

Table 3-20(A)
Effects of Tup Calibration Order on 'Tup Energy'

Specimen ID	1st Order [ft-lb]	2nd Order [ft-lb]
H6	54.2	53.3
H7	52.5	51.6
H8	48.4	47.6
H10	50.5	49.7
H11	54.1	53.2
L2	12.6	12.5
L3	12.2	12.1
L4	13.0	12.9
L5	12.4	12.3
L6	12.6	12.6

Table 3-20(B)
Best 1st and 2nd Order Calibration Curves
for the Effects Technology Pendulum Charpy Tup

$$\hat{Y} = \text{Zerovolts} + .69892\text{e-}04\ X$$

$$\hat{Y} = \text{Zerovolts} + .72656\text{e-}04\ X - .43673\text{e-}09\ X^2$$

where: $\hat{Y}$ ≡ Output of the Effects Technology Pendulum Charpy Tup in volts

X ≡ Load in pounds

Zerovolts ≡ Tup output corresponding to zero load (determined separately from stored data for each test)

3.3.6 Effects of Calibration Constants

Table 3-21 compares the results of using various quadratic calibration curve fits with the result of using the best curve fit according to the statistical analysis shown above. Table 3-21(A) gives the results for the pendulum test; Table 3-21(B) presents the comparable data for the drop tower. The data are given in ft-lb.

3.3.6.1 Pendulum Tup Static Calibration

Table 3-21(A)
Effects of Varying Calibration Constants
on Pendulum Test 'Tup Energy'
(in ft-lb)

Calibration Curve:

$$\hat{Y} = \text{Zerovolts} + \text{Loadfac1} * X + \text{Loadfac2} * X^2$$

where:
- $\hat{Y}$ ≡ Output of Effects Technology Pendulum Charpy tup in volts
- Zerovolts ≡ Tup output corresponding to zero load (determined separately for each test)
- Loadfac1, Loadfac2 ≡ Calibration coefficients
- X ≡ Load in pounds

Load Fac 1 / Load Fac 2	6.2656e-05	7.2656e-05	8.2656e-05
-3.3677e-10	L2: 14.5	L2: 12.4	L2: 10.8
-3.3677e-10	H6: 61.3	H6: 53.1	H6: 46.8
-4.3677e-10	L2: 14.6	L2: 12.5	L2: 10.9
-4.3677e-10	H6: 61.8	H6: 53.3	H6: 47.0
-5.3677e-10	L2: 14.8	L2: 12.6	L2: 11.0
-5.3677e-10	H6: 62.2	H6: 53.6	H6: 47.1

3.3.6.2 Drop Tower Tup Static Calibration

Table 3-21(B)
Effects of Varying Calibration Constants
on Drop Tower Test 'Tup Energy'
(in ft-lb)

Calibration Curve:

$$\hat{Y} = \text{Zerovolts} + \text{Loadfac1} * X + \text{Loadfac2} * X^2$$

where:

$\hat{Y}$	≡	Output of Ireland Associates drop tower Charpy tup in volts
Zerovolts	≡	Tup output corresponding to zero load (determined separately for each test)
Loadfac1, Loadfac2	≡	Calibration coefficients
X	≡	Load in pounds

Load Fac 1 / Load Fac 2	4.4942e-05		5.4942e-05		6.4942e-05	
+1.5000e-10	L10:	20.2	L10:	16.6	L10:	14.1
+1.5000e-10	H5:	73.5	H5:	60.7	H5:	51.7
+0.0000e-10	L10:	20.7	L10:	16.9	L10:	14.3
+0.0000e-10	H5:	74.7	H5:	61.4	H5:	52.1
-1.5000e-10	L10:	21.3	L10:	17.3	L10:	14.5
-1.5000e-10	H5:	76.0	H5:	62.1	H5:	52.5

3.3.6.3 Artificial Calibrations

As described in section 4.2.2.2.4.1, it is possible to find calibration constants resulting in 'tup energies' whose averages closely approximate the averages of the pendulum 'dial energies' for the two populations. The results of such a calibration are shown in Tables 3-22(A) and 3-22(B). The term "artificial calibration" is used at this point pending the discussion in section 4.2.2.2.4.1 concerning the validity of such an approach.

3.3.6.3.1 Artificial Calibration for the Pendulum Tup

Table 3-22(A)
Calibration Curve which Closely Matches
Pendulum 'Tup Energy' to Pendulum 'Dial Energy'
with Associated Calculated 'Tup Energies'

Calibration Curve:

$$\hat{Y} = \text{Zerovolts} + \text{Loadfac1} * X + \text{Loadfac2} * X^2$$

where:

$\hat{Y}$	≡	Output of Effects Technology pendulum Charpy tup in volts
Zerovolts	≡	Tup output corresponding to zero load (determined separately for each test)
Loadfac1, Loadfac2	≡	Calibration coefficients
X	≡	Load in pounds

Specific Calibration Curve:

$$\hat{Y} = \text{Zerovolts} + 6.5811e\text{-}05\ X + 9.5040e\text{-}10\ X^2$$

Specimen ID	'Tup Energy' [ft-lb]
H6	54.7
H7	53.1
H8	49.0
H10	51.0
H11	54.7
High Energy Specimens' Average	52.50
L2	12.4
L3	12.0
L4	12.9
L5	12.2
L6	12.5
Low Energy Specimens' Average	12.40

3.3.6.3.2 Artificial Calibration for the Drop Tower Tup

Table 3-22(B)
Calibration Curve which Closely Matches
Drop Tower 'Tup Energy' to Pendulum 'Dial Energy'
with Associated Calculated 'Tup Energies'

Calibration Curve:

$$\hat{Y} = \text{Zerovolts} + \text{Loadfac1} * X + \text{Loadfac2} * X^2$$

where:

$\hat{Y}$	≡	Output of Ireland Associates drop tower Charpy tup in volts
Zerovolts	≡	Tup output corresponding to zero load (determined separately for each test)
Loadfac1, Loadfac2	≡	Calibration coefficients
X	≡	Load in pounds

Specific Calibration Curve:

$$\hat{Y} = \text{Zerovolts} + 4.70850\text{e-}05\ X + 6.5032\text{e-}09\ X^2$$

Specimen ID	'Tup Energy' [ft-lb]
S2	60.2
H1	47.8
H3	51.5
H4	55.2
H5	50.7
High Energy Specimens' Average	53.08
L7	11.9
L8	12.9
L9	13.0
L10	12.2
L11	11.5
L12	11.8
Low Energy Specimens' Average	12.22

3.3.7 Compliance Calculations

System compliance data were obtained for the pendulum machine by obtaining the general yield load and general yield deflection, then determining the quotient of general yield deflection divided by general yield load. The results are reported in Table 3-23(A). Only "L" series specimens are used because the results have good yield load consistency due to the high sampling rate while the H series specimens had poorer consistency due to a lower sampling rate.

3.3.7.1 Compliance Calculation for the Pendulum (Static Tup Calibration)

Table 3-23(A)
Compliance Data for the Pendulum Test System
Calculated Using Static Tup Calibration

Calibration Curve:

$$\hat{Y} = 0.72656e{-}04\ X - 0.43677e{-}09\ X^2$$

where: $\hat{Y}$ ≡ tup-amplifier combination output in volts
X ≡ load in pounds

Specimen Id	Yield Deflection [in]	Yield Load [lb]	Compliance [μin/lb]	'Tup Energy' [ft-lb]
L2	0.0299	7446	4.02	12.5
L3	0.0294	7111	4.13	12.1
L4	0.0294	7224	4.07	12.9
L5	0.0289	7248	3.99	12.3
L6	0.0313	7424	4.22	12.6

3.3.7.2 Compliance Calculation for the Drop Tower Tup

3.3.7.2.1 Static Calibration for the Drop Tower Tup

System compliance data were obtained for the drop tower in the same way. The results are reported in Table 3-23(B).

Table 3-23(B)
Compliance Data for the Drop Tower Test System
Calculated Using Static Tup Calibration

Calibration Curve:

$$\hat{Y} = 0.54942e\text{-}04\ X$$

where: $\hat{Y}$ ≡ tup-amplifier combination output in volts
X ≡ load in pounds

Specimen Id	Yield Deflection [in]	Yield Load [lb]	Compliance [μin/lb]	'Tup Energy' [ft-lb]
L7	0.0274	9336	2.93	16.6
L8	0.0297	9351	3.18	18.2
L9	0.0279	9233	3.02	18.4
L10	0.0301	9493	3.17	16.9
L11	0.0288	9227	3.12	15.9
L13	0.0288	9480	3.04	16.6

3.3.7.2.2 Dynamic Calibration for the Drop Tower Tup

3.3.7.2.2.1 Energy Matching

System compliance data were also calculated using the calibration curve in Table 3-22(B). This curve causes the drop tower 'tup energy' to match the pendulum 'dial energy.' The results are reported in Table 3-23(C).

Table 3-23(C)
Drop Tower Compliance Calculations
Using Calibration Curve from Table 3-22(B)

Calibration Curve:

$$\hat{Y} = 0.39586e\text{-}04\ X + 0.83032e\text{-}09\ X^2$$

where: $\hat{Y}$ ≡ tup-amplifier combination output in volts
X ≡ load in pounds

Specimen Id	Yield Deflection [in]	Yield Load [lb]	Compliance [μin/lb]	'Tup Energy' [ft-lb]
L7	0.0274	5829	4.70	11.9
L8	0.0297	5835	5.09	12.9
L9	0.0283	5840	4.85	13.0
L10	0.0302	5892	5.12	12.2
L11	0.0288	5785	4.98	11.6
L13	0.0274	5891	4.65	11.9

3.3.7.2.2.2 Load Matching

Finally, system compliance for the drop tower was recalculated using a linear calibration curve which forces the average of the drop tower general yield load to equal the average of the pendulum general yield load. The results are shown in Table 3-23(D).

Table 3-23(D)
Drop Tower Compliance Calculations Using Linear Calibration Curve Which Closely Matches Drop Tower General Yield Load With Pendulum General Yield Load

Calibration Curve:

$$\hat{Y} = 0.7048039e\text{-}04\ X$$

where: $\hat{Y} \equiv$ tup-amplifier combination output in volts
$X \equiv$ load in pounds

Specimen Id	Yield Deflection [in]	Yield Load [lb]	Compliance [μin/lb]	'Tup Energy' [ft-lb]
L7	0.0274	7278	3.76	12.9
L8	0.0297	7289	4.07	14.2
L9	0.0283	7298	3.88	14.4
L10	0.0302	7400	4.08	13.2
L11	0.0288	7192	4.00	12.4
L13	0.0288	7390	3.90	12.9

Using the calibration curve shown in Table 3-23(D), 'tup energies' were calculated for the specimens tested in the drop tower. The result is shown in Table 3-24.

Table 3-24
Calibration Curve Which Closely Matches
Drop Tower General Yield Load With Pendulum General Yield Load
With Associated Calculated 'Tup Energies'

Calibration Curve:

$$\hat{Y} = 0.7048039e\text{-}04 \; X$$

where: $\hat{Y}$ ≡ tup-amplifier combination output in volts
X ≡ load in pounds

Specimen ID	'Tup Energy' [ft-lb]
S2	57.2
H1	47.2
H3	48.1
H4	52.3
H5	48.0
High Energy Specimens' Average	50.56
L7	12.9
L8	14.2
L9	14.4
L10	13.2
L11	12.4
L12	12.9
Low Energy Specimens' Average	13.33

The importance of the compliance calculations is made clear in the Discussion.

4. DISCUSSION

4.1 Calibration

4.1.1 Calibration of the Tups - Finding the Best Calibration Curve

4.1.1.1 Pendulum Tup Calibration

4.1.1.1.1 Ireland Associates 20,000 lb Tup Calibration

Table 3-3(A)(1) shows that a linear curve fit for the voltage output of the Ireland Associates 20,000 lb tup is highly significant and explains the voltage output in terms of load quite well. Several facts support such a claim. First, the curve fit plus the pure error explain all but less than 0.02% of the sum of squares. Second, the t ratios for the intercept and coefficient of the linear term must be about two orders of magnitude larger than the minimum value, 2.62, for the likelihood of their being insignificant to be no more than 1%. Hence these coefficients are <u>highly</u> significant.

Table 3-3(A)(2) proves that a quadratic curve fit need not be considered. The improvement in explaining the sum of squares is quite small, as reflected by the F ratio which is an order of magnitude lower than the value (2.75) which it would have to be to ensure that the probability of the quadratic curve fit being insignificant is no more than 10%. The t ratio of the quadratic term is 0.15902 which is about an order of magnitude smaller than the value (1.29) which it would have to have to ensure that the probability of insignificance of the quadratic term is only 20%. Finally, while the intercept and the coefficient of the linear term have t ratios which ensure that they are highly significant, their values are nearly the same as the corresponding values in the linear model. Realistically, there is no difference between a linear model and a quadratic model whose first two terms are equal to those of the linear model and whose quadratic term is insignificant.

Table 3-3(A)(3) shows a third case. The F ratio reflecting improvement due to the third order is 3.7834 which is approximately equal to the F ratio, 3.84, necessary to make the probability of insignificance no more than 5%. Furthermore, while the intercept and coefficient of the linear term are approximately equal to the corresponding coefficients of the linear model, the t ratios of the quadratic and third order terms, 1.9431 and 1.9451, are almost large enough (1.98) to indicate that they have a likelihood of insignificance of no more than 5%, and they are large enough to show that the probability that they are <u>not</u> significant is smaller than 10%. It can be affirmed that the third order model, while not <u>highly</u> significant, is nonetheless significant. Even so, that model has been rejected for two reasons. First, it is not highly significant. Second, and more importantly, the third order method explains only a small fraction of the unexplained variance and can, therefore, be ignored for the purposes of calibration.

Similar arguments can be used to reject all higher level curve fits.

4.1.1.1.2 Calibration of the Effects Technology Tup Against the Ireland Associates 20,000 lb Tup

Note that while Table 3-3(B)(1) shows that the linear curve fit and both of its coefficients are highly significant in the calibration of the Effects Technology Tup, Table 3-3(B)(2) shows that the quadratic curve fit is also highly significant. Furthermore, the linear model has much more lack of fit than the quadratic curve fit. Clearly then, at least a quadratic curve fit is necessary.

Next consider the data in Tables 3-3(B)(3) and 3-3(B)(4), the analysis of variance tables for the third and fourth order curve fits. These tables show that the the third and fourth order curve fits are insignificant. Consequently, it is reasonable to assume that all higher curve fits are also likely to be insignificant.

Thus, it appears that the quadratic curve fit is both necessary and sufficient. Substituting the linear curve fit in Tables 3-3(A)(1) into the quadratic curve fit in Table 3-3(B)(2) yields the quadratic curve fit in Table 3-3(C) which shows the calibration curve for the Effects Technology Pendulum Charpy tup in terms of potential (in volts) and load (in pounds).

4.1.1.2 Drop Tower Tup Calibration

4.1.1.2.1 Instron Universal Tensile Testing Machine Calibration

The Ireland Associates drop tower Charpy tup was calibrated against the voltage output of an Instron tensile testing machine, but before that could be done, the Instron voltage output had to be calibrated against load. The analysis of variance of the Instron voltage output was somewhat hindered because of lack of replication, but is still reasonably useful.

Table 3-6(A)(1) shows that the linear curve fit is highly significant. By contrast, Table 3-6(A)(2) shows that the quadratic curve fit is insignificant. Also, the F ratio for improvement due to the third order is 1.0172 in Table 3-6(A)(3), but with degrees of freedom equal to 1 and 6, an F ratio equal to 5.78 would have been necessary to ensure that the likelihood of error in affirming significance is no more than 10%. In other words, the cause of the low number of degrees of freedom and an F ratio of 1.0172 is a curve fit which is <u>not</u> significant.

4.1.1.2.2 Ireland Associates Drop Tower Charpy Tup Calibration

Next consider the calibration of the Ireland Associates drop tower Charpy tup. Tables 3-6(B)(1) through 3-6(B)(3) show that both the first and second order curve fits are highly significant and that the third order curve fit is significant, but not highly so. Note that the first order curve fit explains nearly all of the variance and that even though the F test shows it to be significant, the third order curve fit is necessary to explain only a very small amount of the variance. It would seem that the linear curve fit or possibly the quadratic curve fit should be adopted.

Substituting the linear curve fit in Table 3-6(A)(1) into the linear curve fit in Table 3-6(B)(1) yields the linear tup calibration shown in Table 3-6(C)(1). Similarly substituting the linear curve fit in Table 3-6(A)(1) into the quadratic curve fit in Table 3-6(B)(2) yields the quadratic tup calibration shown in Table 3-6(C)(2).

4.1.1.3 Recalibration of the Ireland Associates Drop Tower Charpy Tup

The calibration approach described in section 4.1.1.2 leads to fairly large standard errors in the estimated calibration coefficients. Furthermore, when compliance calculations were performed using the calibration curve reported in Table 3-6(C)(1) a contradiction resulted: the population drop tower "tup energy" averages were significantly higher than the corresponding population pendulum "dial energy" averages, but the calculated drop tower compliance was lower than the corresponding pendulum value. The drop tower Charpy tup was recalibrated against load rings to confirm the results.

The logic upon which the correct calibration curve was chosen from those reported in Tables 3-8(A)(1), 3-8(A)(2) and

3-8(A)(3) is essentially the same as described above. The calibration curve reported in Table 3-8(A)(1) was selected. Note that given the size of the errors reported in 3-6(C)(1), the two curves are not significantly different in value. This fact increases confidence in both.

The calibration curve in Table 3-8(A)(1) was adopted since it has a more precise coefficient.

4.1.2 Effective Mass

4.1.2.1 Effective Mass of the Pendulum

If the period of the pendulum, which is 1.9002 with a high degree of confidence (see Table 3-9), is substituted into the equation in section 5.2.5.2 of ASTM E-23 [44], the result is the radius of percussion (r_p): 2.9452 feet, i.e., 35.342 in.

Substituting r_p, r_{cg}, M_{act}, and r_s from section 3.1.2.1 into Equation 1-80 yields the following calculated value of M_{eff}:

$$M_{eff} \quad = \quad 60.34 \text{ lb} / 1 \text{ gravity} \qquad (4\text{-}1)$$

4.1.2.2 Effective Mass of the Drop Tower Crosshead

As has been seen in section 3.1.2.2, the effective mass was determined to be 179.62 lb divided by the acceleration of gravity.

4.1.3 Initial Velocities

4.1.3.1 Initial Velocity of the Pendulum

Efforts to obtain initial and/or final velocities for the pendulum by the flag system were complicated by the fact that the infra-red detector registered the presence of the flag before it arrived directly in front of the detector and continued to register its presence some distance after it was gone. It was decided that the uncertainties involved in the peripheral vision problem were worse than the assumption that there was no drag in the pendulum bearing. Consequently, it was decided to use the well-known equation:

$$V_o = (2gh)^{\frac{1}{2}} \qquad (4\text{-}2)$$

Using the initial height obtained from Tinius Olsen (see section 3.1.3.1), Equation 4-2 yields:

$$V_o = 16.882123 \text{ ft/sec} \qquad (4\text{-}3)$$

4.1.3.2 Initial Velocity of the Drop Tower Crosshead

If the crosshead were falling freely with negligible friction, it should be possible to fit a quadratic model to the data in Tables 3-10(A) and 3-10(B) as follows:

$$V(t) = X_o + V_o t + \tfrac{1}{2}gt^2 \tag{4-4}$$

where:

$V(t) \equiv$ the velocity at time t (4-5)

$X_o \equiv$ the position at time t=0 (4-6)

$V_o \equiv$ the velocity at time t=0, and (4-7)

$g \equiv$ the local acceleration of gravity. (4-8)

However, when the curve fit was attempted (Tables 3-11(A) and 3-11(B)), the linear coefficient was consistent with V_o as calculated for frictionless free fall, but the quadratic coefficient was not consistent with ($\tfrac{1}{2}$)(32.17 ft/sec^2). Hence, it was decided to try another approach to the problem.

Assume Equation 4-4 and assume that the data in Table 3-10(A) or 3-10(B) are highly accurate. If X(t=0) is substituted for X_o and the X(t) and t values from each row of

Tables 3-10(A) and 3-10(B) were substituted into Equation 4-4, the result would be six equations (for Table 3-10(A)) and five (for Table 3-10(B)) in two unknowns. Linear algebra could be applied to each combination of two equations to obtain the value of $\frac{1}{2}g$ and V_0, thus producing a large number of estimates for V_0 and g. These estimates could be averaged to obtain an extremely good estimate of the values for V_0 and g. Tables 3-12(A) and 3-12(B) provide the results of that approach, once again given drop tower height and the position of the image when t=0. In each case, the estimated value for V_0 <u>is</u> consistent with frictionless free fall, but the estimated value of g is <u>not</u>!

Consider the interval between two flashes (0.0049"). The contribution to distance due to acceleration during the interval is $(\frac{1}{2}g)(\Delta t^2)$ which is about 0.0005". However, distance traveled in the picture was measured by comparing the distance between images recorded at adjacent flash marks to a steel scale calibrated in units of 1/100", photographed immediately after the event. In other words, successful estimation of g from adjacent images of the tape mark would have required accuracy just a little finer than last count.

Estimation of V_0 is easier and hence more accurate. The contribution to distance travelled during an interval due to the initial velocity at the start of that interval is V_0t which amounts to about an inch or about 100 times last count. Therefore, V_0 can be estimated to a precision of about 1% which, as will be seen, is accurate enough for estimation of impact energies to three significant figures. Note that this is in line with the precision of the V_0 estimates in Tables 3-12(A) and 3-12(B).

The reason for the difficulty was that the experiment was well designed for its intended purpose -- i.e., estimation of the important variable V_0, but it was poorly designed for the unimportant variable, g.

Since V_0 is consistent with frictionless free fall, the accepted value of local gravitational acceleration, 32.17 ft/sec^2, and the remaining distances to fall until the level of impact was reached could be used to calculate the initial velocity used in the impact energy calculations. It was decided to use the equivalent approach of using Equation 4-2 and the data from section 3.1.3.2.2. The result was initial velocity of 16.717562 ft/sec. Since this is an intermediate result, it is not rounded to four significant figures as dictated by the accuracy of the gravitational acceleration.

4.1.4 Amplifier Rise Time

It is crucial that amplifier rise time be smaller than tup output rise time during instrumented impact testing. If this is not so, the impact event signal can be distorted. [46]. In fact, the amplifier used in this study <u>is</u> fast enough to produce an undistorted signal.

The amplifier used for this analysis has several gain settings, but the lowest gain level was used for all data collected in this study. Figure 3-8 shows the output (in red) of the amplifier set to its lowest gain setting resulting when a square wave (in blue) is input whose amplitude is equivalent to instrumented pendulum Charpy tup output corresponding to about 7700 lb.

On the vertical scale 13 small divisions are equivalent to 0.5 volts. On the horizontal scale, 13 small divisions are equivalent to 0.5 μsec.

Note that rise time from 10% to 90% of the square wave amplitude takes 3 small divisions which is equivalent to 0.12 μsec. Consider Figure 3-9(A), the ASTIR output for instrumented pendulum Charpy impact test L2. The general yield load (7446 lb from Table 3-23(A)) was not reached until 0.148 milliseconds after event start. In other words, the amplifier response is more than three orders of magnitude faster than the event it is monitoring.

In spite of the fact that only the first gain setting was used, other studies of rise time were performed for higher gain settings. It was not surprising that the higher gain settings produced a slower response, but the highest gain setting (which had the slowest response) had a rise time of approximately 0.75 μsec -- some two orders of magnitude faster than the event.

Thus the amplifier rise time of the first gain setting was easily fast enough to ensure that the amplifier could not possibly have distorted the signal from the tup.

4.1.5 Summary of the Calibration Discussion

The calibration data used for the experimental results section are summarized in Table 3-13. Since these are intermediate results used for other calculations, they have not been rounded in order to avoid round-off error.

4.2 Impact Tests

In the Introduction, a theory of measurement of impact energy was developed. If it is correct, the 'tup energy' and 'dial energy' can be used interchangeably and so can the pendulum and drop tower Charpy tests. Furthermore, there would then be reason for confidence in using the intermediate results (for example load and displacement) in making quantitative dynamic fracture toughness and stress-strain calculations as well as reason to believe that subdivisions of the impact energy (such as crack initiation and crack propagation) have validity. The similarities or differences among the impact energy measurements will be considered in the light of statistics, the requirements of ASTM specification E-23, and intuition.

4.2.1 Statistical Comparisons of Energy Results

Consider the four groups shown in Table 3-16:

1. the low energy specimens which were tested in the pendulum machine;
2. the low energy specimens which were tested in the drop tower machine;
3. the high energy specimens which were tested in the pendulum machine;
4. the high energy specimens which were tested in the drop tower machine.

Now also consider Tables 3-14 and 3-15. In terms of chemistry and dimensions, there is no bias among any of the groups. Recall from Chapter 2, "Procedure," that groups 1 and 2 were heat treated in one way and groups 3 and 4 were heat treated differently. There was, however, no heat-treat bias between group 1 and 2 or between groups 3 and 4. Thus, groups 1 and 2 form one population and groups 3 and 4 form another.

It follows that any differences which might be observed in the impact energies absorbed by the two populations do not reflect differences between the populations, but rather differences in test methods. Conversely, if impact energies ascribed by two different methods to either population agree, then the two methods are in agreement.

To examine the similarity or differences among the groups, the F ratio [47] and the t ratio (for the case that the variances cannot be assumed equal [48]) were used. It will be seen that the variances of some of the groups could be assumed equal, but if the methodology for unequal variance is applied to two sets of data with equal variances, the t ratio and the degrees of freedom calculated will be nearly the same as those calculated if the t methodology for equal variances were used. The converse is <u>not</u> true. The t methodology for equal variances, if applied to two sets of data with different variances, will give results which are clearly different from the correct ones. It seems intuitively obvious that the methodology for unequal variances should be used.

The impact energies as measured in the three different ways are summarized in Table 3-16.

When the pendulum 'dial energies' were compared with the pendulum 'tup energies,' the following results were found:

1. For the high energy specimens:

   F: 1.17 with 4 & 4 degrees of freedom;

   t: 1.16 with 8 degrees of freedom.

2. For the low energy specimens:

   F: 1.23 with 4 & 4 degrees of freedom;

   t: 0.985 with 8 degrees of freedom.

The F tests and t tests do not indicate significant differences between the precision or accuracy of the 'dial' and 'tup energies' in the pendulum Charpy test.

A comparison of the pendulum 'tup energies' and the drop tower 'tup energies' shows a different outcome:

1. For the high energy specimens:

   F: 4.73 with 4 & 4 degrees of freedom;

   t: 5.16 with approximately 5 degrees of freedom.

The F test shows that the drop tower 'tup energy' has less precision than the pendulum 'tup energy' with between 5 and 10% likelihood of being wrong. The t test shows that the drop tower 'tup energy' and the pendulum 'tup energies' are significantly different with less than 1% chance of being in error.

2. For the low energy specimens:

   F: 10.61 with 5 & 4 degrees of freedom;

   t: 10.85 with approximately 4 degrees of freedom;

The F test shows that the drop tower 'tup energy' has less precision than the pendulum 'tup energy' with between 2.5 and 5% likelihood of being in error. The t test shows that the drop tower 'tup energies' and the pendulum 'tup energies' are significantly different with less than 1% chance of being wrong.

An examination of the outcome of the statistical results of the pendulum 'dial energy' in comparison with the drop tower 'tup energy' shows similar lack of agreement:

1. For the high energy specimens:

   F: 5.53 with 4 & 4 degrees of freedom;

   t: 4.56 with approximately 5 degrees of freedom.

The F test shows that the drop tower 'tup energy' is less precise than the pendulum 'dial energy' with between 5 and 10% likelihood of being in error. The t test shows that the drop tower 'tup energy' is significantly different from the pendulum 'dial energy' with less than 1% chance of being wrong.

2. For the low energy specimens:

F: 13.01 with 5 & 4 degrees of freedom;

t: 11.39 with approximately 4 degrees of freedom.

The F test shows that the drop tower 'tup energy' is less precise than the pendulum 'dial energy' with between 1 and 2.5% chance of being in error. The t test shows that the two impact energy measurements are different, with much less than 0.5% likelihood of being wrong.

ASTM method E-23 requires that any qualified Charpy testing machine should match, on average, the AMMRC standards to within 1.0 ft-lb or 5.0% of the nominal value, whichever is larger. [49]. The Tinius Olsen model 74 Charpy testing machine used in this study has been calibrated to the ASTM E-23 requirements, and thus the pendulum 'dial energies' reported in column 3 of Table 3-16 can be assumed to be correct (i.e., they are secondary standards) to within those requirements, <u>but no closer</u>. Therefore, if a measurement were within 1.0 ft-lb or 5% of the pendulum 'dial energy,' one would have to conclude that it would have an excellent likelihood of meeting the E-23 accuracy requirement.

The averages of the pendulum 'dial energies' for the low energy and high energy specimens are, respectively: 12.3 ft-lb and 52.8 ft-lb. The averages of the pendulum 'tup energies' for the low energy and high energy specimens are, respectively: 12.48 ft-lb and

51.08 ft-lb. If the pendulum 'dial energy' were actually the AMMRC standard, the pendulum 'tup energy' would have easily passed the E-23 requirement.

The averages of the drop tower 'tup energies' for the low energy and high energy specimens are respectively: 17.10 ft-lb and 64.54 ft-lb. These results would not have met the requirements of ASTM E-23 if compared to <u>either</u> the pendulum 'dial' <u>or</u> 'tup energies.'

As explained earlier, 'tup energies' are calculated from electric potential vs. time records and, as Figures 3-9(A) - 3-9(U) show, such records are not free of electrical noise. One would have to conclude that the pendulum 'tup energy' data <u>must</u> contain more variance than the pendulum 'dial energy.' Indeed, such would seem to be the case if the data in Table 3-16 are considered. The variances of the pendulum 'dial energy' for the low energy and high energy specimens are respectively: 0.075 and 5.075. The variances of the pendulum 'tup energies' for the low energy and high energy specimens, respectively, are: 0.092 and 5.927. The difference between these pairs of variances can be identified as the variance due to electronic noise. Clearly, it is a fraction of the variance due to the pendulum test itself. Intuition shows that the electronic noise causes a reduction in precision, but as the statistical results show, the reduction in precision due to the noise is insignificant.

The variances of drop tower 'tup energy' for the low energy and high energy specimens are: 0.976 and 28.063 respectively. Since the same amplifier with the same amount of electronic noise was used for the pendulum 'tup energy' and the drop tower 'tup energy' results, it can immediately be seen by applying the results of the last paragraph that the loss in precision caused by going to the drop tower test could not have been due solely to electronic noise. Indeed, the variance contribution from the electronic noise was an insignificant amount of the total increase in variance.

Intuition leads to one final insight. From the impact energy data in Table 3-16, note that the pendulum 'dial energy' is always lower than, or equal to, the pendulum 'tup energy' for the low energy specimens and always higher than the pendulum 'tup energy' for the high energy specimens. These observations suggest that, in spite of the conclusions from the t test above, there may be a systematic difference between the pendulum 'dial' and 'tup energy' measurements.

4.2.2 Possible Causes of Discrepancies

As shown in section 4.2.1, pendulum 'tup energy' would be an acceptable measure of Charpy impact energy even though some residual differences between pendulum 'tup energy' and pendulum 'dial energy' remain. There is clearly a major problem with the drop tower 'tup energy' as a measure of Charpy impact energy. There are several possibilities for the source of the problem(s), but they all fit within three classifications:

1. Something is wrong with the numerical method.
2. Something is wrong with the calibration. Within this classification lie the following possibilities:
    a. The order of tup calibration is too low.
    b. The physical constants of the testing machine have been incorrectly or inaccurately measured.
    c. The calibration constants of the tups were incorrectly, inaccurately, or inappropriately measured.
3. Something is basically wrong with the theory, and hence, the method.

Each of these possibilities is discussed in turn.

4.2.2.1 Effect of Integration Method on Using Pendulum 'Tup Energy' as a Measure of Charpy Impact Energy

The numerical method used in the ASTIR program can be seen either as a double integration by a low order Newton-Cotes method or as a low order Euler solution to the simultaneous differential equations:

$$dV/dt = -P(t)/M \tag{4-9}$$

$$dx/dt = V \tag{4-10}$$

As has been stated earlier, it can be shown that such methods asymptotically approach the exact solution as the order of the numerical method is raised. Applying this fact to Table 3-17, it can be stated that the remaining error due to the numerical method at order two is less than 0.1 ft-lb. Intuitively, it appears that if there is an error in pendulum 'tup energy,' increasing the order reduces the error for the low energy specimens but makes it worse for the high energy specimens.

The results in Table 3-17 show that the error due to the numerical method is about 0.1 ft-lb. Therefore, further increases in the order of the method will probably not have a significant effect on the outcome. In fact, these results show that the 0^{th} order (rectangle rule) integration scheme is completely satisfactory and more sophisticated integration schemes are unnecessary.

4.2.2.2 Effect of Calibration on Using 'Tup Energy' as a Measure of Charpy Impact Energy

4.2.2.2.1 Effective Mass

The effects of errors in the effective mass of the pendulum or crosshead can be appreciated by examining Tables 3-18(A) and 3-18(B). Note that as the effective mass is raised, the calculated 'tup energy' first rises rapidly and then asymptotically approaches some upper limit. Note also that the mass at which the upper limit is reached increases as toughness of the sample increases.

The explanation for these two effects is as follows. The impulse applied by the specimen to the tup decelerates the pendulum or crosshead; hence the latter parts of the integration of (P)(ds) are reduced. If the effective mass is small, the deceleration is large, and the calculated 'tup energy' is small. However, if the effective mass is large, the deceleration is small, and the calculated energy is large. If the deceleration were negligible, any increase in pendulum or crosshead mass would have a negligible further effect. Obviously, the effective mass necessary to make deceleration negligible will depend upon the magnitude of the impulse. Therefore, increasing the effective mass will increase the calculated 'tup energy' until an asymptotic limit is reached. The effective mass necessary to reach that limit will increase with the toughness of the specimen. A simpler but less lucid way of restating this would be: as assumed effective mass rises, the second term of the Augland equation approaches zero.

There is an important consideration here for the design of impact machines. ASTM E-23 states that it would be ideal for the Charpy test to be performed with constant velocity. Constant velocity could be achieved by using a large effective mass. The difficulty of getting precise values for absorbed energy could be overcome by using an instrumented tup and calculating 'tup energy.'

4.2.2.2.1.1 Effective Mass of the Pendulum

As will be seen, for the average of the pendulum 'tup energies' to match exactly the average of the pendulum 'dial energies,' specimen L2 would have to have had a 'tup energy' of 12.4 ft-lb. That would require an effective mass of 40 to 50 lb mass, but the difference between even 50 lb mass and the measured effective mass of the pendulum is more than an order of magnitude larger than the experimental uncertainty. Similarly, for the two averages to be identical, H6 would have to have a 'tup energy' of 54.7 ft-lb which is inconsistent with the measurements. Not only are the required changes in the effective mass incompatible with actual measurements, but they are in the opposite directions, and hence are incompatible with one another!

4.2.2.2.1.2 Effective Mass of the Drop Tower Crosshead

The rejection of effective mass as a cause of the lack of agreement is even more obvious in the case of the drop tower. For the averages of the drop tower 'tup energies' to match the averages of the pendulum 'dial energies,' specimen S2 would have to have a 'tup energy' of 60.2 ft-lb and specimen L8 would have to have a 'tup energy' of 12.9 ft-lb. Table 3-18(B) does not even contain such values. The effective mass required would hence have to be less than the lowest effective mass on the table (namely 100 lb mass), but it is completely impossible that the scale used to measure the effective mass of the crosshead could be so much in error, since it was calibrated against a triple beam balance prior to use.

4.2.2.2.2 Initial Velocity of the Tup

As can be appreciated from the results in Tables 3-19(A) and 3-19(B), the effect of initial velocity on calculated 'tup energy' is less complicated than the effect of mass on calculated 'tup energy;' as the initial velocity rises, the 'tup energy' rises and the increase is very nearly proportional to the increase in velocity. Suppose that the effective mass were high enough that the deceleration was insignificant. Then the velocity would be a constant and the increment in absorbed energy, (P)(ds) would be equivalent to $(P)(V_O)(dt)$, i.e., $(V_O)(P)(dt)$. Since V_O is a constant, the absorbed energy would be just V_O multiplied by the impulse. Consequently, the absorbed energy would be proportional to V_O. If V were varying with time, as normally happens, its value at any given moment would still be a monotonically increasing function of V_O and, hence, absorbed energy is still a monotonically increasing function of V_O as seen in Tables 3-19(A) and 3-19(B).

The discussion in the last paragraph suggests that it might be possible to multiply the impulse by V_O and apply some sort of correction factor which would be a function only of impulse, effective mass, and initial velocity. This is exactly true and can be mathematically proven: the resulting equation is the Augland-Grumbach equation mentioned in the Introduction to this paper.

4.2.2.2.2.1 Initial Velocity of the Pendulum

As noted earlier, a perfect match between the pendulum 'dial' and 'tup energies' requires that specimen L2 have a 'tup energy' of 12.4 ft-lb. From Table 3-19(A) such a 'tup energy' would require an initial velocity of about 16.8 ft/sec. It is not implausible to assume that the initial velocity could be lower than the expected 16.8+ ft/sec by only 0.08 ft/sec. Windage losses or errors in measuring initial height might be large enough. However, the initial velocity also affects the 'tup energy' of the high energy specimens. For a perfect match between average pendulum 'dial' and average pendulum 'tup energies,' specimen H6 would have to have a 'tup energy' of 54.7. However, at an initial velocity of 16.8 ft/sec, specimen H6 has a 'tup energy' of 53.1. H6 has the desired 'tup energy' only if the initial velocity is between 17.2 ft/sec and 17.3 ft/sec. At that initial velocity, L2 has a 'tup energy' of 12.8 ft/sec -- which is unacceptable. More significantly, the pendulum may have some way (such as windage) to lose energy and move more slowly at the impact point than expected, but it has no way to get extra energy and move more rapidly than expected at the impact point. Due to conservation of energy, it is not possible for the pendulum to have a tangential velocity of 17.1 ft/sec or above.

4.2.2.2.2.2 Initial Velocity of the Drop Tower Crosshead

Once again, the case of the drop tower is more obvious than that of the pendulum. For the average of the drop tower 'tup energies' to match the average of the pendulum 'dial energies' would require specimen S2 to have a 'tup energy' of 60.2 ft-lb which, according to Table 3-19(B), would require an initial velocity of approximately 14.5 ft/sec. For the match required, specimen L8 would have to have a 'tup energy' of 12.9 ft-lb which would require an initial velocity of about 12.5 ft/sec. The required initial velocities are impossible for two reasons: First, they do not match each other. Second, the initial velocity measurements by strobe velocity treated in section 3.1.3.2.1 are completely incompatible with such low velocities.

4.2.2.2.3 Order of the Calibration Curve

It might be thought that the apparent lack of agreement between the pendulum 'dial energy' and the pendulum 'tup energy' is due to inadequate calibration curve order and that a comparison of the results from using the best linear calibration curve with the results from using the best quadratic calibration curve to calculate the absorbed energies of the pendulum Charpy tests should be done. Table 3-20(A), which contains such a comparison using the calibration curves shown in Table 3-20(B), shows that even in this case, in which quadratic calibration behavior was shown by statistics to be highly significant, the effects of using the lower order calibration curve resulted in an error of no more than 1.0 ft-lb for the high energy specimens and no more than 0.1 ft-lb for the low energy specimens. One would expect that the higher order calibration curves would help much less than the quadratic curve since their statistics show them to be one or more orders of magnitude less significant in terms of F ratio than the quadratic calibration curve. As a rough estimate, one would expect that the error eliminated in that way would be about an order of magnitude smaller than the error eliminated by changing from a linear to a quadratic calibration curve.

By the same logic, it can be argued that the ninth order curve which would result from using the two third order calibration curves found when calibrating the drop tower Charpy tup in two steps would be unlikely to change the results from the drop tower 'tup energy' calculations by even as much as 1.0 ft-lb. Hence it would be unable to explain the discrepancy between the drop tower and pendulum results.

4.2.2.2.4 Tup Calibration Constants

4.2.2.2.4.1 Static Calibrations

Small changes in the static calibration constants such as those shown in Tables 3-21(A) and 3-21(B) cannot make the average pendulum 'tup energy' exactly coincide with the pendulum 'dial energy' or make the average drop tower 'tup energy' even close. Nonetheless, Tables 3-21(A) and 3-21(B) can serve as starting points for a modified secant search for calibration curves which can do so. Table 3-22(A) contains a calibration curve which makes the average pendulum 'tup energies' virtually coincide with the average pendulum 'dial energies.' Table 3-22(B) contains a calibration curve which does the same for the drop tower 'tup energies.' The real objective is not to force 'tup energies' to fit the 'dial energies' by the use of some arbitrary calibration curve, but to determine whether such a curve is justifiable on the basis of sound physics.

A comparison of Table 3-3(C) with Table 3-22(A), and 3-8(A)(1) and 3-8(A)(2) with Table 3-22(B) suggests that it is statistically unlikely that the artificially modified calibration curves in Tables 3-22(A) and 3-22(B) could have any physical significance. This conclusion is based on the fairly small size of the standard errors in Table 3-3(C) which are calculated from the results in the two stages of the calibration. It must be borne in mind, however, that the calibration of the pendulum Charpy tup involved not two, but four, steps.

1. Weights (primary standard) were used to calibrate load rings (secondary standard).

2. Load rings (secondary standard) were used to calibrate a Tinius Olsen universal testing machine (certified testing device).
3. The Tinius Olsen universal testing machine was used to calibrate an Ireland Associates 20,000 lb tup (uncertified testing device).
4. The Ireland Associates tup was used to calibrate the pendulum Charpy tup.

It seems likely that the calibration curve in Table 3-22(A) is incorrect, but without more data the curve cannot be refuted.

4.2.2.2.4.2 Dynamic Calibration

Nonetheless, it could be argued that the static calibrations performed in this work are irrelevant to the experiment here. It has been argued [50] that tups for instrumented impact testing machines must be dynamically and not statically calibrated as done herein. Apart from the difficulty of obtaining and using a dynamic tup calibration, it is possible to argue against the physical validity of the results.

For the static tup calibration to be different from the dynamic calibration, at least one of three considerations must be true:

1. The elastic constants of the material from which the tup was manufactured would have to be strain rate dependent. Below the elastic limit this is just not true for common engineering materials [51] including low alloy steels like the material from which the tups used in this study were made. Furthermore, it is well known that the speed of sound in steel is independent of frequency, a fact which immediately implies that the elastic constants of steel must be independent of strain rate. [52];[53].

2. The amplifier used in the load measurement might respond differently at dynamic rates than it did at static rates. Such a contention requires that the amplifier rise time is slower than the event rise time, which is refuted in section 3.1.4.

3. The strain gauges or materials used to attach them might be strain rate dependent. It is more difficult to refute this possibility immediately, but in <u>The Strain Gauge Primer</u>, Perry and Lissner imply that strain gauges are strain rate insensitive. [54].

4.2.2.3 Basic Theoretical Considerations

The pendulum 'dial energy' results are different from the pendulum 'tup energy' for the two populations tested and the drop tower 'tup energy' results for the same two populations differ from both the pendulum 'dial energy' and pendulum 'tup energy' results. In the case of pendulum 'tup energies,' the difference when compared to pendulum 'dial energies' was shown to be insignificant. Therefore, the associated intermediate results can be used to calculate dynamic material constants. The drop tower 'tup energies' differ significantly from the pendulum 'dial energies.' Therefore the intermediate results calculated from the drop tower tup output have no fundamental significance.

If there is any problem with the static calibration, it has to do with strain gauge response. If this is rejected, one is forced to accept the final possibility: there is something basically wrong with the theory. In the case of the pendulum 'tup energy,' the problem is insignificant and hence unimportant in regard to the use of intermediate results in calculating material constants. In the case of the drop tower system used in this study, the problem is significant and the intermediate results have no fundamental significance.

There are two possible sources of error to be considered: friction and compliance.

4.2.2.3.1 Friction

ASTIR does not account for the amount of friction which occurs in the pendulum before, during, or after the impact event. On the other hand, the pendulum dial is adjusted so that after one swing with no specimen in place the dial shows zero absorbed energy even though it is obvious that some energy was absorbed and, at the same time, the non-zero energies are distributed along the scale between zero energy and maximum energy in some plausible manner. Remember, however, that this calibration would be acceptable if it were within 5 % of standard energy or 1 ft-lb, whichever were closer. Thus it is unreasonable to reject the 'tup energies' if they were within such a range of the 'dial energies' -- as in fact they were. Further statistics show that the pendulum 'tup' and 'dial energies' have no significant difference.

The linear bearings used in the drop tower testing machine are quite loose and it is entirely possible that a significant amount of friction could exist in the drop tower machine during the impact event if the event causes the crosshead to be torqued. ASTIR cannot take this into account and so would calculate a high energy value.

4.2.2.3.2 Compliance

When an impact event occurs, some of the energy and, hence, some of the load is used to begin elastic ringing in the machine and the specimen. Neither of these has to do with material constants of the specimen, but the ringing of the specimen should be the same in the pendulum and drop tower versions of the Charpy test. The ringing of the instrument is more serious. Not only does it have nothing to do with material constants, but it will vary from machine to machine according to the compliances of the machines.

Over the years, the pendulum version of the Charpy test has been purged of compliance problems. Izod obviously recognized the problem. Note that his original machine had guy wires attached to the pendulum in a configuration commonly used in naval rigging to add stiffness to masts. [55]. Bluhm [56] showed that a pendulum machine which produced excessively high absorbed energy results did so because of unacceptably high compliance in the pendulum. The requirement in ASTM E-23 that all Charpy pendulum machines adhere to standards based on results obtained from the very stiff machines at Watertown arsenal (later AMMRC) guaranteed that only very stiff pendulum Charpy machines would be used for serious studies of the impact properties of materials.

Since the drop tower Charpy test is not even permissible under ASTM E-23, it has not felt such pressures. In fact, in spite of the inherent stiffness of the crosshead, the drop tower machines can be MORE compliant than modern pendulum machines. For example, such could be the case if the tup and/or anvil used in the drop tower were more compliant than those used in the pendulum machine.

Note that Bluhm's model is <u>more</u> applicable to drop towers than it is to pendulum machines. [57]. It predicts that the compliance effect will be worse for tests in which high loads are reached than for those in which loads are not great. The low energy specimens had higher peak loads, and they had the larger absolute discrepancies between average drop tower 'tup energies' and average pendulum 'dial energies' when compared to the high energy specimens -- just as predicted by Bluhm years ago.

4.2.3 Test of the Theory

It is tempting to explain the discrepancy between the drop tower and pendulum Charpy tests in terms of tup and anvil compliance, but since it is just possible that crosshead friction during the event rises in a random manner and since tup response may vary with strain rate, it is more correct to put such an explanation to a direct experimental test.

The total system compliance in either test must equal the displacement at general yield divided by the general yield load. These quantities can be readily calculated from the data collected during the experimental part of the work, but since the sampling rate used for the low energy specimens was about six times higher than that used for the high energy specimens, yielding a higher horizontal resolution for the load-time plots, the most accurate determinations of general yield come from the data collected from the former rather than the latter. Therefore, the test of the theory will consist of a comparison of the compliance and general yield load information for the pendulum and drop tower machines, but only results for low energy specimens will be considered.

Tables 3-23(A) and 3-23(B) contain the necessary information. Note that the total system compliance of the drop tower is <u>lower</u> than that of the pendulum. Low system compliance could not possibly explain high absorbed energy. Hence, system compliance does <u>not</u> explain the discrepancy. Note also that the general yield load in the pendulum averages 7290 lb while the general yield load in the drop tower averages 9353 lb. From beam theory we can derive:

$$\text{Dynamic Yield Stress} = (33.3\ \text{in}^{-2}) * \text{General Yield Load} \quad [58] \qquad (4\text{-}11)$$

Hence, the general yield load is directly proportional to the yield stress which must be the same in both tests. The only way to resolve the conflict is to conclude that the drop tower tup calibration is wrong. Because of the high degree of confidence in the static calibration of the drop tower tup, the only possible conclusion is that the drop tower tup responds differently at dynamic strain rates than it does at static rates. In other words, even though the same is not true of the pendulum Charpy tup, the drop tower Charpy tup does require dynamic calibration.

Ireland has suggested that the correct way to perform dynamic tup calibration is to apply a known impulse or energy to a tup and to adjust the calibration curve until the calculated energy or impulse matches the standard energy or impulse. [59]. The Augland equation shows that total absorbed energy is a function of impulse. Hence, dynamic tup calibration by energy matching is equivalent to dynamic tup calibration by impulse matching. In essence, that is what was done in Table 22(B). The calibration curve from Table 3-22(B) was used to recompute the compliance information, and Table 3-23(C) is the result. The average compliance in Table 3-23(C) is higher than the average compliance in Table 3-23(A) which should have made the 'tup energy' higher, but did not. Furthermore, the average general yield load in Table 3-23(C) is 5845 which does not agree with 7290, the general yield load from Table 3-23(A).

The energy (or impulse) matching dynamic tup calibration approach implicitly assumes that the only causes of tup acceleration are the acceleration of gravity and the load applied to the tup. If such is not the case, the method must fail. To be specific, consider the ASTIR reports in Figures 3-9(A) - 3-9(U). Note that the event length of low energy specimens tested in the pendulum machine was 187.2 μsec while for the drop tower the event length was 193.3 μsec. For the areas under the drop tower load time curves to match corresponding pendulum areas, the curve had to be adjusted to produce erroneously low load values.

Hence excessive elastic energy storeage or bearings which are not practically frictionless or do not accurately control tup motion can cause an energy or impulse matching dynamic tup calibration approach to fail.

An alternate procedure for dynamic tup calibration is as follows. Generate several homogeneous population impact specimens and test them dynamically using a tup whose dynamic behavior is well calibrated. From the results, assign a general yield load and standard error to each population. Test a statistically significant quantity of each population dynamically using a tup of unknown dynamic response. Apply the usual curve fitting procedures to establish a calibration curve for the unknown tup.

A linear calibration function which nearly matches the general yield load in the drop tower test to that in the pendulum test was calculated. (In essence, this is a simplified version of the procedures described in the last paragraph.) The results are shown in Table 3-23(D). Student's t ratio for the difference between the average of the general yield load in Tables 3-23(A) and 3-23(D) is 0.243 with 6 degrees of freedom which clearly shows that the difference is insignificant. The F ratio for the same comparison is 3.35 with 4 and 5 degrees of freedom, suggesting that the drop tower Charpy test is a less precise means of measuring dynamic yield strength than the pendulum Charpy test. (The likelihood of being wrong, however, is a little greater than 10 %.)

Table 3-24 shows the drop tower 'tup energies' calculated with the calibration curve from Table 3-23(D). The t ratios for the comparison of the pendulum 'dial energy' with the 'tup energies' in Table 3-24 are 1.05 with 6 degrees of freedom for the high energy specimens and 2.98 with 4 degrees of freedom for the low energy specimens. These results are not significant for the high energy specimens and they are significant, but not highly so (likelihood of being wrong between 2 and 5 percent), for the low energy specimens.

If several general yield loads had been used for the calibration reported in Table 3-23(D), the drop tower 'tup energy' results might have been in still better agreement to the pendulum 'tup energies.'

5. CONCLUSIONS

1. The drop tower Charpy test as used in this study is significantly different from the ASTM E-23 approved pendulum Charpy test.

2. The discrepancy between the loads and energies calculated in the pendulum and drop tower tests were significant.

   a. In this study, a static calibration of the drop tower tup resulted in a discrepancy of +28% in load compared to the results obtained from the pendulum tests.

   b. A dynamic calibration of the drop tower tup based on energy standards produced a load discrepancy of -25% compared to the results obtained from the pendulum tests.

   c. A dynamic calibration of the drop tower based upon the assumption of linear tup response and the use of one dynamic load standard produced good agreement in load compared to the results obtained from the pendulum test, but energy results which could be rejected by use of Student's t test or the requirements of ASTM E-23.

3. The cause of the energy and load discrepancies between the pendulum and drop tower versions of the Charpy test is the dynamic response of the drop tower Charpy tup.

4. The results of this study do not eliminate the possibility that dynamic load standards could be used to calibrate a drop tower Charpy test so that it could be used to obtain impact energies equivalent to ASTM E-23 energies and accurate quantitative load information.

5. The results of this study show that the drop tower Charpy test gives energy results which have higher variance (i.e., less precision) than the pendulum Charpy test absorbed energy.

6. The following causes for high drop tower energy variance have been eliminated:

    a. Numerical method;

    b. System dimensions and physical constants;

    c. Tup calibration; and

    d. System compliance.

7. The 0^{th} order integration method or rectangle rule is adequate; more sophisticated techniques are unnecessary.

8. Depending on tup design, nonlinear or piecewise linear tup calibration curves may be required to yield acceptably accurate load and energy calculations.

9. There is no significant problem with the 'tup energy' calculated in the instrumented pendulum Charpy test. It can be used interchangeably with the pendulum Charpy test 'dial energy' and the intermediate results can be used to calculate material properties, providing: care has been taken to ensure proper calibration and adequately small granularity in the recorded results and the correct algorithms are used in the calculations.

6. FUTURE WORK

6.1 Dynamic Calibration of Tups

While this work shows statistically that it is highly likely that instrumented tups respond differently dynamically and statically, the following remains to be done:

1. A first principles method of dynamic calibration;
2. Investigation of the cause(s) of dynamic response;
3. Investigation of the cause(s) leading to failure of energy matching dynamic calibration in this study.

6.1.1 First Principles Dynamic Tup Calibration

Static calibration of tups and other load cells is based on Newton's law of gravitation, F=mg. In other words, the primary standard for force is an unaccelerated weight in a known local gravitational field. Such an approach in a dynamic, and hence inherently accelerated situation, is, of course, impossible.

An alternative would be to use the equation: F=ma. If a tup attached to a rigid pendulum or crosshead of known effective mass is accelerated by an impact force varying from zero to the rated load of the tup, the force can be calculated as effective mass multiplied by acceleration. Repeated tests could be used to provide replication for statistical analysis.

6.1.1.1 Ronchi Gratings and Shaft Encoders

The discussion in section 4.1.3 shows that measurements of position must be very accurate if useful acceleration measurements are to be made. OGC's stroboscopic equipment is not adequate. However, it seems highly likely that velocity measurement by the Ronchi grating system described in section 1.4.1.2.2.2 or by shaft encoders could provide sufficiently accurate velocity measurements to make accurate acceleration determinations. If such velocity determinations could be synchronized with corresponding tup output determinations, a first principles tup calibration could be done.

6.1.1.2 The Need for Dependable Bearings

Section 6.1.3 below suggests that during impact the drop tower crosshead experiences a frictional force with the linear bearings which hold it in place. Such frictional forces would decelerate the crosshead and, in a dynamic tup calibration based on F=ma, make the force applied by the specimen to the tup appear to be erroneously high, thus distorting the tup calibration. The problem could be avoided in one of two ways:

1. using a pendulum with removable tups; or
2. using a drop tower with superior linear bearings.

6.1.1.2.1 Pendulum with Removable Tups

There is every reason to suppose that there is no significant frictional component to the forces decelerating the pendulum in E-23 type Charpy machines. Therefore, first principles tup calibration could be done if a pendulum impact machine with the following attributes were used for the calibration. The machine's bearings would be of equal or better quality than those used in E-23 impact machines. Its pendulum would be as stiff or stiffer than those currently on E-23 pendulum impact machines. The machine would be equipped with a shaft encoder or Ronchi grating velocity measurement apparatus, and would have provisions whereby it would be easy both to replace anvils and mount a variety of tup designs. The design would also provide for attachment of weights on the pendulum to allow adjustment of the center of percussion so that it could always correspond to the center of strike. The tup mount on the pendulum and the anvils would have to be designed to prevent impact between the anvils and the pendulum-tup combination.

6.1.1.2.2 Drop Tower with Superior Linear Bearings

A drop tower could be constructed similar to the current OGC drop tower apparatus but with square instead of circular guide bars. On one end, the crosshead would have a rectangular hole whose longer sides would be perpendicular to the line between the guide bars. Along each of these sides would be two roller bearings at the top and bottom of the hole in the crosshead, spaced so as to contact the guide bars with low friction, thus accurately controlling the position of the crosshead in the direction parallel to a line between the guide bars. The short ends of the rectangular hole would not touch the guide bars. At the opposite end of the crosshead would be a second rectangular hole of the same geometry as the first, but with its long sides perpendicular to those on the opposite end. The roller bearings in the second hole would thus control crosshead position in the direction perpendicular to the line between the guide bars. The crosshead would be weighted or have balancing holes so as to very accurately position its center of gravity on the vertical line midway between guide bars and halfway between the front and back of the crosshead.

This arrangement should accurately control the position of the crosshead while imposing very little friction on it during impact loading.

6.1.2 Causes of Dynamic Tup Response

Several causes of dynamic tup response have been demonstrated to be inoperative in this study. (See section 4.2.2.2.4.2.) The only remaining cause would be dynamic strain gauge response differing from static strain gauge response. (This could possibly be due to mounting technique rather than inherent strain gauge response.) It would be useful to explore dynamic strain gauge response experimentally.

6.1.2.1 Direct Methods of Strain Measurement

Such an experiment would not be impossible: a Ronchi grating could be ruled onto the surface of a metal plate. The strain gauge would be attached to the same surface and the Ronchi grating would be observed through a transparent Ronchi grating with the same spacing as the first Ronchi grating using a photocell. The metal would be held rigidly at one end and loaded dynamically at the other. Both the spacing and position of the Ronchi grating on the steel plate would change with strain so that photocell output would be a complicated function of plate strain, but in principle the photocell output would measure strain while the strain gauge resistance was simultaneously measured.

Direct optical measurement of plate strain could also be done through the use of a diffraction grating. A diffraction grating would be ruled on the steel with its lines perpendicular to the direction of the load. A well collimated beam of monochromatic light (a laser beam, for example,) would be reflected at an oblique angle from the diffraction grating. The angle between the plate and the reflected beam is a function of the grating spacing and can be measured electro-optically. This procedure leads to a direct measurement of plate strain which could be carried out simultaneously with strain gauge resistance measurements.

From simultaneous strain and strain gauge resistance curves, calibration curves for the strain gauge could be made for a variety of strain rates and compared.

6.1.2.2 Comparison of Strain Gauges and Gluing Techniques

It is probable that some strain gauges are strain-rate insensitive while others are highly strain-rate sensitive. It is also likely that some glues and curing techniques produce bonds which might transmit strain to strain gauges in a strain-rate sensitive way. The above technique could be used to compare strain gauges, glues and curing techniques; several strain gauges could be glued to the same ruled plate and the results compared with a variety of strain rates.

6.1.3 Failure of Dynamic Tup Calibration by Energy Matching

The discrepancies between Tables 3-23(A) and 3-23(D) might be explained as follows: the linear bearings used in this study were simply loose brass bushings in the crosshead. As the crosshead falls, it is quite possible for it to wander. Hence it is not unreasonable for the tup to strike the specimen in such a way that the load might be concentrated one eighth inch from a line, passing vertically through the crosshead's center of gravity. Since load generated by specimens in this study was up to 8000 lb, such a strike would produce about 1000 in-lb of torque. The torque could rotate the crosshead so that it would apply a horizontal load to the guide bars and hence would experience a frictional load. This load might account for the 6 μsec slowing in drop tower events as compared to the pendulum events.

Such an explanation is consistent with the results in Table 3-23(D). If the torque friction model is correct, the following logic would explain the data shown in Table 3-23(D): in some drops, the tup would strike the specimen so that the impact load was centered with respect to the crosshead. No torque would occur and hence no friction. The event would absorb the same amount of energy as an E-23 Charpy test of the same material. Specimen L11 may be an example of such a drop. In some drops, the tup would be quite far from center. Presumably, the torque and friction would be maximized. ASTIR, not calculating the frictional effect on displacement, would calculate an erroneously high energy. Specimen L9 might be an example of such a drop.

There are several possible approaches to an experimental investigation of the above speculation:

1. metrological examination of low hardness specimens;
2. physical examination of the apparatus;
3. high speed photos;
4. superior linear bearings.

6.1.3.1 Metrological Examination of Low Hardness Specimens

The brinneling on low hardness specimens might be examined to determine just how far off-center the impact can be. This approach would be complicated by the fact that the brinelling marks on the bars are not simple rectangles initially and become distorted by plasticity and by the anvils as the tup pushes the specimen through them. Nonetheless, if the specimen ends were carefully ground flat, smooth, parallel to one another and perpendicular to the specimen sides so that they could be used as datum planes for metrological analysis, some headway might be made in this way -- especially if a large number of specimens were tested and statistically compared to pendulum test results.

6.1.3.2 Physical Examination of the Apparatus

The crosshead would be lowered to the level at which the tup encounters the specimen. The distances that the tup can move horizontally both parallel and perpendicular to the line between the centers of the guide bars would be carefully measured.

The crosshead would be demounted from the guide bars and the bearings would be examined using mirrors or a borescope in an effort to find the wear and impact damage that would have to accompany the hypothesized friction. The exact center of gravity of the crosshead would be carefully determined by balancing the crosshead on knife edges. This would be compared with the position of the center of the tup when mounted in the bottom of the crosshead.

6.1.3.3 High Speed Photos

A darkened background with light colored grid lines could be placed behind the drop tower, and the crosshead could be photographed by high speed photography as the tup impacts a specimen. The motion of the tup during an impact event could thus be quantified, but since a low energy event takes about 200 μsec during which time the supposed tup motion must occur, the photographic system would have to be able to make several photographs in this very small amount of time.

6.1.3.4 Superior Linear Bearings

If the system described in section 6.1.1.2.2 were constructed and the experiment in this dissertation were repeated, the new results should have much smaller variance if, indeed, friction caused by the looseness of the present bearings is the cause of high load and energy variance in drop tower testing.

6.2 Amount of Data Required for Accurate Absorbed Energy Measurement

The pendulum part of this work shows that if 80 data points are collected, an accurate absorbed energy can be calculated. However, the minimum number of points required for such a calibration is unknown. Repeating the pendulum part of this work with slower and slower sampling rates should make it possible to determine the minimum number of data points necessary for an accurate determination.

6.3 Rewrite of ASTIR in Turbopascal

RC Electronics now offers a version of COMPUTERSCOPE which runs under MS-DOS in IBM-PC Compatible computers. ASTIR could be rewritten in Turbopascal. The problems of file access and fast graphics would resurface, but once they were overcome the advantages of faster sampling rates and more data gathered would more than compensate for the difficulty.

REFERENCES

1. Charpy, G.; "On Testing Metals by the Bending of Notched Bars;" Interational Journal of Fracture 25 (1984); Martinus Nijoff Publishers, the Hague, Netherlands; 1984; p. 295. Translated from "Memoires de la Societe des Ingenieurs Civils de France;" October 1904; pp. 466 to 482.

2. Izod, E. G.; "Testing Brittleness of Steel;" Engineering; September 25, 1903; Vol.25, p. 432.

3. ASTM Committee E-28; "Notched Bar Impact Testing of Metallic Materials" 1986 Annual Book of ASTM Standards: Section 3 Metals Test Methods and Analytic Procedures Volume 03.01 Metals - Mechanical Testing; Elevated and Low-Temperature Tests; Publ: American Society for Testing and Materials; Philadelphia, PA; 1986; p. 229-253.

4. Charpy, G.; Op. Cit.; p. 287.

5. Millane, J.; Personal Telephone Conversation with Stevens, W.; Tinius-Olsen Corporation; (215) 675-7100.

6. Fahey, Norbert H.; "Quality Assurance Pamphlet - Mechanical Testing - The Charpy V-Notch Impact Test;" AMXMR-P-702-104; Army Materials and Mechanics Research Center; Watertown Massachusetts; 1 December 1969; pp. 2-3.

7. Landry, Arthur F.; "The Charpy Calibration Program: Its History and Importance; What does the Charpy Test Really Tell Us?" Proceedings of a Symposium Held at the Annual Meeting of the American Institute of Mining, Metallurgical, and Petroleum Engineers; Denver, Colorado; February 27 and 28, 1978; Eds: Rosenfield, A. R.; Gegel, H. L.; Hasson, D. F., Rath, B. B., Stephens, J. R.; American Society for Metals; Metals Park, Ohio; 1978; pp. 1-3.

8. Fahey, Norbert H.; "The Charpy Impact Test - Its Accuracy and Factors Affecting Test Results;" Impact Testing of Metals: ASTM STP 466; American Society for Testing and Materials; 1970; pp. 76-78.

9. Izod, E. G.; Op. Cit.; p. 431.

10. Bluhm, Joseph I.; "The Influence of Pendulum Flexibilities on Impact Energy Measurements;" Symposium on Impact Testing: ASTM STP 176; American Society for Testing and Materials; Philadelphia, Pa.; 1956; pp. 88-99.

11. ASTM Committee E-28; "Dynamic Tear Testing of Metallic Materials;" 1986 Annual Book of ASTM Standards: Section 3 Metals Test Methods and Analytic Procedures Volume 03.01 Metals - Mechanical Testing; Elevated and Low-Temperature Tests; American Society for Testing and Materials; Philadelphia, PA; 1986; p. 645.

12. Heberling, T. G. and Selby, G. E.; "Drop-weight Tear Test-Effect of Variables on Test Results;" Impact Testing of Metals: ASTM STP 466; American Society for Testing & Materials; 1970; p. 228.

13. Eiber, R. J., and Maxey, W. A.; "Correlation of Full-Scale Tests with Laboratory Tests;" Symposium on Line-Pipe Research, A Progress Report on Project NG-10 by the Pipeline Research Committee at Battelle Memorial Institute; Dalles, Texas, November 17-18, 1965; American Gas Association, Inc.; 605 Third Avenue, New York, New York 10016; March 1966; pp. 84-88.

14. ASTM Committee E-28; "Notched Bar Impact Testing of Metallic Materials" 1986 Annual Book of ASTM Standards: Section 3 Metals Test Methods and Analytic Procedures Volume 03.01 Metals - Mechanical Testing; Elevated and Low-Temperature Tests; Publ: American Society for Testing and Materials; Philadelphia, PA; 1986; p. 229.

15 Bluhm; Loc. Cit.

16. For example, Dynatup, 8250 Drop Tower with dynatup 10,000 lb Instrumented Tup, Charpy Tup insert and anvil machined to match the requirements of ASTM E-23.

17. Izod, E. G.; Op. Cit.; p. 431.

18. Bluhm; Op. Cit.; pp. 87-89.

19. Lange, E. A. and Loss, F. J.; "Dynamic Tear Energy - A Practical Performance Criterion for Fracture Resistance;" Impact Testing of Metals: ASTM STP 466; American Society for Testing and Materials; 1970; pp. 247-248.

20. Heberling, T. G. and Selby, G. E.; Op. Cit.; p. 228.

21. Saxton, H. J.; Ireland, D. R.; and Servor W. L.; "Analysis and Control of Inertial Effects During Instrumented Impact Testing;" Instrumented Impact Testing: ASTM STP 563; American Society for Testing and Materials; 1974; pp. 50-73.

22. Ireland, D. R.; "Procedures and Problems Associated with Reliable Control of the Instrumented Impact Test;" Instrumented Impact Testing: ASTM STP 563; American Society for Testing and Materials; 1974; pp. 11-15.

23. Bluhm; Op. Cit.; p. 92.

24. Tardif, P. P. and Marquis, H.; "Force Measurements in the Charpy Impact Test;" Canadian Metallurgical Quarterly, Vol. 2; No. 4; Oct-Dec; 1963.

25. Grumbach, M.; Prudhomme, M.; Sanz, G.; "Essais de Rupture Fragile Aveg Enregistrement;" Revue de Metallurgie; Avril 1969; p. 272.

26. Radon, J. C. and Turner, C. E.; "Fracture Toughness Measurements by Instrumented Impact Tests;" Engineering Fracture Mechanics; Vol. 1; 1969; pp. 411-414.

27. Ireland, D. R.; "Procedures and Problems Associated with Reliable Control of the Instrumented Impact Test;" Instrumented Impact Testing; ASTM STP 563; American Society for Testing and Materials; 1974; pp. 4-5.

28. Ewing, A. and Raymond, L.; "Instrumented Impact Testing of Titanium Alloys;" Instrumented Impact Testing; ASTM STP 563; American Society for Testing and Materials; 1974; pp. 199-200.

29. Tardif, H.P. and Marquis, H.; "Force Measurements in the Charpy Impact Test;" Canadian Metallurgical Quarterly; Vol. 2; No. 4; Oct.-Dec. 1963; p. 376.

30. Working Group on Dynamic Toughness Testing PVRC/MPC Task Group on Fracture Toughness Properties for Nuclear Components; "Instrumented Precracked Charpy Testing for Medium Strength Nuclear Pressure Vessel Steels Part 1 and Part 2;" PVRC/MPC Task Group on Fracture Toughness Properties for Nuclear Components Final Report; Welding Research Council; 1977; p. 211.

31. Ewing, A. and Raymond, L.; "Instrumented Impact Testing of Titanium Alloys;" Instrumented Impact Testing: ASTM STP 563; American Society for Testing and Materials; 1974; p. 200.

32. Radon, J. C. and Tunner, C. E.; "Fracture Toughness Measurements by Instrumented Impact Test;" Engineering Fracture Mechanics; 1969; Vol. 1; Pergamon Press; Great Britain; p. 413.

33. Ireland, D.; Personal Communication.

34. Palmer, C. Harvey Jr.; Personal Communication dated March 7, 1986.

35. Augland, B.; "Fracture Toughness and the Charpy V Notch Test;" Correspondence; British Welding Journal; July 1962; p. 435.

36. Grumbach, M. "L'Essai Charpy Avec Enregistrement'" Soudages et Techniques Conneies; Nov.-Dec.; 1967; p. 494.

37. Grumbach, M.; Prudhomme, M.; Sanz, G.; "Essais de Rupture Fragile Ave Enregistrement;" Revue de Metallurgie; April 1969; p. 280.

38. ASTM Committee E24; "Dynamic Tear Testing of Metallic Materials;" 1986 Annual Book of ASTM Standards: Section 3 Metals Test Methods and Analytic Procedures Volume 03.01 Metals - Mechanical Testing; Elevated and Low-Temperature Tests; Publ: American Society for Testing and Materials; Philadelphia, PA; 1986; p. 647.

39. ASTM Committee E-28; "Notched Bar Impact Testing of Metallic Materials" 1986 Annual Book of ASTM Standards: Section 3 Metals Test Methods and Analytic Procedures Volume 03.01 Metals - Mechanical Testing; Elevated and Low-Temperature Tests; Publ: American Society for Testing and Materials; Philadelphia, PA; 1986; p. 232.

40. ASTM Committee E-28; "Dynamic Tear Testing of Metallic Materials;" 1986 Annual Book of ASTM Standards: Section 3 Metals Test Methods and Analytic Procedures Volume 03.01 Metals - Mechanical Testing; Elevated and Low-Temperature Tests; Publ: American Society for Testing and Materials; Philadelphia, PA; 1986; p. 647.

41. Symon, K. R.; Mechanics; 3rd Edition; Addison-Wesley; Reading, Mass.; 1971; p. 210.

42. Symon; Op. Cit.; p. 216.

43. Symon; Op. Cit.; p. 218.

44. ASTM Committee E-28; "Notched Bar Impact Testing of Metallic Materials" 1986 Annual Book of ASTM Standards: Section 3 Metals Test Methods and Analytic Procedures Volume 03.01 Metals - Mechanical Testing; Elevated and Low-Temperature Tests; Publ: American Society for Testing and Materials; Philadelphia, PA; 1986; p. 201.

45. Ireland, D. R.; "Procedures and Problems Associated with Reliable Control of the Instrumented Impact Test;" Instrumented Impact Testing; ASTM STP 563; American Society for Testing and Materials; 1974; p. 6.

46. Ireland, D. R.; "Procedures and Problems Associated with Reliable Control of the Instrumented Impact Test;" Instrumented Impact Testing; ASTM STP 563; American Society for Testing and Materials; 1974; pp. 8-11.

47. Davies, O. L. and Goldsmith, P. L.; Statistical Methods in Research and Production; Longman Group LTD; NY; 4th Edition; 1980; pp. 75-56.

48. Davies, O. L. and Goldsmith, P. L.; Op. Cit.; pp. 61-62.

49. ASTM Committee E-28; Op. Cit.; p. 203.

50. Ireland, D. R.; Personal Communication.

51. Nicholas, T.; "Dynamic Tensile Testing of Structural Materials Using a Split Hopkinson Bar Apparatus;" AFWAL-TR-80-4053; Materials Laboratory; Air Force Wright Aeronautical Laboratories, Air Force Systems Command; Wright-Patterson Air Force Base; Ohio; 1980; pp. 21-40.

52. Kolsky; Stress Waves in Solids; Dover; 1963; p. 19.

53. Carlin; Ultrasonics; McGraw-Hill; 1949. (Note that the ratio of wavelength to frequency remains constant; hence speed of sound remains constant as frequency changes.)

54. Perry, C. C. and Lissner, H. R.; The Strain Gauge Primer; McGraw-Hill Book Company; New York; 1962; p. 82. Note also that on pp. 88-89 dynamic systems are designed with statically determined calibration factors.

55. Izod, E. G.; "Testing Brittleness of Steel;" Engineering (Sept. 25, 1903); Volume 25; p. 431.

56. Bluhm, Joseph I.; "Influences of Pendulum Flexibilities on Impact Energy Measurements;" Symposium on Impact Testing: ASTM STP 176; American Society for Testing and Materials; 1955; p. 92.

57. Bluhm, Joseph I.; "Influences of Pendulum Flexibilities on Impact Energy Measurements;" Symposium on Impact Testing: ASTM STP 176; American Society for Testing and Materials; 1955; pp. 89-92.

58. Server. W. L. and Ireland D. R.; "General Yielding of Notched Three-Point Bend Specimens;" TR 72-19 Effects Technology Inc.; Santa Barbara, Ca.; 1972; pp. 2-5.

59. Ireland, D. R.; Personal Communication; 1987.

60. Worth, D. and Lachner, P.; Beneath Apple DOS; Quality Software; Reseda, Ca; 1981; p. 3-1 and p. 4-8.

61. Apple Computer Inc.; Apple][Reference Manual; Apple Computer Inc.; Cupertino, Ca.; 1981; p. 21.

62. Forsythe, George E. and Moler, Cleve B.; Computer Solution of Linear Algebraic Systems; Prentice-Hall; New Jersey; 1967; p. 88.

63. Davies, O. L. and Goldsmith, P. L.; Op. Cit.; Chapters 7 and 8: Analysis of Variance; pp. 178-303; Chapter 4: F and t tests; pp. 58-93.

64. Forsythe, George E. and Moler, Cleve B.; Computer Solution of Linear Algebraic Systems; Prentice-Hall; New Jersey; 1967; pp. 28-118 and pp. 59-135, but especially pp. 68-69.

65. Forsythe, George E.; Malcom, Michael A.; and Moler, Cleve B.; Computer Methods for Mathematical Computations; Prentice-Hall; New Jersey; 1978; pp. 30-55.

66. Forsythe and Moler; Op. Cit.; p. 78.

APPENDIX A

THE ASTIR PROGRAM

A.1 INTRODUCTION

ASTIR -- Apple Scope to Impact Report -- is a computer program which enables a user familiar with the basic concepts of instrumented impact testing but not necessarily experienced in using computers to calculate results and generate an impact report from data he has already collected using the COMPUTERSCOPE Program (Formerly APPLE SCOPE written by Richard E. Renck at R. C. Electronics in Santa Barbara, California). The input data is 1024 bytes, each of which represents a voltage and 256 bytes which encode information about the status of the COMPUTERSCOPE program at the time of data storage. Each voltage, in turn, represents a force.

ASTIR runs on most versions of the Apple II computer, but it requires that there be two 5.25 inch floppy disk drives controlled by a single controller card in slot 6. Furthermore, one of those drives must be used to boot or reboot the system.

A.2 A WALK-THROUGH OF ASTIR

To start the ASTIR program, the user inserts the ASTIR turnkey disk in the boot disk drive and a data disk in drive 2 and turns the computer on or reboots it.

A.2.1 The Startup Menu

When the computer is turned on or rebooted, the appearance of the screen is as shown in Figure A-1. The first two lines are the title of this menu: ASTIR Method Choices. The next three lines are the body of the menu and represent the items that the user can change at this time. The final line is the prompt line which consists of the commands which ASTIR will accept at this time. To enter a command, the user types the letter associated with the desired command (for example A); the effect of the command is briefly described in parenthesis.

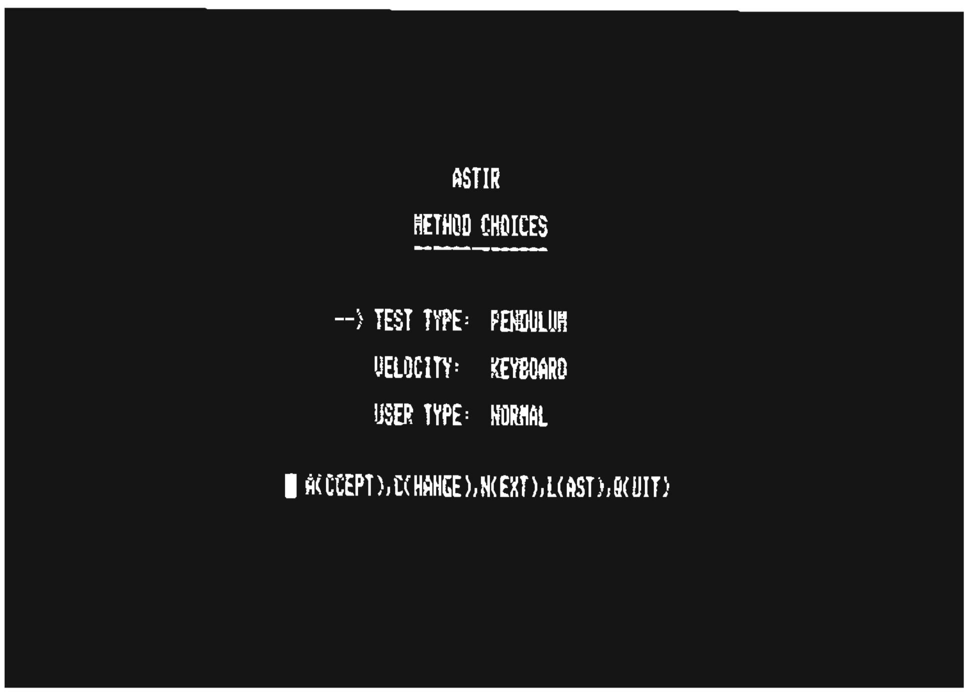


Figure A-1. The Startup Menu

In the prompt line of the Start-up Menu, there are five commands: A, C, N, L and Q. Q causes the ASTIR program to terminate and leaves the user in the Apple version of the UCSD p-machine operating system. N causes the arrow next to the choices to move down until it reaches the last choice, at which point the arrow will appear at the top of the screen. L does the reverse of N, moving the arrow up until it reaches the first choice. It will then move back to the last choice. C changes the choice next to the arrow. The program always shows a choice (see the next paragraph for a list of the choices). A accepts the current choices and moves the user to the Data Selection Menu.

Here are the possible choices which can be made during the first menu by using the command C:

Test Type:	Pendulum, Drop Tower or Slow Bend
Velocity:	Keyboard or Curve
User Type:	Normal, Pro or Super

The first choice describes the basic type of test apparatus, and the user can be assumed to be familiar with the possibilities shown. The second choice allows the user to specify if he plans to input the initial velocity by hand or rather obtain the velocity from existing input data that is already recorded on the data disc. The third choice describes the user privileges. Pro or Super users can customize the state transitional diagram from inside the program. In other words, they can change the order in which the program performs its operations. In addition, the Super user sees diagnostics which have to do with debugging the current version of the program. Normal users must use the normal order of operation and do not see any debugging diagnostics. An intelligent choice of Pro options would be the fastest way to process data. The use of the Normal user choice would be the easiest way to use ASTIR. (Pro customization of the state transitional diagram is not yet implemented.)

There are no diagnostics associated with the Startup Menu, but when the user types any command, it appears in inverse next to the prompt line. If the command does not exist in that prompt line, nothing happens. These precautions are implemented in each of the prompt lines.

A.2.2 Choosing the Recorded Curve - the Data Selection Menu

After the user has accepted the Startup Menu by pressing the A key, the Data Selection Menu, shown in Figure A-2, appears. There is no title; the body consists of up to a screenful of available file names from the DOS 3.3 data disc. The prompt line shows that there are only two commands: S and C.

If the user types C, the program does one of three things depending on the number of files on the data disc. If there is more than one more screenful of files in the catalog, ASTIR shows another screenful. If there are files in the catalog that have not been shown, but less than a screenful, ASTIR shows all of the remaining files. If ASTIR has already shown all the file names, ASTIR goes back to the beginning of the catalog and tries to show a screenful. In each case, the new body is shown with the original prompt line. If the user types S, the prompt line changes and ASTIR asks the user to select a curve by typing the number which appears in brackets next to the desired curve. The number which is typed will appear in the brackets next to the prompt line. (See Figure A-2). After the user selects a curve by typing a number and pressing return, ASTIR shows the name of the file selected and asks if it is acceptable. If the user types anything starting with N or n, ASTIR starts the Data Selection Menu over. If the user types anything beginning with Y or y, ASTIR loads the curve from the COMPUTERSCOPE data disc, goes on to the Feature Identification Menu, and the user is committed to completing the ASTIR program, rebooting or turning off the computer.

The only diagnostic at this stage is that if the user tries to accept a file which does not share certain salient features with COMPUTERSCOPE files, ASTIR warns the user that he has not selected a data file and returns the user to the beginning of the Data Selection Menu.

A.2.3 The Feature Identification Menu

During an impact event, several things may or may not happen. Among them are that the event starts; general yielding may occur; a maximum load will occur; fast fracture may commence; and the event will end. The Feature Identification Menu (Figure A-3) enables the user to select the times when these things happened. The user must select these features since otherwise ASTIR would have to perform sophisticated pattern recognition and hence it would be larger slower and less reliable than it is.

Figure A-2. The Data Selection Menu

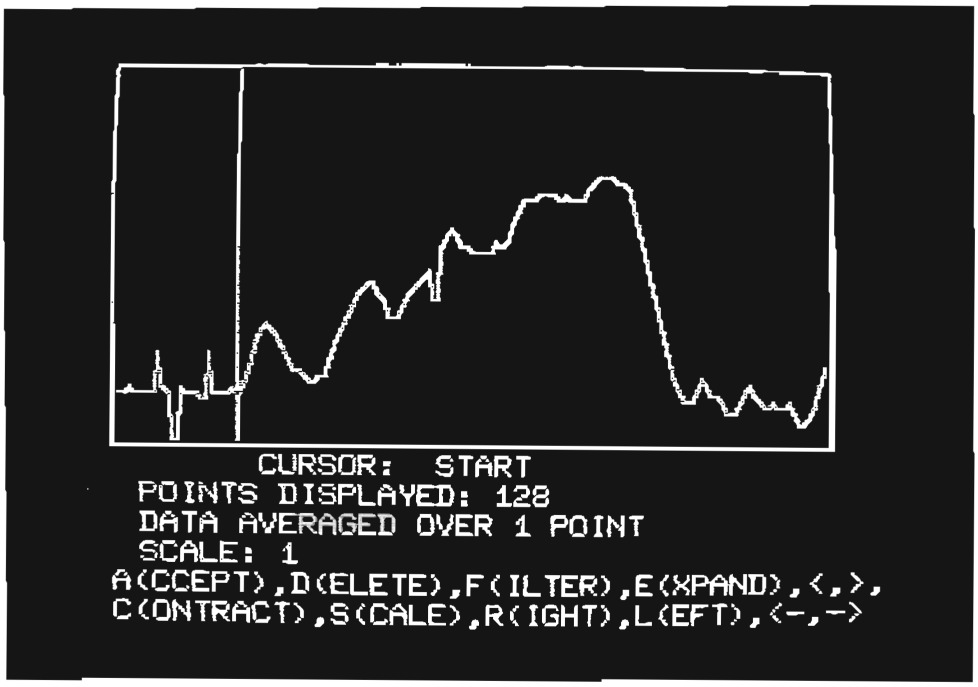


Figure A-3. The Feature Identification Menu

The Feature Selection Menu has no title. Its body consists of a graph of the data, below which is the name of the feature (cursor) currently under consideration, the number of points currently displayed, point averaging data, and a number called the scaling factor which has to do with the degree to which the vertical scale is adjusted. The prompt line contains twelve commands: A, D, F, E, <, >, C, S, R, L, right arrow (->), and left arrow (<-). The name of the feature (cursor) under consideration is either one of the above-named features or the beginning or end of the range over which the background voltage level is to be averaged. The number of points displayed is an integer between 8 and 1024.

If the right or left arrow are pressed, the current cursor (indicated by a vertical line) will move one data point in the corresponding direction. If < or > are pressed the cursor will move sixteen data points to the right or left. If R or L is pressed the screen is redrawn with the cursor one half a current screen width in the appropriate direction if possible.

In previous menus, A accepted the whole menu. In the Feature Selection Menu, A accepts only the current cursor. ASTIR does not go on to the next section until all the cursors are either accepted of deleted (by pressing D). Accepting a cursor means that the appropriate feature has occurred at the current cursor location. Deleting the cursor means that the feature did not occur.

Commands E, C and S change the way the data is graphed. If E is pressed, ASTIR asks for an integer between 0 and 9 and ASTIR changes the number of points to be displayed to the current number of points displayed divided by two raised to the number chosen by the user. Then ASTIR redraws the screen centered about the current cursor position with the new number of points displayed. If C is pressed the opposite occurs. The number of points is increased by setting it equal to the old number of points times two raised to the input number. If S is pressed, ASTIR asks for a number between 0 and 9. The screen is redrawn centered about the current cursor position with the height of the curve equal to the byte at each location multiplied by two raised to the scale number minus 1. Note that the effects of E and C are cumulative but the effects of S are not. It is useful to input zero with one of the above options since that will enable fine tuning of the position of the center of the graph.

Pressing F allows the user to do filtering by point averaging. When F is pressed, ASTIR asks for an integer between 0 and 9. Each byte of the data is then replaced by the truncated average of the points from the input number before the current byte to the input number after the current byte. Then the newly computed curve is replotted centered about the current cursor with the current number of points and the current scale number displayed. If the data are filtered and then refiltered with the filtering number equal to zero, the original data is restored.

The only diagnostics at this point are that ASTIR will not allow the user to reject the start or end of event cursors and warns the user that such an event is impossible. (Clearly, there can never be an impact event which has no start or no end, folk songs and western music notwithstanding). ASTIR will also not allow the user to move any cursor beyond the beginning or end of the data or try to display more than 1024 points or less than 8 points. Furthermore, it will not let the scaling factor exceed 9 or be less than 0, but in these events no message is displayed.

A.2.4 The Calibration Data Menu

After the event start and finish cursors have been selected and the rest of the cursors either accepted or rejected, the Calibration Data Menu appears. (See Figure A-4). Its title is "Calibration Constants". Its body consists of the names of the calibration constants with their current values next to them to the right. The prompt line shows five commands: A, C, L, N and F.

The names and meanings of the calibration constants are:

Flag1: The width in inches of the first timing flag.

Flag2: The width in inches of the second timing flag.

Velterm: A number which, when added to the Flag widths, accounts for the peripheral vision effect in the infra- red detector.

Gee: The local acceleration of gravity in feet per second squared.

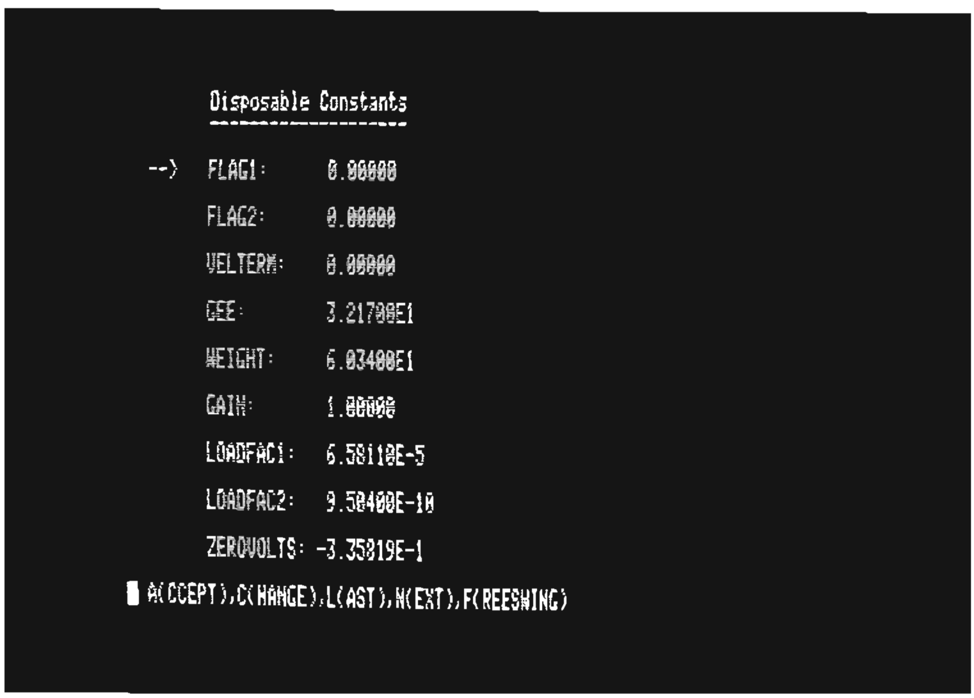


Figure A-4. The Calibration Data Menu

Weight: The effective weight in pounds of the pendulum or cross head.

Gain: A pure number which, when multiplied by the load calculated from the gain one calibration curve, yields the correct load when some gain factor other than "one" was set on the amplifier. (In point of fact, no such number exists or is even possible, although fairly good approximations can be made. In subsequent versions of ASTIR, Gain will be eliminated in favor of the use of various calibration curves.)

Loadfac1: The coefficient of the linear term in in the calibration curve. Its unit is volts per pound.

Loadfac2: The coefficient of the quadratic term in the calibration curve. Its unit is volts per pound squared.

Zerovolts: The Y intercept of the calibration curve. Its unit is volts.

Commands L and N work exactly as they do in the Startup Menu. When C is pressed, the current value of the constant is erased and an input cursor appears at the extreme left end of its old field. Whatever is typed into that field becomes the new value of the constant after return is pressed.

If F is pressed, the value of Zerovolts is changed to the average of the voltage encoded into the points from the start cursor to the finish cursor. F is useful if a freeswing (impact test with no specimen) has been performed to measure Zerovolts. Zerovolts will be calculated from the current curve if both the Zero Range cursors have been accepted and that calculated Zerovolts value will then be used in the calculation of load, displacement and tup energy. (There is currently a program bug here: The value of Zerovolts calculated from the curve IS used in the load, displacement and tup energy calculation, but it does not replace the value of Zerovolts in the Calibration Constants Menu, and the value from the menu is used to plot the zero line in the report graph).

When A is pressed, the Calibration Constants Menu is accepted and ASTIR proceeds to the Test Conditions Menu, shown in Figure A-5.

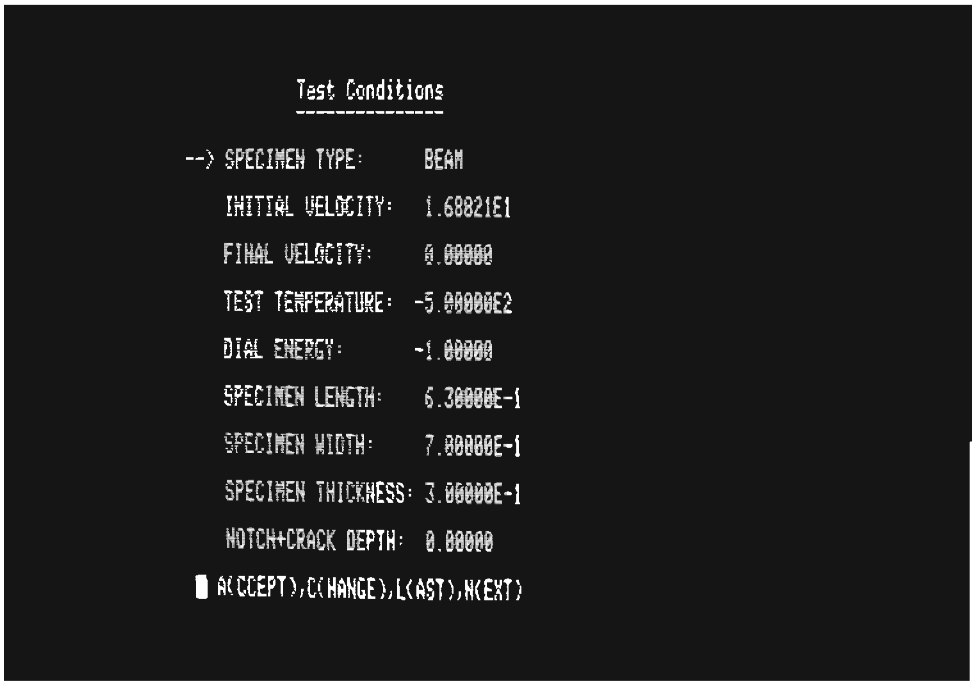


Figure A-5. The Test Conditions Menu

A.2.5 The Test Conditions Menu

The Test Conditions Menu is similar to the Calibration Constants Menu except for the fact that it has no Z command and it has different items:

Specimen Type:	The general geometry of the specimen. When C is pressed, Specimen Type changes between beam and cantilever.
Initial Velocity:	The initial velocity in feet per second. It is used only if the keyboard velocity option is used.
Final Velocity:	It may be that the user has some way of knowing the final velocity without using the recorded curve. (Such possibilities are discussed in the Introduction.) If so and if the user has taken the keyboard velocity option, it may be recorded here. If both velocities are known, ASTIR will calculate the flag energy as the difference between the initial and final kinetic energies. ASTIR expects the user to input the final velocity in feet per second.
Test Temperature:	The specimen temperature in degrees Fahrenheit at the time of the impact. It must be added here since there is no other way for ASTIR to obtain it and it is necessary for a complete impact report.
Dial Energy:	It will, no doubt, be desired to compare the dial, flag, and tup energies. ASTIR cannot obtain the dial energy even if a valid dial energy is available unless the user records it. ASTIR expects the user to input the Dial Energy in foot-pounds.

The specimen dimensions:	These are necessary for calculation of the dynamic mechanical properties of the material from which the specimen was made. They are expected in inches.

A.2.6 The Report Menu

After the Test Conditions Menu has been accepted, ASTIR proceeds with its central task: calculation of fracture information from the input data. The calculations take a minute or two and then an auto-scaled graph of the load-time record is drawn on the high resolution graphics screen and a text report is printed on the eighty column text screen.

There is a prompt line at the bottom of both the graphics and text screens and they are different in only one command. The graphics screen has T in its prompt line and the text screen has G in its prompt line. In effect, the graphics and text screens are a single menu with the following commands in its prompt line: H, T, G, C, F, Q and no title. (See Figure A-6).

The body of the Report Menu consists of the following:

1. the load-time graph;

2. the number of data points from event start to event end;

3. the number of points over which the data was averaged to do digital filtering by the method of running averages;

4. the name of the file in which the data was found;

5. the test method (pendulum, drop tower or slow bend);

6. the test temperature;

7. the load at each of the features except event start;

8 the time from event start to each of the other features;

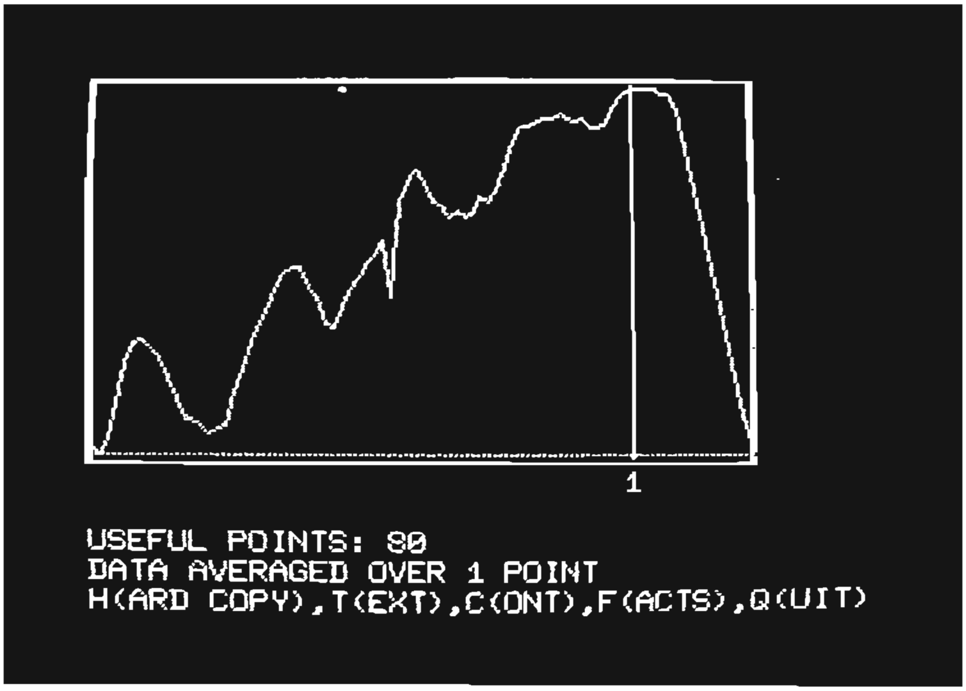
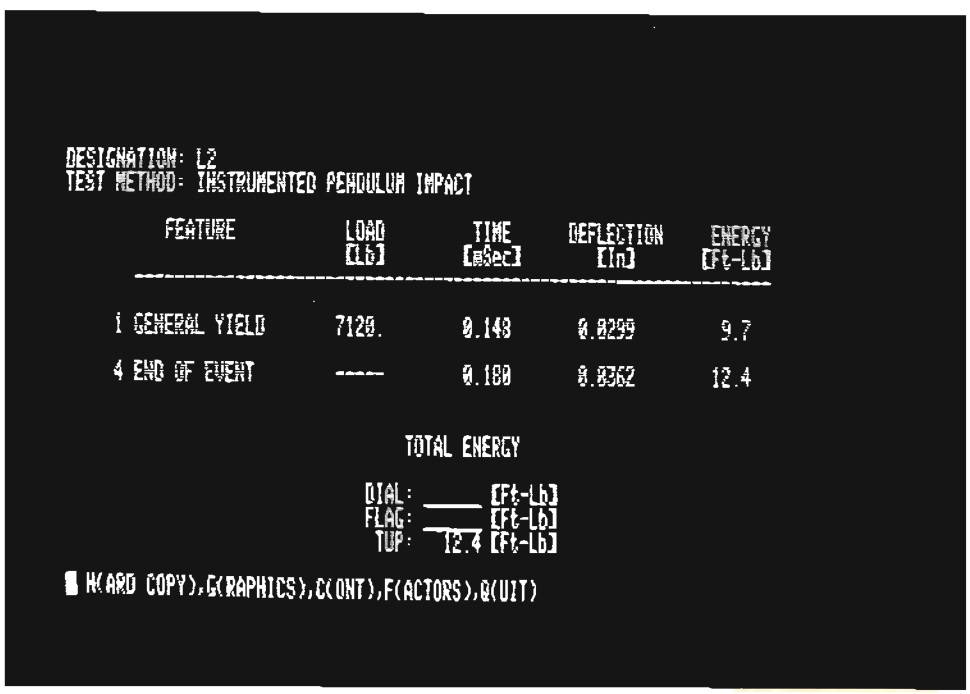


Figure A-6(A). The Report Menu - Graphics


DESIGNATION: L2
TEST METHOD: INSTRUMENTED PENDULUM IMPACT

FEATURE	LOAD [Lb]	TIME [mSec]	DEFLECTION [In]	ENERGY [Ft-Lb]
1 GENERAL YIELD	7120.	0.148	0.0299	9.7
4 END OF EVENT	----	0.180	0.0362	12.4

TOTAL ENERGY

DIAL: ____ [Ft-Lb]
FLAG: ____ [Ft-Lb]
TUP: 12.4 [Ft-Lb]

H(ARD COPY),G(RAPHICS),C(ONT),F(ACTORS),Q(UIT)

Figure A-6(B). The Report Menu - Text

9. the total deflection at each of the features except event start;

10. the energy dissipated at each of the features except event start;

11. the total energy absorbed as obtained by:
    a. reading the pendulum dial (if the test type is pendulum);
    b. subtracting the final kinetic energy from the initial kinetic energy (if the option to calculate velocities from the curve was taken and if all four flag cursors were accepted or if initial and final velocities were input in the Test Conditions Menu);
    c. calculating the energy from the load-time data as explained in the body of this report.

If H is pressed, a hard copy of both the graphics and text report is made. (Currently only the printer combination Grappler+ interface card and Epson printer are supported. It is planned to support the combination Apple IIgs printer port image writer in the future).

If G is pressed, the text display is replaced with the graphics display. If T is pressed, the graphics display is replaced with the text display. Thus, T and G can be used to toggle between text and graphics.

If C is pressed, ASTIR starts over from the beginning, allowing the user to continue by choosing a new curve. (In this case all input from the user and the data disk is discarded. The body of the program alone is kept in memory.)

If F is pressed, ASTIR goes back to the Calibration Data Menu and proceeds from that point. (In this case the original curve, the filtered curve and the information about the features are retained. Improper use of the F command could cause difficulties and ASTIR may be modified to solve the problem in the near future.)

Q causes the ASTIR program to terminate and leaves the user in the Apple version of the UCSD p-machine operating system.

A.3 ISSUES IN THE THE DEVELOPMENT OF ASTIR

A.3.1 Choice of Language

The only options available on the Apple II series computers at the time that ASTIR was started were 6502 Assembly language, Basic, Fortran, and Pascal. Basic, an interpreted language, would have been unacceptably slow in operation. Pascal was chosen because it was apparent that the task would be quite complicated and it was expected that it would not be possible to complete it without the use of a strongly typed language with substantial structure built into it.

A.3.2 Transfer of Data Files Between Two Operating Systems

While it is true that both the Apple version of the UCSD P-machine operating system, under which Apple Pascal runs and Apple DOS 3.3, under which COMPUTERSCOPE runs use five and one quarter inch floppy diskettes for storage and the same number of tracks and sectors, the exact organization by which the two do so is very different. This difference represents one of the thorniest problems in the development of ASTIR though a problem which is not directly related to metallurgy.

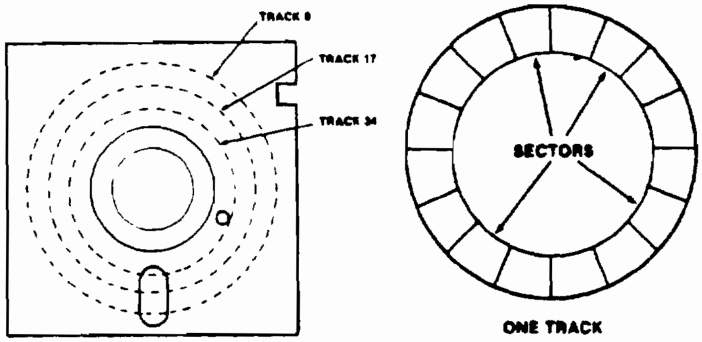
The way Apple DOS 3.3 does storage and recovery of information is illustrated in Figures A-7(A) and A-7(B). [60]. The Volume Table of Contents is read from track 17 sector 0 to find the address (track and sector) of the first catalog sector. The catalog sectors each contain the names, types, and locations of track sector lists of several files and the address of the next catalog sector. The last catalog sector contains a null catalog sector address.

Once a file is chosen, its track sector list is read to find its data sectors which are then read in order.

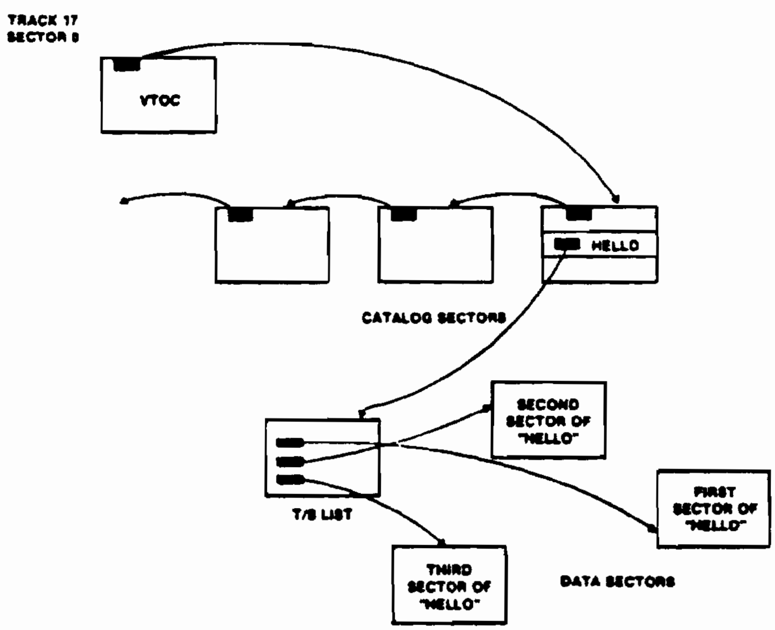
ASTIR takes advantage of a built in procedure in Apple Pascal called blockread which reads the information on two consecutive sectors into an array. A special subroutine called ASTIRREAD essentially emulates the action of DOS 3.3 in retrieving the stored data.

A.3.3 Memory Management

It was obvious from the start that ASTIR would be far too large to fit in memory on an Apple II+ or Apple IIe. The solution was to take advantage of a feature of Apple Pascal known as segment procedures to load and unload certain parts of the program from memory as needed.



A. The DOS 3.3 Disc Format



B. Finding the Data Sectors Under DOS 3.3

Figure A-7. Data Retrieval Under Apple DOS 3.3

A.3.4 Compilation

Compilation on any Apple II series computer is excruciating slow. Apple Pascal, however allows precompiled library units to be linked to separately compiled host programs. Use of this feature allowed ASTIR to be developed much more rapidly than might otherwise have been the case.

A.3.5 Graphics

Apple Pascal includes a very powerful graphics library providing for reversed colors, drawing of text on a high resolution graphics screen and point to point plotting but very slow speed. It was decided to use the Apple Pascal graphics library where speed was not of the essence and to provide a simple 6502 micro processor assembly language routine for point to point plotting so that drawing the curve which would otherwise take a great deal of time could be done reasonably quickly.

The Apple high resolution graphics screen is illustrated in Figure A-8. [61]. Note the extremely complicated way in which pixels are encoded. Each byte in the graphics area of memory represents seven pixels. The high bit of the byte is not displayed, the next highest bit represents the leftmost pixel, etc. In Figure A-8, the column row and box numbers are summed to find the memory location of each graphics byte.

Any point-to-point line drawing routine must find the address of the start and end point and turn on pixels between the start and end point so as to produce an image of a line or curve connecting them.

As stated in the first paragraph, it was desired to have an extremely fast point to point plotting routine but one which produced a graph which would be easy for the user to comprehend. The routine which was finally used for point-to-point plotting produces a compromise between these ideals.

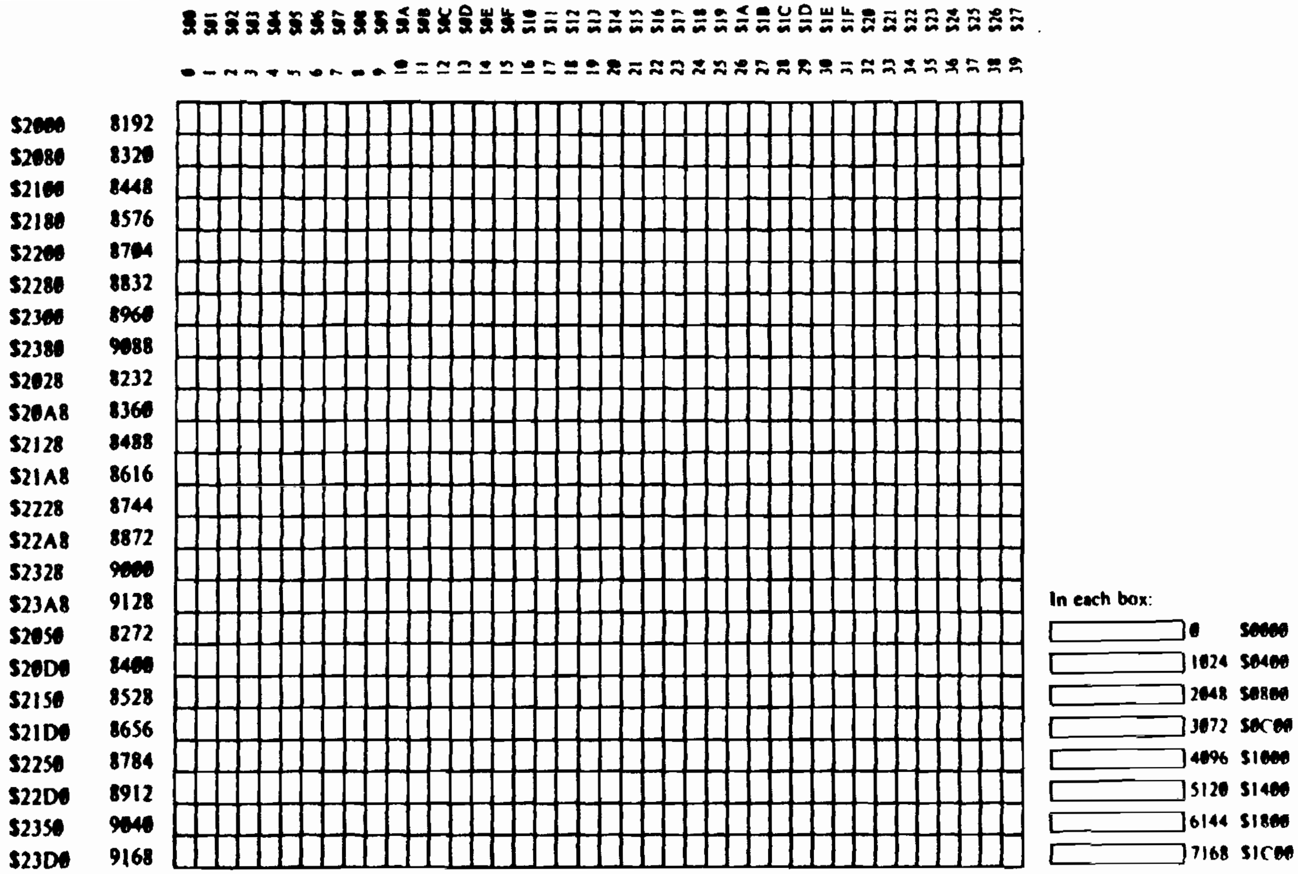


Figure A-8. The Apple II High Resolution Graphics Screen

A.4 DERIVATION OF NUMERICAL METHODS USED IN ASTIR FOR VELOCITY, DISPLACEMENT, AND ABSORBED ENERGY CALCULATIONS

The simplest method for determining velocity, displacement, and absorbed energy assumes that load is constant during each sampling interval. This is the procedure that was described in the body of the text. However, it is obvious that such is not actually the case. It is possible to use the load data from subsequent time intervals to improve the accuracy of the calculations.

Clearly load must be a continuous function of time. However, COMPUTER SCOPE, like all digital measurement methods, can measure load only at discrete intervals. By using a polynomial equation that is derived using "n" data points (discrete load measurements), it is possible to approximate the continuous nature of the load, expressing load as an $(n-1)^{th}$ function of time. The greater the number of data points used for determining this polynomial, the more accurate the overall shape of the load-time curve and, more importantly, the more accurate the resulting calculations. However, in this analysis, no more than 3 data points have been used due to the complexity of higher order methods.

The impulse is the integral of the load as a function of time. Dividing the impulse by mass yields the change in velocity. Since, in impact tests, the force and the initial velocity are in opposite directions, subtracting impulse divided by effective tup mass from the velocity at the beginning of the first interval yields velocity as a function of time. In this case, load is an $n\text{-}1^{th}$ function of time, so velocity is an n^{th} order function of time. The initial velocity is known. Subsequent velocities are calculated as described above.

By integrating velocity, v_i, it is possible to determine displacement. If as in this analysis, velocity is an n^{th} order function of time, then displacement is an $(n + 1)^{th}$ order function of time. Power is determined by multiplying load and velocity. Since load is an $n\text{-}1^{th}$ order function of time and velocity is an n^{th} order function of time, the result is power expressed in this analysis as a $(2n - 1)^{th}$ order function of time. Integrating the power result yields energy, expressed as a $(2n)^{th}$ order function of time.

To determine the velocity, displacement, power, and energy measurements at subsequent sampling intervals during the event, the first load data is removed from the data set of n elements, and the first load data point following the set is added. Thus, what had been P_{i+1} now becomes P_i and what was v_{i+1} is now v_i. The same process as used with the first sampling interval is then applied to the new data set to derive velocity, displacement, power, and energy measurements for that new interval. This process is repeated until all the intervals in the event have been dealt with.

Since the method of obtaining the impulse is, in effect, an $(n-1)^{th}$ order Newton-Cotes integration scheme, it is to be expected that the accuracy of these calculations will asymptotically approach the correct values as n increases. Furthermore, when the results from any odd order method are subtracted from the next lower order method, the result will be a good estimate of the remaining error attributable to the numerical method. [62].

The specific method which assumes load to be a first order function of time and the method which assumes load to be a second order function of time were derived using MACSYMA, a symbolic algebra program. The edited outputs and the associated MACSYMA inputs are shown in the sections below. Following the MACSYMA information is a proof demonstrating the equivalence of the equations derived in the first and second order methods and the Augland-Grumbach equation.

Note that the derivations shown are strictly true only for the pendulum method and that a correction must be added for drop tower calculations. How this is done is explained on page 262.

A.4.1 Load as a First Order Function of Time

A.4.1.1 Derivation

Let P(t) be defined as the load as a function of time. Assume that this is a linear function:

$$P(t) = P0 + P1\ t \tag{A-1}$$

where P_0 and P_1 are arbitrary constants.

Let:

$$\Delta t \equiv \text{the sampling interval} \tag{A-2}$$

P_i is defined as the load at the beginning of the interval. Hence:

$$P_i = P(0) \tag{A-3}$$

P_{i+1} is defined as the load at the end of the interval. Hence:

$$P_{i+1} = P(\Delta t) \tag{A-4}$$

Substituting Equation A-3 into Equation A-1 yields:

$$P_i = P0 \tag{A-5}$$

Substituting Equation A-4 into Equation A-1 yields:

$$P_{i+1} = \Delta t\ P1 + P0 \tag{A-6}$$

Solving Equation A-5 and Equation A-6 simultaneously and substituting in Equation A-1:

$$P(t) = \frac{(P_{i+1} - P_i)\ t}{\Delta t} + P_i \tag{A-7}$$

Let V(t) be defined as the velocity as a function of time.
V_i is defined as the velocity at the beginning of the interval. Hence:

$$V_i = V(0) \tag{A-8}$$

V_{i+1} is defined as the velocity at the end of the interval. Hence:

$$V_{i+1} = V(\Delta t) \tag{A-9}$$

V_{i+2} is defined as the velocity at the end of the following interval. Hence:

$$V_{i+2} = V(2\Delta t) \tag{A-10}$$

Impulse is the integral of load with respect to time. The change in velocity as a function of time is the impulse as a function of time divided by the effective mass of the tup (M). Hence:

$$V(t) - V_i = - \left(\frac{(P_{i+1} - P_i)\, t^2}{2\Delta t\, M} + \frac{P_i\, t}{M} \right) \qquad \text{(A-11)}$$

It follows that:

$$V_{i+1} = V_i - \frac{\Delta t\, (P_{i+1} - P_i)}{2M} - \frac{\Delta t\, P_i}{M} \qquad \text{(A-12)}$$

and:

$$V_{i+2} = V_i - \frac{2\, \Delta t\, P_{i+1}}{M} \qquad \text{(A-13)}$$

The assumption that load varies as a linear function of time leads to the conclusion that velocity varies as a quadratic function of time. Hence:

$$V(t) = V0 + V1\, t + \frac{1}{2}\, V2\, t^2 \qquad \text{(A-14)}$$

where V0, V1 and V2 are arbitrary constants.

Substituting Equation A-8 into Equation A-14 yields:

$$V_i = V0 \qquad \text{(A-15)}$$

Substituting Equation A-9 into Equation A-14 yields:

$$V_{i+1} = \frac{\Delta t^2 \, V2}{2} + \Delta t \, V1 + V0 \qquad \text{(A-16)}$$

Substituting Equation A-10 into Equation A-14 yields:

$$V_{i+2} = 2\Delta t^2 \, V2 + 2\Delta t \, V1 + V0 \qquad \text{(A-17)}$$

Simultaneously solving Equations A-15 - A-17 and substituting the result into Equation A-14 yields the following result:

$$V(t) = V_i + \frac{(V_{i+2} - 2V_{i+1} + V_i)\, t^2}{2\Delta t^2} - \frac{(V_{i+2} - 4V_{i+1} + 3V_i)\, t}{2\Delta t} \qquad \text{(A-18)}$$

Let X(t) be defined as the displacement as a function of time. X_i is the displacement at the beginning of the interval. Ie.:

$$X_i = X(0) \qquad \text{(A-19)}$$

X_{i+1} is the displacement at the end of the interval. Hence:

$$X_{i+1} = X(\Delta t) \qquad \text{(A-20)}$$

The change in displacement is the integral of velocity with respect to time. Hence by integrating Equation A-18 and using Equation A-19 it follows that:

$$X(t) = X_i + \frac{(V_{i+2} - 2V_{i+1} + V_i)\, t^3}{6\Delta t^2} + \frac{(-V_{i+2} + 4V_{i+1} - 3V_i)\, t^2}{4\Delta t} + V_i t \qquad \text{(A-21)}$$

Substituting Equation A-20 into Equation A-21:

$$X_{i+1} = X_i - \left(\frac{\Delta t\ V_{i+2} - 8\ \Delta t\ V_{i+1} - 5\ \Delta t\ V_i}{12}\right) \qquad \text{(A-22)}$$

Power is the product of load and velocity. Hence power is represented by:

$$P(t)\ V(t) = \left(\frac{(P_{i+1} - P_i)\ t}{\Delta t} + P_i\right) \qquad \text{(A-23)}$$

$$\left(\frac{(V_{i+2} - 2V_{i+1} + V_i)\ t^2}{2\Delta t^2}\right.$$

$$\left. - \frac{(V_{i+2} - 4V_{i+1} + 3V_i)\ t}{2\Delta t} + V_i\right)$$

Let E(t) be defined as the energy absorbed up to a given time. Ei is defined as the energy absorbed by the end of the interval. Hence:

$$Ei = E(0) \qquad \text{(A-24)}$$

and

$$Ei+1 = E(\Delta t) \qquad \text{(A-25)}$$

The energy absorbed up to a given time is the integral of power absorbed with respect to time. Hence integrating Equation A-23 and using Equation A-24:

$$
\begin{aligned}
E(t) \quad = \quad E_i \quad & + \quad \frac{(3P_{i+1} \; - \; 3P_i)\; V_{i-2}^{4}\; t}{24\Delta t^3} \qquad\qquad \text{(A-26)} \\
& + \quad \frac{(6P_i \; - \; 6P_{i+1})\; V_{i+1} \;\; t^4}{} \\
& + \quad \frac{3V_i \;\; P_{i+1} \;\; t^4}{24\Delta t^3} \quad - \quad \frac{3P_i \; V_i)\; t^4}{24\Delta t^3} \\
& + \quad \frac{(8\Delta t \; P_i \; - \; 4\Delta t \; P_{i+1})\; V_{i+2} \;\; t^3}{24\Delta t^3} \\
& + \quad \frac{(16\Delta t \; P_{i+1} \;\; - \;\; 24\Delta t \; P_i)\; V_{i+1} \;\; t^3}{24\Delta t^3} \\
& - \quad \frac{12\Delta t \; V_i \; P_{i+1} \;\; t^3}{24\Delta t^3} \quad + \quad \frac{16\Delta t \; P_i \; V_i)\; t^3}{24\Delta t^3} \\
& + \quad \frac{-6\Delta t^2 \; P_i \; V_{i+2} \;\; t^2}{24\Delta t^3} \quad + \quad \frac{24\Delta t^2 \; P_i \; V_{i+1} \;\; t^2}{24\Delta t^3} \\
& + \quad \frac{12\Delta t^2 \; V_i \; P_{i+1} \;\; t^2}{24\Delta t^3} \quad - \quad \frac{30\Delta t^2 \; P_i \; V_i \;\; t^2}{24\Delta t^3} \\
& + \quad \frac{24\Delta t^3 \; P_i \; V_i \;\; t}{24\Delta t^3}
\end{aligned}
$$

Substituting Equation A-25 into Equation A-26 yields the following result:

$$E_{i+1} = E_i - \left(\frac{(\Delta t\, P_{i+1} + \Delta t\, P_i)\, V_{i+2}}{24} + \frac{(-\,10\Delta t\, P_{i+1} - 6\Delta t\, P_i)\, V_{i+1}}{24} - \frac{3\Delta t\, V_i\, P_{i+1}}{24} - \frac{7\Delta t\, P_i\, V_i}{24}\right) \qquad \text{(A-27)}$$

A.4.1.2 Recapitulation of the First Order Method

P_i and P_{i+1} are calculated from the input data as described in the text in section 1.4. The initial value of V_i is obtained from the input data as described in the text in section 1.4.1.2. Subsequent values of V_i are the value of V_{i+1} in the previous interval (Equation A-12). The initial value of X_i is zero. Subsequent values of X_i are the values of X_{i+1} in the previous interval (Equation A-22). The initial value of E_i is zero. Subsequent values of E_i are the values of E_{i+1} in the previous interval (Equation A-27).

$$V_{i+1} = V_i - \frac{\Delta t\ (P_{i+1} - P_i)}{2M} + \frac{\Delta t\ P_i}{M} \tag{A-12}$$

$$V_{i+2} = V_i - \frac{2\ \Delta t\ P_{i+1}}{M} \tag{A-13}$$

$$X_{i+1} = X_i - \left(\frac{\Delta t\ V_{i+2} - 8\ \Delta t\ V_{i+1} - 5\ \Delta t\ V_i}{12}\right) \tag{A-22}$$

$$E_{i+1} = E_i - \left(\frac{(\Delta t\ P_{i+1} + \Delta t\ P_i)\ V_{i+2}}{24} + \frac{(-\ 10\Delta t\ P_{i+1} - 6\Delta t\ P_i)\ V_{i+1}}{24} - \frac{3\Delta t\ V_i\ P_{i+1}}{24} - \frac{7\Delta t\ P_i\ V_i}{24}\right) \tag{A-27}$$

A.4.1.3 MACSYMA program Used to Derive the Equations Used for the First Order Method

```
P(t) := P0 + P1 * t;
P[i] = P(0);
P[i+1] = P(Δt);
d3-d2;
solve(d4,P1);
solve(d2,P0);
P[t] = P0 + P1 * t;
d7,d5;
d8,d6;
integrate(rhs(d9),t);
(d10/M);
V[t] - V[i] = d11;
t = Δt;
d11,d13;
V[i+1] = V[i] - d14;
ratsimp(d15);
t = 2 * Δt;
d11,d17;
V[i+2] = V[i] - d18;
ratsimp(d19);
V(t) := V0 + V1 * t + (1/2) * V2 * t**2;
V[i] = V(0);
V[i+1] = V(Δt);
V[i+2] = V(2 * Δt);
solve(d22,V0);
d23-d22;
d24-d22;
d27/2;
d26-d28;
solve(d29,V2);
d27,d30;
solve(d31,V1);
V[t] = V(t);
d33,d25;
d34,d32;
d35,d30;
ratsimp(d36);
integrate(rhs(d37),t);
X(t) := X[i] + d38;
X[t] = X(t);
```

```
ratsimp(d40);
t = Δt;
d38,d42;
X[i+1] = X[i] + d43;
ratsimp(d44);
d9 * d36;
integrate(rhs(d46),t);
E(t) := E[i] + d47;
E[t] = E(t);
t = Δt;
d47,d50;
E[i+1] = E[i] + d51;
ratsimp(d52);
" ";
"------------------------------------------------------------";
d16;
d19;
d45;
d53;
"------------------------------------------------------------";
" ";

quit();
```

A.4.2 Load as a Second Order Function of Time

A.4.2.1 Derivation

Let P(t) be defined as the load as a function of time. Assume that this is a quadratic function:

$$P(t) = P0 + P1\ t + \frac{1}{2}\ P2\ t^2 \qquad \text{(A-28)}$$

where P_0 and P_1 are arbitrary constants.

Let:

$$\Delta t \equiv \text{the sampling interval} \qquad \text{(A-29)}$$

P_i is defined as the load at the beginning of the interval. Hence:

$$P_i = P(0) \qquad \text{(A-30)}$$

P_{i+1} is defined as the load at the end of the interval. Hence:

$$P_{i+1} = P(\Delta t) \qquad \text{(A-31)}$$

P_{i+2} is defined as the load at the end of the next interval. Hence:

$$P_{i+2} = P(2\Delta t) \qquad \text{(A-32)}$$

Substituting Equation A-30 into Equation A-28 yields:

$$P_i = P0 \qquad \text{(A-33)}$$

Substituting Equation A-31 into Equation A-28 yields:

$$P_{i+1} = \frac{\Delta t^2\ P2}{2} + \Delta t\ P1 + P0 \qquad \text{(A-34)}$$

Substituting Equation A-32 into Equation A-28 yields:

$$P_{i+2} \quad = \quad 2\Delta t^2 \; P2 \quad + \quad 2\Delta t \; P1 \quad + \quad P0 \qquad \text{(A-35)}$$

Solving Equations A-33, A-34, and A-35 simultaneously and substituting into Equation A-28:

$$P_t \quad = \quad \frac{(P_{i+2} - 2P_{i+1} + P_i)\; t^2}{2\Delta t^2} \qquad \text{(A-36)}$$

$$- \quad \frac{(P_{i+2} - 4P_{i+1} + 3P_i)\; t}{2\Delta t} \quad + \quad P_i$$

Let V(t) be defined as the velocity as a function of time. V_i is defined as the velocity at the beginning of the interval. Hence:

$$V_i \quad = \quad V(0) \qquad \text{(A-37)}$$

V_{i+1} is defined as the velocity at the end of the interval. Hence:

$$V_{i+1} \quad = \quad V(\Delta t) \qquad \text{(A-38)}$$

V_{i+2} is defined as the velocity at the end of the following interval. Hence:

$$V_{i+2} \quad = \quad V(2\Delta t) \qquad \text{(A-39)}$$

V_{i+3} is defined as the velocity at the end of the second interval after the one over which calculations are being made. Hence:

$$V_{i+3} \quad = \quad V(3\Delta t) \qquad \text{(A-40)}$$

Impulse is the integral of load with respect to time. The change in velocity as a function of time is the impulse as a function of time divided by the effective mass of the tup (M). Hence:

$$V(t) - V_i = \left(\frac{(P_{i+2} - 2P_{i+1} + P_i)\, t^3}{6\Delta t^2\, M} + \frac{(-\Delta t\, P_{i+2} + 4\Delta t\, P_{i+1} - 3\Delta t\, P_i)\, t^2}{4\Delta t\, M} + \frac{P_i\, t}{M} \right) \tag{A-41}$$

It follows that:

$$V_{i+1} = V_i + \frac{\Delta t\, P_{i+2} - 8\Delta t\, P_{i+1} - 5\Delta t\, P_i}{12M} \tag{A-42}$$

and:

$$V_{i+2} = V_i + \frac{-\Delta t\, P_{i+2} - 4\Delta t\, P_{i+1} - \Delta t\, P_i}{3M} \tag{A-43}$$

and:

$$V_{i+3} = V_i + \frac{-9\Delta t\, P_{i+2} - 3\Delta t\, P_i}{4M} \tag{A-44}$$

The assumption that load varies as a quadratic function of time leads to the conclusion that velocity varies as a cubic function of time. Hence:

$$V(t) = V0 + V1\, t + \frac{1}{2}\, V2\, t^2 + \frac{1}{3}\, V3\, t^3 \tag{A-45}$$

where V0, V1, V2, and V3 are arbitrary constants.

Substituting Equation A-37 into Equation A-45 yields:

$$V_i = V0 \tag{A-46}$$

Substituting Equation A-38 into Equation A-45 yields:

$$V_{i+1} = \frac{\Delta t^3\ V3}{3} + \frac{\Delta t^2\ V2}{2} + \Delta t\ V1 + V0 \tag{A-47}$$

Substituting Equation A-39 into Equation A-45 yields:

$$V_{i+2} = \frac{8\Delta t^3\ V3}{3} + 2\Delta t^2\ V2 + 2\Delta t\ V1 + V0 \tag{A-48}$$

Substituting Equation A-40 into Equation A-45 yields:

$$V_{i+3} = 9\Delta t^3\ V3 + \frac{9\Delta t^2\ V2}{2} + 3\Delta t\ V1 + V0 \tag{A-49}$$

Simultaneously solving Equations A-46, A-47, A-48, and A-49 and substituting the result into Equation A-45 yields the following result:

$$V(t) = V_i + \frac{(V_{i+3} - 3V_{i+2} + 3V_{i+1} - V_i)\ t^3}{6\Delta t^3} \tag{A-50}$$

$$- \frac{(V_{i+3} - 4V_{i+2} + 5V_{i+1} - 2V_i)\ t^2}{2\Delta t^2}$$

$$+ \frac{(2V_{i+3} - 9V_{i+2} + 18V_{i+1} - 11V_i)\ t}{6\Delta t}$$

Let $X(t)$ be defined as the displacement as a function of time. X_i is the displacement at the beginning of the interval. Ie.:

$$X_i = X(0) \tag{A-51}$$

X_{i+1} is the displacement at the end of the interval. Hence:

$$X_{i+1} = X(\Delta t) \tag{A-52}$$

The change in displacement is the integral of velocity with respect to time. Hence by integrating Equation A-50 and using Equation A-51 it follows that:

$$\begin{aligned} X(t) = X_i &+ \frac{(V_{i+3} - 3V_{i+2} + 3V_{i+1} - V_i)\, t^4}{24\Delta t^3} \\ &- \frac{(V_{i+3} - 4V_{i+2} + 5V_{i+1} - 2V_i)\, t^3}{6\Delta t^2} \\ &+ \frac{(2V_{i+3} - 9V_{i+2} + 18V_{i+1} - 11V_i)\, t^2}{12\Delta t} \\ &+ V_i\, t \end{aligned} \tag{A-53}$$

Substituting Equation A-52 into Equation A-53:

$$X_{i+1} = X_i + \frac{\Delta t\ V_{i+3} - 5\Delta t\ V_{i+2} + 19\Delta t\ V_{i+1} + 9\Delta t\ V_i}{24\Delta t^3} \tag{A-54}$$

Power is the product of load and velocity. Hence power is represented by:

$$P(t)\ V(t) = \left(\frac{(P_{i+2} - 2P_{i+1} + P_i)\ t^2}{2\Delta t^2} - \frac{(P_{i+2} - 4P_{i+1} + 3P_i)\ t}{2\Delta t} + P_i\right) \tag{A-55}$$

$$\left(\frac{(V_{i+3} - 3V_{i+2} + 3V_{i+1} - V_i)\ t^3}{6\Delta t^3} - \frac{(V_{i+3} - 4V_{i+2} + 5V_{i+1} - 2V_i)\ t^2}{2\Delta t^2} + \frac{(2V_{i+3} - 9V_{i+2} + 18V_{i+1} - 11V_i)\ t}{6\Delta t} + V_i\right)$$

Let E(t) be defined as the energy absorbed up to a given time. E_i is defined as the energy absorbed by the end of the interval. Hence:

$$E_i = E(0) \tag{A-56}$$

and

$$E_{i+1} = E(\Delta t) \tag{A-57}$$

The energy absorbed up to a given time is the integral of power absorbed with respect to time. Hence integrating Equation A-55 and using Equation A-56:

$$E(t) = E_i + \frac{(10P_{i+2} - 20P_{i+1} + 10P_i)\ V_{i+3}\ t^6}{720\Delta t^5} \tag{A-58}$$

$$+ \frac{(-30P_{i+2} + 60P_{i+1} - 30P_i)\ V_{i+2} t^6}{720\Delta t^5} + \frac{(30V_{i+1} - 10V_i)\ P_{i+2}\ t^6}{720\Delta t^5}$$

$$+ \frac{(30P_i - 60P_{i+1})\ V_{i+1} t^6}{720\Delta t^5} + \frac{20V_i\ P_{i+1} t^6}{720\Delta t^5} - \frac{10P_i\ V_i)\ t^6}{720\Delta t^5}$$

$$+ \frac{(-48\Delta t\ P_{i+2} + 120\Delta t\ P_{i+1} - 72\Delta t\ P_i)\ \ V_{i+3}\ t^5}{720\Delta t^5}$$

$$+ \frac{(180\Delta t\ P_{i+2} - 432\Delta t\ P_{i+1} + 252\Delta t\ P_i)\ \ V_{i+2}\ t^5}{720\Delta t^5}$$

$$+ \frac{(84\Delta t\ V_i - 216\Delta t\ V_{i+1})\ P_{i+2}\ t^5}{720\Delta t^5}$$

$$+ \frac{(504\Delta t\ P_{i+1} - 288\Delta t\ P_i)\ V_{i+1}\ t^5}{720\Delta t^5}$$

$$- \frac{192\Delta t\ V_i\ P_{i+1}\ t^5}{720\Delta t^5} + \frac{108\Delta t\ P_i\ V_i)\ t^5}{720\Delta t^5}$$

$$+ \frac{(75\,\Delta t^2 P_{i+2} - 240\Delta t^2 P_{i+1} + 195\Delta t^2 P_i)\ V_{i+3}\ t^4}{720\Delta t^5}$$

$$+ \frac{(-315\Delta t^2 P_{i+2} + 990\Delta t^2 P_{i+1} - 765\Delta t^2 P_i)\ V_{i+2}\ t^4}{720\Delta t^5}$$

$$+ \frac{(495\,\Delta t^2 V_{i+1} - 255\Delta t^2 V_i)\ P_{i+2}\ t^4}{720\Delta t^5}$$

$$+ \frac{(1035\Delta t^2 P_i - 1440\Delta t^2 P_{i+1})\ V_{i+1}\ t^4}{720\Delta t^5}$$

$$+ \frac{690\,\Delta t^2 V_i\ P_{i+1} t^4}{720\Delta t^5} - \frac{465\,\Delta t^2 P_i\ V_i\ t^4}{720\Delta t^5}$$

$$+ \frac{(-40\Delta t^3 P_{i+2} + 160\,\Delta t^3 P_{i+1} - 240\Delta t^3 P_i)\ V_{i+3}\ t^3}{720\Delta t^5}$$

$$+ \frac{(180\Delta t^3 P_{i+2} - 720\Delta t^3 P_{i+1} + 1020\Delta t^3 P_i)\ V_{i+2}\ t^3}{720\Delta t^5}$$

$$+ \frac{(340\,\Delta t^3 V_i - 360\Delta t^3 V_{i+1})\ P_{i+2}\ t^3}{720\Delta t^5}$$

$$+ \frac{(1440\Delta t^3 P_{i+1} - 1680\Delta t^3 P_i)\ V_{i+1}\ t^3}{720\Delta t^5}$$

$$- \frac{1120\Delta t^3 V_i P_{i+1} t^3}{720\Delta t^5} + \frac{1020\Delta t^3 P_i V_i) t^3}{720\Delta t^5}$$

$$+ \frac{(120\Delta t^4 P_i V_{i+3} t^2}{720\Delta t^5} - \frac{540\Delta t^4 P_i V_{i+2} t^2}{720\Delta t^5}$$

$$- \frac{180\Delta t^4 V_i P_{i+2} t^2}{720\Delta t^5} + \frac{1080\Delta t^4 P_i V_{i+1} t^2}{720\Delta t^5}$$

$$+ \frac{720\Delta t^4 V_i P_{i+1} t^2}{720\Delta t^5} - \frac{1200\Delta t^4 P_i V_i t^2}{720\Delta t^5}$$

$$+ \frac{720 \Delta t^5 P_i V_i t}{720\Delta t^5}$$

Substituting Equation A-57 in Equation A-58:

$$E_{i+1} = E_i - \left(\frac{(3\Delta t\ P_{i+2} - 20\Delta t\ P_{i+1} - 13\Delta t\ P_i)\ V_{i+3}}{720}\right. \tag{A-59}$$

$$+ \frac{(-15\Delta t\ P_{i+2} + 102\Delta t\ P_{i+1} + 63\ \Delta t\ P_i)\ V_{i+2}}{720}$$

$$+ \frac{(51\Delta t\ V_{i+1} + 21\Delta t\ V_i)\ P_{i+2}}{720}$$

$$+ \frac{(-444\Delta t\ P_{i+1} - 177\Delta t\ P_i)\ V_{i+1}}{720}$$

$$\left. - \frac{118\Delta t\ V_i\ P_{i+1}}{720} - \frac{173\Delta t\ P_i\ V_i}{720}\right)$$

A.4.2.2 Recapitulation of the Second Order Method

P_i, P_{i+1}, and P_{i+2} are calculated from the input data as described in the text in section 1.4. The initial value of V_i is obtained from the input data as described in the text in section 1.4.1.2. Subsequent values of V_i are the value of V_{i+1} in the previous interval (Equation A-52). The initial value of X_i is zero. Subsequent values of X_i are the values of X_{i+1} in the previous interval (Equation A-53). The initial value of E_i is zero. Subsequent values of E_i are the values of E_{i+1} in the previous interval (Equation A-54).

$$V_{i+1} = V_i + \frac{\Delta t\ P_{i+2} - 8\Delta t\ P_{i+1} - 5\Delta t\ P_i}{12M} \tag{A-42}$$

and:

$$V_{i+2} = V_i + \frac{-\Delta t\ P_{i+2} - 4\Delta t\ P_{i+1} - \Delta t\ P_i}{3M} \tag{A-43}$$

and:

$$V_{i+3} = V_i + \frac{-9\Delta t\ P_{i+2} - 3\Delta t\ P_i}{4M} \tag{A-44}$$

$$X_{i+1} = X_i + \frac{\Delta t\ V_{i+3} - 5\Delta t\ V_{i+2} + 19\Delta t\ V_{i+1} + 9\Delta t\ V_i}{24\Delta t^3} \tag{A-54}$$

$$E_{i+1} = E_i - \Big(\frac{(3\Delta t\ P_{i+2} - 20\Delta t\ P_{i+1} - 13\Delta t\ P_i)\ V_{i+3}}{720} \tag{A-59}$$

$$+ \frac{(-15\Delta t\ P_{i+2} + 102\Delta t\ P_{i+1} + 63\ \Delta t\ P_i)\ V_{i+2}}{720}$$

$$+ \frac{(51\Delta t\ V_{i+1} + 21\Delta t\ V_i)\ P_{i+2}}{720}$$

$$+ \frac{(-444\Delta t\ P_{i+1} - 177\Delta t\ P_i)\ V_{i+1}}{720}$$

$$- \frac{118\Delta t\ V_i\ P_{i+1}}{720} - \frac{173\Delta t\ P_i\ V_i}{720} \Big)$$

A.4.2.3 MACSYMA program Used to Derive the Equations Used for the Second Order Method

```
P(t) := P0 + P1 * t + (1/2) * P2 * t**2;
P[i] = P(0);
P[i+1] = P(Δt);
P[i+2] = P(2 * Δt);
d3-d2;
d4-d2;
d6/2;
d7-d5;
solve(d2,P0);
solve(d8,P2);
d5,d10;
solve(d11,P1);
P[t] = P(t);
d13,d9;
d14,d12;
d15,d10;
ratsimp(d16);
integrate(rhs(d17),t);
(d18/M);
V[t] - V[i] = d19;
t = Δt;
d19,d21;
V[i+1] = V[i] - d22;
ratsimp(d23);
t = 2 * Δt;
d19,d25;
V[i+2] = V[i] - d26;
ratsimp(d27);
t = 3 * Δt;
d19,d29;
V[i+3] = V[i] - d30;
ratsimp(d31);
V(t) := V0 + V1 * t + (1/2) * V2 * t**2 + (1/3) * V3 * t**3;
V[i] = V(0);
V[i+1] = V(Δt);
V[i+2] = V(2*Δt);
V[i+3] = V(3*Δt);
d35-d34;
d36-d34;
d37-d34;
```

```
(d39/2);
(d40/3);
d38-d41;
d38-d42;
ratsimp(d43);
ratsimp(d44);
(2*d45);
d46-d47;
solve(d48,V3);
d46,d49;
solve(d50,V2);
d40,d49;
d52,d51;
solve(d53,V1);
solve(d34,V0);
V[t] = V(t);
d56,d55;
d57,d54;
d58,d51;
d59,d49;
integrate(rhs(d60),t);
X(t) := X[i] + d61;
X[t] = X(t);
ratsimp(d63);
t = Δt;
d61,d65;
X[i+1] = X[i] + d66;
ratsimp(d67);
d16 * d60;
integrate(rhs(d69),t);
E(t) := E[i] + d70;
E[t] = E(t);
t = Δt;
d70,d73;
E[i+1] = E[i] + d74;
ratsimp(d75);
" ";
"------------------------------------------------------------";
d24;
d28;
d32;
d68;
d76;
"------------------------------------------------------------";
" ";
quit();
```

A.5 EQUIVALENCE PROOF: MACSYMA DERIVED EQUATIONS AND THE AUGLAND-GRUMBACH EQUATION

It can be shown that the equations in the first and second order versions of the double integral method are specific cases of the Augland-Grumbach Equation. That equation is:

$$\Delta E_{tot} = E_a - \frac{E_a^2}{4E_o} \tag{A-60}$$

$$\text{where: } E_a = V_i \int P(t)dt \tag{A-61}$$

$$\text{and } E_o = 1/2\, M_{eff}\, V_i^2 \tag{A-62}$$

so:

$$\Delta E_{tot} = V_i \int P(t)\,dt - \frac{V_i^2 \left(\int P(t)\,dt\right)^2}{4 \frac{1}{2} M_{eff} V_i^2} \tag{A-63}$$

$$\Delta E_{tot} = V_i \int P(t)\,dt - \frac{\left(\int P(t)\,dt\right)^2}{2M} \tag{A-64}$$

The derivations of the first and second order methods described earlier in this appendix have the following form:

$$V(t) = V_i - \int \frac{P(t)\,dt}{M} \tag{A-65}$$

$$\text{Power} = \frac{dE}{dt} = P(t)\,V(t) \tag{A-66}$$

Hence:

$$\frac{dE}{dt} = P(t)\left(\frac{V_i - \int P(t)\,dt}{M}\right) \qquad (A\text{-}67)$$

and:

$$\frac{dE}{dt} = V_i\ P(t) - \frac{P(t)\int P(t)\,dt}{M} \qquad (A\text{-}68)$$

Integrating:

$$\Delta E = \int V_i\ P(t)\,dt - \int P(t)\left[\frac{\int P(t)\,dt}{M}\right] dt \qquad (A\text{-}69)$$

but:

$$P(t)\,dt = d\left(\int P(t)\,dt\right) \qquad (A\text{-}70)$$

Therefore:

$$\Delta E = V_i \int P(t)\,dt - \frac{(P(t)\ dt)^2}{2M} \qquad (A\text{-}71)$$

The Augland-Grumbach equation in the form shown above (Equation A-64) and the double integral energy equation (A-71) are identical. Because of the simplicity of the form of the Augland-Grumbach equation it, not the double integral energy equation, was used in ASTIR (see ASTIRCALC text). In the event that a drop tower test has been conducted, the following correction must be made to be strictly correct.

$$\Delta E = V_i \int P(t)\,dt - \frac{(P(t)\ dt)^2}{2M} + g\int P(t)\,t\,dt \qquad (A\text{-}72)$$

where $g \equiv$ the local acceleration of gravity.

A.6 SOURCE CODE

A.6.1 Overview of the Astir Source Code

Conceptually, ASTIR is composed of six subroutines, five of which use so much memory that only one of the five can be loaded into memory at a time. In UCSD Pascal, such memory overlays are possible and are called segment procedures. Actually, tow of the five segment procedures, ASTIRGRAF and ASTIRREPT, were so large that each had to be split into two segment procedures, the first performing initialization and display setup and the second carrying out the actual work.

ASTIRMETH is the smallest of the subroutines in ASTIR. It obtains the test type, velocity input method, and user type.

ASTIRREAD obtains the Apple DOS 3.3 catalog from the data disk, displays it to the user, obtains the user's data file choice and reads the data file into memory. This is done by emulating the file retrieval subroutines in the Apple DOS 3.3 operating system.

ASTIRGRAF displays the load time data to the user graphically and obtains from him cursor positions marking salient features of the record. Machine language graphics were necessary to plot the data sufficiently rapidly.

ASTIRDISP obtains the calibration and test data from the user and records the data in two files on the program disk. Input consists of strings which are then converted to real numbers, thus preventing the program from crashing due to inappropriate input.

ASTIRCALC is the heart of the ASTIR program. It performs numerical calculations on the load time data using the calibration and test condition data supplied by the user obtaining load, displacement, and energy data.

ASTIRREPT displays the input data graphically and the calculated results in tabular form and, if requested, makes hard copy.

The ASTIR program depends on several libraries which contain subroutines called by its six main subroutines. The libraries are ASTIRSEGS, TYPESTUFF, and PLOTSTUFF.

ASTIRSEGS contains STARTGRAF, the start-up routine for ASTIRGRAF, STARTREPT, the start-up routine for ASTIRREPT and high level graphics subroutines for both ASTIRGRAF and ASTIRREPT.

TYPESTUFF contains constant and type declarations for ASTIR and several low level input-output subroutines.

PLOTSTUFF contains mid-level subroutines upon which ASTIRGRAF, ASTIRREPT, and ASTIRSEGS depend.

Both TYPESTUFF and PLOTSTUFF contain subroutines which are defined as external (i.e., they are 6502 machine language subroutines). These machine language subroutines are found in ASTIRSTUFF.

A.6.2 The ASTIR Program

```
(*$S+,N+*)
program astir(input,output);

uses turtlegraphics,transcend,typestuff,plotstuff;

var
      curve            :       apfiletype;
      cursors          :       cursorstype;
      abdec,
      vel1,vel2        :       real;
      channel          :       char;
      nrg              :       nrgtype;
      heading          :       ident;
      totals           :       duo;
      next             :       boolean;
      disposables      :       factortype;
      test             :       testtype;
      run              :       runtype;
      filternum,
      scale            :       byte;
      cmd              :       cmds;
      labels           :       labelstype;
      n                :       integer;
```

```
(*$I astirread.text *)
(*$I astirsegs.text *)
(*$I astirgraf.text *)
(*$I astirdisp.text *)
(*$I astircalc.text *)
(*$I astirrept.text *)
{The above inclusions are segment procedures.}
{The following inclusion is an ordinary procedure.}
(*$I astirmeth.text *)

begin{astir}
(*$R typestuff*)
   while true do
   begin
       astirmeth(run);
       astirread(curve,heading);
       startgraf(cmd,labels,filternum,scale,curve,cursors,channel);
       astirgraf(curve,cursors,channel,filternum,scale);
       next := false;
       while not(next) do
       begin
           astirdisp(disposables,test);
           astircalc(curve,cursors,disposables,abdec,vel1,vel2,
                     channel,nrg,totals);
           startrept(cmd,scale);
           astirrept(curve,cursors,nrg,heading,totals,next,
           vel1,vel2,filternum,scale);
       end;
   end;
end.{astir}
```

A.6.3 ASTIRMETH

```
procedure astirmeth(var run:runtype);
var
     ch       :        char;
     runbook  :        file of runtype:

     procedure loadr(var run:runtype);

     begin
          with run do
          begin
               kind := DROP;
               velocity := KEYB;
               user := SUPER;
          end;
     end;

     (*$I-*)
     procedure disci(var run:runtype);

     begin
          reset(runbook,'INSTRUCT');
          if ioresult = 0 then
          begin
               get(runbook);
               close(runbook);
               run := runbook^;
          end else begin
               close(runbook);
               loadr(run);
          end;
     end;
     (*$I+*)

     procedure disco(run:runtype);

     begin
          rewrite(runbook,'INSTRUCT');
          runbook^ := run;
          put(runbook);
          close(runbook,lock);
     end;
```

```
procedure printkind(run:runtype);

begin
    case run.kind of
        PEND    :       write('PENDULUM');
        DROP    :       write('DROP TOWER');
        SLOW    :       write('SLOW BEND');
    end;
end;

procedure printvel(run:runtype);

begin
    case run.velocity of
        KEYB    :       write('KEYBOARD');
        CURV    :       write('CURVE');
    end;
end;

procedure printuser(run:runtype);

begin
    case run.user of
        NORMAL  :       write('NORMAL');
        SUPER   :       write('SUPER');
        PRO     :       write('PRO');
    end;
end;
```

```
procedure screen;

begin
    cleartext;
    textmode;
    gotoxy(37,5);
    write('ASTIR');
    gotoxy(33,7);
    write('METHOD CHOICES');
    gotoxy(33,8);
    write('--------------');
    gotoxy(29,11);
    write('TEST TYPE:  ');
    printkind(run);
    gotoxy(29,13);
    write('VELOCITY:   ');
    printvel(run);
    gotoxy(29,15);
    write('USER TYPE:  ');
    printuser(run);
    gotoxy(25,11);
    write('-->');
    gotoxy(20,18);
    write('  A(CCEPT),C(HANGE),N(EXT),L(AST),Q(UIT)');
    gotoxy(20,18);
end;

procedure controls(var run:runtype);

var
    l,o      :       integer;
    ch       :       char;

    procedure change(var run:runtype);

        procedure prepscreen;

        begin
            gotoxy(41,(2 * l + 9));
            write('          ');
            gotoxy(41,(2 * l + 9));
        end;
```

```
procedure getkind(var run:runtype);

begin
    with run do
    begin
        case kind of
            PEND :  kind := DROP;
            DROP :  kind := SLOW;
            SLOW :  kind := PEND;
        end;
    end;
end;

procedure getvel(var run:runtype);

begin
    with run do
    begin
        case velocity of
            KEYB :  velocity := CURV;
            CURV :  velocity := KEYB;
        end;
    end;
end;

procedure getuser(var run:runtype);

begin
    with run do
    begin
        case user of
            NORMAL : user := PRO;
            PRO    : user := SUPER;
            SUPER  : user := NORMAL;
        end;
    end;
end;
```

```
begin{change}
    case l of
        1       :       getkind(run);
        2       :       getvel(run);
        3       :       getuser(run);
    end;
    prepscreen;
    case l of
        1       :       printkind(run);
        2       :       printvel(run);
        3       :       printuser(run);
    end;
end{change};

procedure movearrow(o,l:integer);

begin
    gotoxy(25,(2 * o + 9));
    write('   ');
    gotoxy(25,(2 * l + 9));
    write('-->');
end;
```

```
    begin{controls}
        l := 1;
        while ch <> 'A' do
        begin
            ch := chr(0);
            while not(ch in ['A','C','L','N','Q']) do
            begin
                o := 1;
                gotoxy(20,18);
                getc(ch);
                if (ord(ch) >= 32) and
                (ord(ch) <= 126) then write(ch);
            end;
            case ch of
                'A'     :       begin
                        end;
                'C'     :       change(run);
                'L'     :       begin
                            l := l - 1;
                            if l < 1 then
                              l :=  3;
                        end;
                'N'     :       begin
                            l := l + 1;
                            if l > 3 then
                              l := 1;
                        end;
                'Q'     :       begin
                            cleartext;
                            exit(astir);
                        end;
            end;
            movearrow(o,l);
        end;
        cleartext;
    end;{controls}

begin{astirmeth}
    disci(run);
    screen;
    controls(run);
    disco(run);
end;{astirmeth}
```

A.6.4 ASTIRREAD

```
segment procedure astirread(var curve:apfiletype;var heading:ident);

var
     catalog :       cattype;

     segment procedure getcatalog(var catalog : cattype);
     var
          offset,
          trk,sct         :         byte;
          n,m             :         integer;
          sect            :         sector;
          choice          :         byte;

          procedure getsector(var out : sector;disc,trk,sect:byte);
          var
               blck       :         integer;
               offset     :         byte;
               buf        :         block;
          begin
               sect := 15 - sect;
               if sect = 0 then sect := 15
               else if sect = 15 then sect := 0;
               blck := (trk * 8) + trunc(sect / 2);
               unitread(disc,buf,512,blck);
               offset := (trk * 16 + sect) - (blck * 2);
               out := buf[offset];
          end;
```

```
begin
    catalog.length := 0;
    getsector(sect,datadisc,17,0);
    trk := sect[1];
    sct := sect[2];
    while (trk + sct) > 0 do
    begin
        getsector ( sect,datadisc,trk,sct);
        trk := sect[1];
        sct := sect[2];
        for n := 0 to 6 do
        begin
            offset := (11 + n * 35);
            if (sect[offset]<>0) and
            (sect[offset] <> 255) then
            begin
                with catalog.entries[catalog.length] do
                begin
                    trk := sect[offset + 0];
                    sct:= sect[offset + 1];
                    filetype := sect[offset + 2];
                    for m := 0 to 29 do
                        name[m] :=
                         chr(sect[offset+3+m]);
                    length:= sect[offset + 33];
                end;
            catalog.length := catalog.length + 1;
            end;
        end;
    end;
end;

segment procedure getcurve(catalog:cattype;var curve:apfiletype;
                var heading:ident);

var
    choice  :       byte;
    fail    :       boolean;
```

```
procedure getsector(var out : sector;disc,trk,sect:byte);

var

    blck    :       integer;
    offset  :       byte;
    buf     :       block;

begin
    sect := 15 - sect;
    if sect = 0 then sect := 15
    else if sect = 15 then sect := 0;
    blck := (trk * 8) + trunc(sect / 2);
    unitread(disc,buf,512,blck);
    offset := (trk * 16 + sect) - (blck * 2);
    out := buf[offset];
end;

procedure choosecurve(catalog:cattype;var choice:byte);
{Because of the depth of indentation in choose curve, half}
{tabs are used for this procedure                        }

var
    counter :       integer;
    exponent,
    chosen  :       boolean;
    ch      :       char;
    diag,
    line    :       string;
    xx      :       real;

    procedure badchoice;

    begin
    gotoxy(0,22);
    write('INVALID CHOICE PLEASE PRESS RETURN        ');
    readln;
    end;

begin
    cleartext;
    counter := 0;
    chosen := false;
    while not(chosen) do
```

```
begin
with catalog.entries[counter] do
begin
    gotoxy(0,23);
    write('[',counter,']');
    gotoxy(13,23);
    write(filetype);
    gotoxy(17,23);
    write(length);
    gotoxy(21,23);
    write(name);
end;
writeln;
counter := counter + 1;
if (counter mod 20 = 0) or (counter = catalog.length)
then begin
    writeln;
    writeln;
    ch := chr(0);
    while (ch <> 'S') and (ch <> 'C') do
    begin
    gotoxy(0,22);
    write('   S(ELECT CURVE), C(ONTINUE)');
    gotoxy(0,22);
    getc(ch);
    end;
    if ch = 'C'
    then begin
    writeln;
    writeln;
    if counter = catalog.length
    then counter := 0;
    end
    else begin
    diag := 'NG';
    while (diag = 'NG') or
     ((choice < 0) or (choice > catalog.length)) do
```

```
        begin
            diag := 'NG';
            gotoxy(0,22);
            write('[    ] TYPE NUMBER IN');
            write(' [BRACKETS] OF');
            writeln(' CURVE.');
            gotoxy(1,22);
            getline(line,'  ',1,22,2);
            exponent := false;
            sttofp(xx,diag,line,exponent);
            if (diag='NG') or
             ((xx<0) or (xx>catalog.length)) then
            badchoice
            else begin
            choice := trunc(xx);
            chosen := true;
            end;
        end;
        end;
        if chosen
        then begin
        writeln;
        ch := chr(0);
        while not(ch IN ['N','Y']) do
        begin
            gotoxy(0,21);
            writeln('YOU HAVE CHOSEN:  ',
            catalog.entries[choice].name);
            write('IS THAT SATISFACTORY? (Y/N): ');
            write('             ');
            gotoxy(29,22);
            getc(ch);
            writeln;
        end;
        if ch = 'N'
        then begin
            writeln;
            writeln;
            counter := 0;
            chosen := false;
        end else
            cleartext;
        end;
    end;
    end;
end;
```

```
procedure fetchcurve(catalog:cattype;choice:byte;
             var fail:boolean;var curve:apfiletype;
             var heading:ident);

var
    trk,sct,offset,
    length,filetype,
    n,counter       :       byte;
    sect,tslist     :       sector;

begin
    fail := false;
    filetype := catalog.entries[choice].filetype;
    length := catalog.entries[choice].length - 1;
    if ((filetype <> 4) and (filetype <> 132))
    or (length - 2 > maxpage)
    then begin
        if (length - 2 > maxpage)
        then begin
            writeln('FILE IS TOO LONG.');
            writeln('PLEASE PRESS RETURN.');
            readln;
        end else begin
            writeln('FILE IS NOT BYTE TYPE.');
            writeln('PLEASE PRESS RETURN.');
            readln;
        end;
        fail := true;
    end
    else begin
        with catalog.entries[choice] do
        begin
            heading := name;
            getsector(tslist,datadisc,trk,sct);
        end;
        counter := 0;
        while counter < length do
        begin
            offset := 12 + 2 * counter;
            trk := tslist[offset];
            sct := tslist[offset + 1];
            getsector(sect,datadisc,trk,sct);
            with curve do
```

```
                begin
                    if counter < length - 1
                    then For n := 4 to 255 do
                    page[counter,n - 4] := sect[n];
                    if counter >= 1
                    then for n := 0 to 3 do
                    page[(counter-1),(n+252)] :=
                        sect[n];
                end;
                counter := counter + 1;
            end;
            curve.zeropage := counter - 2;
        end;
    end;

  begin{getcurve}
    fail := true;
    while fail do
    begin
        choosecurve(catalog,choice);
        fetchcurve(catalog,choice,fail,curve,heading);
    end;
  end;{getcurve}


begin{astirread}
  getcatalog(catalog);
  getcurve(catalog,curve,heading);
  if run.user = SUPER then
  begin
      write('FINISHED ASTIRREAD');
      readln;
  end;
end{astirread};
```

A.6.5 ASTIRGRAF

```
{-----------------------------------------------------------------}
{   NOTE:  astirgraf is in danger of overwriting the Hi Res graphics  }
{          screen because of its size.  It should be rewritten for    }
{          example by breaking it into 2-3 segments.                  }
{-----------------------------------------------------------------}

segment procedure astirgraf(var curve:apfiletype;var cursors:cursorstype;
                    var channel:char;var filternum,scale:byte);

      function xp(x,start:loc;nopoints:fract):integer;

      var
           temp    :       integer;

      begin
           temp := diff(x,start);
           temp := temp div nopoints.num;
           xp := trunc(temp * nopoints.den + lft);
      end;

      procedure setpoints(start,finish:loc;var nopoints:fract);

      var
           temp    :       integer;
      begin
           temp := diff(finish,start) + 1;
           nopoints.den := 1;
           nopoints.num := 1;
           if temp >= 256 then nopoints.num := temp div 256
           else nopoints.den := 256 div temp;
      end;
```

```
procedure titles(start,finish:loc;l:integer;filternum:byte);

var
    mumble   :       string;

begin
    pencolor(none);
    moveto(52,50);
    wstring('CURSOR:  ');
    moveto(115,50);
    wstring(cmd[0]);
    moveto(115,50);
    wstring(labels[l]);
    str((diff(finish,start) + 1),mumble);
    mumble := concat('POINTS DISPLAYED: ',mumble,'    ');
    moveto(10,40);
    wstring(mumble);
    str((2 * filternum + 1),mumble);
    mumble := concat('DATA AVERAGED OVER ',mumble,' POINT');
    if filternum > 0 then mumble := concat(mumble,'S ');
    moveto(10,30);
    wstring(mumble);
    str(scale,mumble);
    mumble := concat('SCALE: ',mumble);
    moveto(10,20);
    wstring(mumble);
    moveto(0,10);
    wstring(cmd[1]);
    moveto(0,0);
    wstring(cmd[2]);
end;
```

```
procedure redraw(curve:apfiletype;cursors:cursorstype;
         start,finish,place:loc;l,n:integer;
         filternum,scale:byte;nopoints:fract);

var
    m,xpp            :       integer;

begin{redraw}
    viewport(1,258,61,190);
    fillscreen(black);
    plotcurve(curve,start,finish,scale);
    viewport(0,279,0,191);
    grafmode;
    m := 1;
    while (m < n) and (l <> 5) do
    begin
        xpp := xp(cursors[m],start,nopoints);
        if (xpp>=lft) and (xpp<=rgt)
        then begin
            pencolor(none);
            moveto(xpp,bot);
            pencolor(white);
            moveto(xpp,top);
            pencolor(none);
        end;
        m := m + 1;
    end;
    titles(start,finish,n,filternum);
end;{redraw}
```

```
procedure plus(var result:loc;place:loc;expnse:integer);

var
    m       :       integer;

begin
    m := 256 * place.page + place.point;
    m := m + expnse;
    if m < 0 then m := 0;
    result.page := m div 256;
    m := m - result.page * 256;
    result.point := m;
end;

procedure changelimits(var start,finish:loc;
               expanse:integer;place:loc;
               curveexpanse:integer;
               curvestart,curvefinish:loc);

var
    front,rear,ctr   :       integer;

begin
    if expanse >= curveexpanse
    then begin
        start := curvestart;
        finish := curvefinish;
    end else begin
        front := diff(place,curvestart) + 1;
        rear := diff(curvefinish,place) + 1;
        ctr := expanse div 2;
        if front < ctr
        then begin
            start := curvestart;
            plus(finish,curvestart,(expanse - 1));
        end else begin
            if rear <= ctr
            then begin
                finish := curvefinish;
                plus(start,curvefinish,(1 - expanse));
            end else begin
                plus(start,place,(1 - ctr));
                plus(finish,place,(ctr));
            end;
        end;
    end;
end;
```

```
procedure getfac(var exfac:byte;name:string);

var
    ch          :       char;

begin
    cleartext;
    textmode;
    ch := chr(0);
    while not(ch in ['0','1','2','3','4','5','6','7','8','9']) do
    begin
        gotoxy(0,12);
        write('PLEASE TYPE THE ',name,' FACTOR: ');
        getc(ch);
    end;
    exfac := ord(ch) - ord('1') + 1;
    grafmode;
end;

procedure movecursors(l,n:byte;var filternum:byte;
      var cursors:cursorstype;var lastcursor,start,finish:loc;
      curveexpanse:integer;curvestart,curvefinish:loc);

var
    place       :       loc;
    ch          :       char;
    xpp         :       integer;
    nopoints    :       fract;
    i           :       integer;

procedure refuse(n:byte;var ch:char);

begin
    if n in [6,10] then
    begin
        ch := chr(0);
        cleartext;
        textmode;
        gotoxy(0,23);
        writeln('YOU MAY NOT REJECT THE ',
            labels[n],' CURSOR.');
        writeln;
        writeln('PLEASE PRESS RETURN.');
        readln;
        grafmode;
    end;
end;
```

```
procedure nozero(var ch:char);

begin
    if (n = 2) and (cursors[1] = place) then
    begin
        cleartext;
        textmode;
        gotoxy(0,23);
        write('YOU ARE ABOUT TO SET THE END OF THE FIRST FLAG ');
        writeln('EQUAL TO THE START OF THE FIRST');
        write('FLAG.  THE RESULT WILL BE THAT THE PROGRAM WILL');
        writeln('NOT BE ABLE TO CALCULATE INITIAL');
        write('VELOCITY.  INSTEAD THE PROGRAM WILL USE');
        writeln('THE KEYBOARD INPUT.  DO YOU STILL');
        while not(ch in ['Y','N']) do
        begin
            gotoxy(0,23);
            write('WANT TO ACCEPT?(Y/N) ');
            getc(ch);
        end;
        if ch = 'Y' then
            ch := 'A'
        else
            ch := chr(0);
        grafmode;
    end;
end;
```

```
procedure arrow(var place:loc;lastcursor,finish:loc;dist:byte;
        dir:boolean);

var
    m               :       byte;

begin
    if dir then
    begin
        m := 0;
        while (less(place,finish)) and (m<dist) do
        begin
            m := m + 1;
            ink(place);
        end;
    end else begin
        m := 0;
        while (less(lastcursor,place)) and (m<dist) do
        begin
            m := m + 1;
            dek(place);
        end;
    end;
end;

procedure setscale(var start,finish:loc;place:loc;
         l,n:integer;var nopoints:fract;var scale:byte);

var
    rear,front,
    expanse         :       integer;
    exfac           :       byte;
begin
    getfac(scale,'SCALE');
    expanse := diff(finish,start) + 1;
    setpoints(start,finish,nopoints);
    redraw(curve,cursors,start,finish,place,l,n,
           filternum,scale,nopoints);
end;
```

```
procedure contract(var start,finish:loc;place:loc;
         l,n:integer;var nopoints:fract);

var
    rear,front,
    expanse         :       integer;
    exfac           :       byte;
begin
    if (start <> curvestart) or (finish <> curvefinish)
    then begin
        getfac(exfac,'CONTRACTION');
        expanse := diff(finish,start) + 1;
        while (expanse < curveexpanse) and (exfac > 0) do
        begin
            expanse := expanse * 2;
            exfac := exfac - 1;
        end;
        changelimits(start,finish,expanse,place,
                 curveexpanse,curvestart,curvefinish);
        setpoints(start,finish,nopoints);
        redraw(curve,cursors,start,finish,place,l,n,
               filternum,scale,nopoints);
    end;
end;
```

```
procedure expand(var start,finish:loc;place:loc;
        l,n:integer;var nopoints:fract);

var
    rear,front,
    expanse         :       integer;
    exfac           :       byte;

begin

    if expanse > 8 then
    begin
        getfac(exfac,'EXPANSION');
        expanse := diff(finish,start) + 1;
        while (expanse > 8) and (exfac > 0) do
        begin
            expanse := expanse div 2;
            exfac := exfac - 1;
        end;
        changelimits(start,finish,expanse,place,
                 curveexpanse,curvestart,curvefinish);
        setpoints(start,finish,nopoints);
        redraw(curve,cursors,start,finish,place,l,n,
              filternum,scale,nopoints);
    end;
end;
```

```
procedure digfil(var curve:apfiletype;var filternum:byte;
         start,finish,place:loc;l,n:integer);

var
    newpt,current   :               loc;
    data,
    sum,pts,j,jnew  :               integer;
    temp            :               packed array[0..18] of byte;

begin
    getfac(filternum,'AVERAGING');
    curvi(curve);
    pencolor(none);
    moveto(55,118);
    wstring('Filtering.  Please wait.');
    if filternum >= 1 then
    begin
        newpt.page := 0;
        newpt.point := 0;
        pts := 2 * filternum + 1;
        sum := 0;
        for j := 0 to (pts - 1) do
        begin
            temp[j] := curve.page[newpt.page,newpt.point];
            sum := sum + temp[j];
            ink(newpt);
        end;
        j := filternum;
        current.page := 0;
        current.point := j;
        newpt.page := 0;
        newpt.point := 2 * filternum + 1;
        repeat
            curve.page[current.page,current.point] :=
              round(sum / pts);
            {It is critical that the quotient be rounded}
            {NOT truncated.                              }
            jnew := (j + filternum + 1) mod pts;
            sum := sum - temp[jnew];
            temp[jnew] := curve.page[newpt.page,newpt.point];
            sum := sum + temp[jnew];
            j := j + 1;
            ink(current);
            ink(newpt);
        until newpt.page >= curve.zeropage;
    end;
    redraw(curve,cursors,start,finish,place,l,n,
           filternum,scale,nopoints);
end;
```

```
procedure lleft(var start,finish:loc;var place:loc;
         l,n:integer);
var
    expanse         :       integer;
    temp            :       loc;

begin
    expanse := diff(finish,start) + 1;
    plus(temp,curvestart,(expanse div 2));
    if less(place,temp) then
    begin
        place := curvestart
    end else begin
        plus(place,place,(-1 * (expanse div 2)));
    end;
    changelimits(start,finish,expanse,place,
             curveexpanse,curvestart,curvefinish);
    redraw(curve,cursors,start,finish,place,l,n,
           filternum,scale,nopoints);
end;

procedure rright(var start,finish:loc;var place:loc;
         l,n:integer);
var
    expanse         :       integer;
    temp            :       loc;

begin
    expanse := diff(finish,start) + 1;
    plus(temp,curvefinish,(-1 * (expanse div 2)));
    if less(temp,place) then
    begin
        place := curvefinish;
    end else begin
        plus(place,place,(expanse div 2));
    end;
    changelimits(start,finish,expanse,place,
             curveexpanse,curvestart,curvefinish);
    redraw(curve,cursors,start,finish,place,l,n,
           filternum,scale,nopoints);
end;
```

```
begin{movecursors}
    titles(start,finish,n,filternum);
    setpoints(start,finish,nopoints);
    place := lastcursor;
    if (n=1) or (n=6)then
    begin
        for i := 1 to nopoints.num do
        begin
            ink(place);
        end;
        xpp := xp(place,start,nopoints);
        pencolor(none);
        moveto(xpp,bot);
        pencolor(reverse);
        moveto(xpp,top);
    end;
    ch := chr(0);
    while not(ch in ['A','D']
              8 : arrow(place,lastcursor,finish,1,false);
              21 : arrow(place,lastcursor,finish,1,true);
              60 : arrow(place,lastcursor,finish,jump,false);
              62 : arrow(place,lastcursor,finish,jump,true);
              65 : nozero(ch);
              67 : contract(start,finish,place,l,n,nopoints);
              68 : refuse(n,ch);
              69 : expand(start,finish,place,l,n,nopoints);
              70 : digfil(curve,filternum,start,
                   finish,place,l,n);
              76 : lleft(start,finish,place,l,n);
              82 : rright(start,finish,place,l,n);
              83 : setscale(start,finish,place,l,n,nopoints,
                   scale);
        end;
        xpp := xp(place,start,nopoints);
        pencolor(none);
        moveto(xpp,bot);
        pencolor(reverse);
        if xpp > xp(lastcursor,start,nopoints)
        then moveto(xpp,top);
    end;
```

```
    case ch of
        'A'     :       begin
                    cursors[n] := place;
                    lastcursor := place;
                end;
        'D'     :       begin
                    cursors[n].page := 255;
                    if place <> lastcursor
                    then begin
                        pencolor(reverse);
                        moveto(xpp,bot);
                    end
                end;
    end;
end;

procedure setlimits(var start,finish:loc;channel:char);

begin
    if ord(channel) >= 97 then channel := chr(ord(channel) - 32);
    case channel of
        'A'     :       begin
                    start.page := 0;
                    finish.page := (curve.zeropage
                    div 2) - 1;
                end;
        'B'     :       begin
                    start.page := curve.zeropage
                    div 2;
                    finish.page := curve.zeropage
                    - 1;
                end;
        'C'     :       begin
                    start.page := 0;
                    finish.page := curve.zeropage
                    - 1;
                end;
    end;
    start.point := 0;
    finish.point := 255;
end;
```

```
procedure timing(var cursors:cursorstype;channel:char;
         var filternum,scale:byte);
var
    min,max,
    n,l                       :       byte;
    curveexpanse              :       integer;
    lastcursor                :       loc;
    start,finish,
    curvestart,curvefinish    :       loc;
    nopoints                  :       fract;

begin{timing}
    setlimits(start,finish,channel);
    setpoints(start,finish,nopoints);
    curvestart := start;
    curvefinish := finish;
    curveexpanse := diff(curvefinish,curvestart) + 1;
    plotcurve(curve,curvestart,curvefinish,scale);
    moveto(52,50);
    wstring('CURSOR:  ');
    lastcursor := curvestart;
    l := 1;
    for n := 1 to 4 do
        movecursors(l,n,filternum,
        cursors,lastcursor,start,finish,
        curveexpanse,curvestart,curvefinish);
    initturtle;
end;{timing}
```

```
procedure impact(var cursors:cursorstype;channel:char;
         var filternum,scale:byte);

var
    min,max,
    n,l                          :       byte;
    curveexpanse                 :       integer;
    lastcursor                   :       loc;
    start,finish,
    curvestart,curvefinish       :       loc;
    nopoints                     :       fract;

begin{impact}
    case channel of
        'A'     :       channel := 'B';
        'B'     :       channel := 'A';
        'C'     :       channel := 'C';
    end;
    setlimits(start,finish,channel);
    setpoints(start,finish,nopoints);
    curvestart := start;
    curvefinish := finish;
    curveexpanse := diff(curvefinish,curvestart) + 1;
    plotcurve(curve,curvestart,curvefinish,scale);
    moveto(52,50);
    wstring('CURSOR:  ');
    lastcursor := curvestart;
    l := 6;
    for n := 6 to 10 do
        movecursors(l,n,filternum,cursors,lastcursor,start,
                finish,curveexpanse,curvestart,
                curvefinish);
     lastcursor := curvestart;
     l := 11;
     for n := 11 to 12 do
        movecursors(l,n,filternum,cursors,lastcursor,start,
                finish,curveexpanse,curvestart,
                curvefinish);
end;{impact}
```

```
begin{astirgraf}
(*$R turtlegraphics*)
   if (run.kind <> SLOW) and (channel <> 'C')
   then
       timing(cursors,channel,filternum,scale);
   impact(cursors,channel,filternum,scale);
   cleartext;
   textmode;
end;{astirgraf}

{-----------------------------------------------------------------------}
{   NOTE:  astirgraf is in danger of overwriting the Hi Res graphics   }
{          screen because of its size.  It should be rewritten for     }
{          example by breaking it into 2-3 segments.                   }
{-----------------------------------------------------------------------}
```

A.6.6 ASTIRDISP

```
segment procedure astirdisp(var disposables:factortype;var test:testtype);

       procedure displ(var disposables:factortype);

       var
               dispbook         :        file of factortype;

               procedure loadd(var disposables:factortype);

               begin
                       with disposables do
                       begin
                               flag1 := 0;
                               flag2 := 0;
                               velterm := 0;
                               gee := 32.170;
                               weight := 180.00;
                               gain := 1.000;
                               loadfac1 :=  +6.0000E-5;
                               loadfac2 :=  -1.1000E-9;
                               zerovolts := -1.1000E-1;
                       end;
               end;

               (*$I-*)
               procedure disci(var disposables:factortypes);

               begin
                       reset(dispbook,'HANDBOOK');
                       if ioresult = 0 then
                       begin
                               get(dispbook);
                               close(dispbook);
                               disposables := dispbook^;
                       end else begin
                               close(dispbook);
                               loadd(disposables);
                       end;
               end;
               (*$I+*)
```

```
procedure disco(disposables:factortypes);

begin
    rewrite(dispbook,'HANDBOOK');
    dispbook^ := disposables;
    put(dispbook);
    close(dispbook,lock);
end;

procedure screen1;

begin{screen1}
    cleartext;
    gotoxy(10,2);
    write('Disposable Constants');
    gotoxy(10,3);
    writeln('--------------------');
    writeln;
    with disposables do begin
        writeln('          FLAG1:     ',flag1);
        writeln;
        writeln('          FLAG2:     ',flag2);
        writeln;
        writeln('          VELTERM:   ',velterm);
        writeln;
        writeln('          GEE:       ',gee);
        writeln;
        writeln('          WEIGHT:    ',weight);
        writeln;
        writeln('          GAIN:      ',gain);
        writeln;
        writeln('          LOADFAC1:  ',loadfac1);
        writeln;
        writeln('          LOADFAC2:  ',loadfac2);
        writeln;
        writeln('          ZEROVOLTS: ',zerovolts);
        writeln;
    end;
    gotoxy(4,23);
    write('A(CCEPT),C(HANGE),L(AST),N(EXT)');
    if run.kind = PEND then write(',F(REESWING)');
    if run.kind = DROP then write(',F(REEDROP)');
    if run.kind = SLOW then write(',F(REESTROKE)');
    gotoxy(4,5);
    write('-->');
    gotoxy(2,23);
end;{screen1}
```

```
procedure control1(var disposables:factortypes);
{Because fo the depth of indentation, half tabs are used}
{in procedure control1                               }

var
    i,o          :       integer;
    ch           :       char;

    procedure change(var disposables:factortypes);

    procedure getfac(var xx:real);

    var
        diag,
        line     :   string;
        exponent:   boolean;
        temp     :   real;

    begin
        getline(line,'                    ',21,
        (2 * (1 + 1) + 1),19);
        exponent := false;
        sttofp(temp,diag,line,exponent);
        if diag = 'NG' then
        begin
        gotoxy(21,(2 * (1 + 1) + 1));
        write('INVALID: PRESS RTN.');
        readln;
        end else
        xx := temp;
    end;

    begin{change}
    with disposables do case i of
        1   :   getfac(flag1);
        2   :   getfac(flag2);
        3   :   getfac(velterm);
        4   :   getfac(gee);
        5   :   getfac(weight);
        6   :   getfac(gain);
        7   :   getfac(loadfac1);
        8   :   getfac(loadfac2);
        9   :   getfac(zerovolts);
    end;
```

```
gotoxy(21,(2 * (l + 1) + 1));
write('                    ');
gotoxy(21,(2 * (l + 1) + 1));
with disposables do case l of
    1   :   writeln(flag1);
    2   :   writeln(flag2);
    3   :   writeln(velterm);
    4   :   writeln(gee);
    5   :   writeln(weight);
    6   :   writeln(gain);
    7   :   writeln(loadfac1);
    8   :   writeln(loadfac2);
    9   :   writeln(zerovolts);
end;
end;{change}

procedure free(var disposables:factortypes);
{Because of the depth of indentation, half}
{tabs are used in this procedure          }

var
channel:char;
abdec:real;
{perhaps char and abdec should be passed to  }
{astircalc which needs them too and hence has}
{to calculate them...                        }

procedure getchannel(var channel:char);
{ ie find channel of impact data }

var
    abflag  :       byte;
    afinish :       loc;
```

```
begin
    afinish.page := (curve.zeropage div 2) - 1;
    afinish.point := 255;
    abflag := curve.page[curve.zeropage,170];
    case abflag of
    4   :   channel := 'A';
    8   :   channel := 'B';
       12   :   begin
                if less(afinish,cursors[10])
                then
                channel := 'B'
                else
                channel := 'A';
        end;
    end;
end;

procedure getabdec(var abdec:real);

var
    adec,bdec    :   byte;

begin
    adec := curve.page[curve.zeropage,135];
    bdec := curve.page[curve.zeropage,136];
    if channel = 'A' then
    begin
    if adec = 0 then abdec := 1.0000;
    if adec = 1 then abdec := 0.10000;
    end;
    if channel = 'B' then
    begin
    if bdec = 0 then abdec := 1.0000;
    if bdec = 1 then abdec := 0.10000;
    end;
end;

procedure setfree(var freeval:real);

var
    n        :       integer;
    place    :       loc;
```

```
    begin
        place := cursors[6];
        n := 1;
        freeval := (curve.page[place.page,place.point]
            - 128) / 128;
        while less(place,cursors[10]) do
        begin
        ink(place);
        freeval := freeval +
            (curve.page[place.page,place.point]
             - 128) / 128;
        n := n + 1;
        end;
        freeval := freeval / n;
        freeval := freeval * abdec * voltfac;
    end;

    begin{free}
    getchannel(channel);
    getabdec(abdec);
    setfree(disposables.zerovolts);
    gotoxy(21,21);
    write('                         ');
    gotoxy(21,21);
    writeln(disposables.zerovolts);
    end;{free}

procedure movearrow(o,l:integer);

begin
    gotoxy(4,(2 * (o + 1) + 1));
    write('   ');
    gotoxy(4,(2 * (l + 1) + 1));
    write('-->');
end;

begin{control1}
l := 1;
while ch <> 'A' do
begin
    ch := chr(0);
    while not(ch in ['A','C','F','L','N']) do
```

```
        begin
        o := 1;
        gotoxy(2,23);
        getc(ch);
        if (ord(ch) >= 32) and (ord(ch) <= 126) then
            write(ch);
        end;
        case ch of
        'A' :   begin
            end;
        'C' :   begin
                change(disposables);
            end;
        'F' :   begin
                free(disposables);
            end;
        'L' :   begin
                l := l - 1;
                if l < 1 then
                l :=  9;

            end;
        'N' :   begin
                l := l + 1;
                if l > 9 then
                l := 1;
            end;
        end;
        movearrow(o,l);
    end;
    cleartext;
    end;{control1}

begin{disp1}
    disci(disposables);
    screen1;
    control1(disposables);
    disco(disposables);
end;{disp1}
```

```
procedure disp2(var test:testtype);

var
    testbook        :       file of testtype;

    procedure screen2;

    begin{screen2}
        gotoxy(17,2);
        writeln('Test Conditions');
        gotoxy(17,3);
        write('---------------');
        with test do begin
            gotoxy(7,5);
            write('   SPECIMEN TYPE:      ');
            if kind = BEAM then write(' BEAM');
            if kind = CANTILEVER then write(' CANTILEVER');
            gotoxy(7,7);
            writeln('   INITIAL VELOCITY:   ',initvel);
            gotoxy(7,9);
            writeln('   FINAL VELOCITY:     ',finalvel);
            gotoxy(7,11);
            writeln('   TEST TEMPERATURE:   ',testtemp);
            gotoxy(7,13);
            writeln('   DIAL ENERGY:        ',dialnrg);
            gotoxy(7,15);
            writeln('   SPECIMEN LENGTH:    ',length);
            gotoxy(7,17);
            writeln('   SPECIMEN WIDTH:     ',width);
            gotoxy(7,19);
            writeln('   SPECIMEN THICKNESS:',thickness);
            gotoxy(7,21);
            writeln('   NOTCH+CRACK DEPTH: ',notchdepth);
            writeln;
            gotoxy(9,23);
            write('A(CCEPT),C(HANGE),L(AST),N(EXT)');
            gotoxy(6,5);
            write('-->');
            gotoxy(7,23);
            end;
    end;{screen2}
```

```
procedure disci(var test:testtype);

    procedure loadt(var test:testtype);

    begin
        with test do
        begin
            kind := CANTILEVER;
            initvel := 2;
            finalvel := -1;
            testtemp := -500;
            dialnrg := -1;
            length := 0.63;
            width := 0.70;
            thickness := 0.30;
            notchdepth := 0;
        end;
    end;
(*$I-*)
begin{disci}
    reset(testbook,'RESULTS');
    if ioresult = 0 then
    begin
        get(testbook);
        close(testbook);
        test := testbook^;
    end else begin
        close(testbook);
        loadt(test);
    end;
end{disci};
(*$I+*)

procedure control2(var test:testtype);
{Because fo the depth of indentation, half tabs are used}
{in procedure control2                                  }

var
    l,o     :       integer;
    ch      :       char;
```

```
procedure change(var test:testtype);
procedure getfac(var xx:real);

var
    diag,
    line    :   string;
    exponent:   boolean;
    temp    :   real;

begin
    getline(line,'                        ',
        29,(2 * (1 + 1) + 1),19);
    exponent := false;
    sttofp(temp,diag,line,exponent);
    if diag = 'NG' then
    begin
    gotoxy(29,(2 * (1 + 1) + 1));
    write('INVALID: PRESS RTN.');
    readln;
    end else
    xx := temp;
end;

begin{change}
with test do case 1 of
    1   :   case kind of
        CANTILEVER  :   kind := BEAM;
        BEAM        :   kind := CANTILEVER;
        end;
    2   :   getfac(initvel);
    3   :   getfac(finalvel);
    4   :   getfac(testtemp);
    5   :   getfac(dialnrg);
    6   :   getfac(length);
    7   :   getfac(width);
    8   :   getfac(thickness);
    9   :   getfac(notchdepth);
end;
```

```
    gotoxy(29,(2 * (l + 1) + 1));
    write('                    ');
    gotoxy(29,(2 * (l + 1) + 1));
    with test do case l of
        1   :   case kind of
            CANTILEVER  :   write(' CANTILEVER');
            BEAM        :   write(' BEAM');
            end;
        2   :   writeln(initvel);
        3   :   writeln(finalvel);
        4   :   writeln(testtemp);
        5   :   writeln(dialnrg);
        6   :   writeln(length);
        7   :   writeln(width);
        8   :   writeln(thickness);
        9   :   writeln(notchdepth);
    end;
    end;{change}

    procedure movearrow(o,l:integer);

    begin
    gotoxy(6,(2 * (o + 1) + 1));
    write('   ');
    gotoxy(6,(2 * (l + 1) + 1));
    write('-->');
    end;

begin{control2}
    l := 1;
    while ch <> 'A' do
    begin
    ch := chr(0);
    while not(ch in ['A','C','L','N']) do
    begin
        o := l;
        gotoxy(7,23);
        getc(ch);
        if (ord(ch) >= 32) and (ord(ch) <= 126) then
        write(ch);
    end;
```

```
        case ch of
            'A' :   begin
                end;
            'C' :   begin
                change(test);
                end;
            'L' :   begin
                l := l - 1;
                if l < 1 then
                    l :=  9;
                end;
            'N' :   begin
                l := l + 1;
                if l > 9 then
                    l := 1;
                end;
        end;
        movearrow(o,l);
        end;
        cleartext;
    end;{control2}

    procedure disco(var test:testtype);

    begin
        rewrite(testbook,'RESULTS');
        testbook^ := test;
        put(testbook);
        close(testbook,lock);
    end;

  begin{disp2}
    disci(test);
    screen2;
    control2(test);
    disco(test);
  end;{disp2}

begin{astirdisp}
  disp1(disposables);
  disp2(test);
end;{astirdisp}
```

A.6.7 ASTIRCALC

```
segment procedure astircalc(curve:apfiletype;cursors:cursorstype;
                            disposables:factortype;var abdec,vel1,vel2:real;
                            var channel:char;var nrg:nrgtype;var totals:duo);

var
       deltatee,
       mass,
       vcomp,
       zero                :       real;

       procedure getdelta(var deltatee:real);

       var
              ptsdiv,
              timptr,msec      :       byte;
              avenum           :       loc;
              methode,ch       :       char;
              division         :       real;

              procedure abort;

              begin
                     cleartext;
                     writeln('NO DIVISION SCALE DEFINED.');
                     exit(astir);
              end;

              procedure microrange(var deltatee,division:real);

              begin
                     case timptr of
                            80      :       division := 0.000500;
                            86      :       division := 0.000250;
                            92      :       division := 0.000200;
                            98      :       division := 0.000032;
                            104     :       division := 0.000016;
                            110     :       division := 0.000008;
                     end;
                     if curve.page[curve.zeropage,170] = 12
                     then begin
                            division := division * 2;
                     end;
                     deltatee := division / ptsdiv;
              end;
```

```
    procedure millirange(var deltatee,division:real);

    begin
        division := msec * 0.001;
        deltatee := division / ptsdiv;
    end;

    procedure secrange  (var deltatee,division:real);

    begin
        division := 0.1 * ((256 * avenum.page) + avenum.point);
        deltatee := division / ptsdiv;
    end;

begin{getdelta}
    ptsdiv := 28;
    timptr := curve.page[curve.zeropage,129];
    msec := curve.page[curve.zeropage,128];
    avenum.point := curve.page[curve.zeropage,191];
    avenum.page := curve.page[curve.zeropage,192];
    methode := 'N';{none}
    if timptr >= 80
    then
        methode := 'M'{micro}
    else
        if timptr < 4
        then
            methode := 'S'{sec}
        else
            if (timptr=4) or (timptr=5)
            then
                methode := 'L';{milli}
    case methode of
        'N' : abort;
        'S' : secrange(deltatee,division);
        'L' : millirange(deltatee,division);
        'M' : microrange(deltatee,division);
    end;
end;{getdelta}
```

```
procedure getchannel(var channel:char);
{ ie Find channel of impact data }

var
    abflag      :       byte;
    afinish     :       loc;

begin{getchannel}
    afinish.page := (curve.zeropage div 2) - 1;
    afinish.point := 255;
    abflag := curve.page[curve.zeropage,170];
    case abflag of
         4      :       channel := 'A';
         8      :       channel := 'B';
        12      :       begin
                    if less(afinish,cursors[10])
                    then
                        channel := 'B'
                    else
                        channel := 'A';
                end;
    end;
end;{getchannel}

procedure getabdec(var abdec:real);

var
    adec,bdec   :       byte;

begin{getabdec}
    adec := curve.page[curve.zeropage,135];
    bdec := curve.page[curve.zeropage,136];
    if channel = 'A' then
    begin
        if adec = 0 then abdec := 1.0000;
        if adec = 1 then abdec := 0.10000;
    end;
    if channel = 'B' then
    begin
        if bdec = 0 then abdec := 1.0000;
        if bdec = 1 then abdec := 0.10000;
    end;
end;{getabdec}
```

```
procedure fetchvels(var vel1,vel2,vcomp:real;cursors:cursorstype);

var
    a,cflag1, cflag2, cflspc        :        real;

    procedure corflgs(var cflag1,cflag2,cflspc:real);
    { corflags takes care of corrections to flag width. }

    begin
        with disposables do
        begin
            cflag1 := (flag1 + velterm) / 12;
            cflag2 := (flag2 + velterm) / 12;
        end;
        { The divisor of 12 accounts for the fact that the }
        { flags are given in inches but are needed in feet.}
    end;

    procedure calcvels(var time1,tdiff,time3,vel1,vel2,vcomp:real);

    begin
        vel1 := -1;
        vel2 := -1;
        vcomp := -1;
        time1 := -1;
        tdiff := -1;
        time3 := -1;
        if (cursors[1].page <> 255) and (cursors[2].page<>255)
        then begin
            if (cursors[1]<>cursors[2])
            then
                time1 := deltatee
                      * diff(cursors[2],cursors[1]);
            if (cursors[2]<>cursors[3]) and
               ((cursors[3].page<>255)
                and (cursors[4].page<>255))
            then
                tdiff := deltatee
                      * diff(cursors[4],cursors[3]);
            if (cursors[1]<>cursors[4])
               and (cursors[4].page<>255)
            then
                time3 := deltatee
                      * diff(cursors[4],cursors[1]);
        end;
```

```
    if time1 <> -1
    then
        vel1 := cflag1 / time1 + (a * time1) / 2;
    if tdiff <> -1
    then
        vel2 := cflag2 / tdiff + (a * tdiff) / 2;
    if (vel1<> -1) and ((time1<> -1) and (time3<> -1))
    then
        vcomp := vel1 + a * (time3 - time1);
    cleartext;
end;

procedure getvels(var vel1,vel2,vcomp:real;
cursors:cursorstype);

var
    time1,tdiff,time3        :        real;

begin
    calcvels(time1,tdiff,time3,vel1,vel2,vcomp);
    if time1 < 0 then
    { CALCVELS DIDN'T EVEN CALCULATE TIME1 }
    { USE KEYBOARD INPUT.                  }
    begin
        vel1 := test.initvel;
        vel2 := test.finalvel;
        { IT MAY STILL BE POSSIBLE TO RECOVER. }
        if run.kind = PEND
        then
            vcomp := vel1
        else
            if (time1 <> -1) and (time3 <> -1)
            then
                vcomp := vel1
                     + a * (time3 - time1);
    end;
end;
```

```
begin{fetchvels}
    vel1 := -1;
    vel2 := -1;
    vcomp := -1;
    corflgs(cflag1,cflag2,cflspc);
    case run.kind of
        DROP    :       begin
                    a := disposables.gee;
                    getvels(vel1,vel2,vcomp,
                        cursors);
                end;
        PEND    :       begin
                    a := 0;
                    getvels(vel1,vel2,vcomp.
                        cursors);
                end;
        SLOW    :       begin
                end;
    end;
end;{fetchvels}

procedure calcmass(var mass:real);

begin
    mass := disposables.weight / disposables.gee;
end;

procedure getzero(var zero:real);

var
    place   :       loc;
    volts   :       real;
    count   :       integer;

begin{getzero}
    if (cursors[11].page = 255) or (cursors[12].page = 255)
    then begin
        zero := disposables.zerovolts;
        writeln('zero from disposables:  ',zero);
    end else begin
        place := cursors[11];
        count := 1;
        zero := (curve.page[place.page,place.point] - 128)
             / 128;
        while less(place,cursors[12]) do
```

```
        begin
            ink(place);
            zero := zero + (curve.page[place.page,place.point]
                    - 128) / 128;
            count := count + 1;
        end;
        zero := zero / count;
        zero := zero * abdec * voltfac;
        writeln('zero from curve:  ',zero);
        writeln('count:  ',count);
    end;
end;{getzero}

procedure calcnrg(var nrg:nrgtype);

var
    i       :       byte;
    v0,v1,
    xold,
    p0,p1,
    time,
    defl,
    energy,
    k1,k2,
    k3,k4   :       real;
    xplace,
    place   :       loc;

    procedure getload(var load:real);

    var
        volts   :       real;

    begin{getload}
        with disposables do
        begin
            volts := voltfac * abdec *
                   (curve.page[place.page,place.point] - 128)
                    / 128;
            if loadfac2 <> 0 then
```

```
        begin
            load := k1
                + k4 * sqrt(k2 + k3 * volts);
            {quadratic calibration curve}
        end else begin
            load := (volts - zero)
                 / (loadfac1 * gain);
            {linear calibration curve}
        end;
    end;
end;{getload}

procedure startslow(var p0,p1,v0,v1:real;var place:loc);

var
    volts  :  real;

begin{startslow}
    if channel = 'A'
    then
        xplace.page := place.page + 2
    else
        xplace.page := place.page - 2;
    if xplace.point > 0 then dek(xplace);
    { IF XPLACE IS THE FIRST POINT IN THE CURVE,    }
    { THEN THE PROGRAM WOULD CRASH WITHOUT THE IF. }
    { WITH THE IF THERE IS A POSSIBILITY OF A       }
    { SMALL ERROR BUT ONLY IF XPLACE IS THE FIRST   }
    { POINT IN THE CURVE                            }
    volts := voltfac * abdec *
            (curve.page[xplace.page,xplace.point]
             - 128) / 128;
    xold := volts * disposables.velterm;
end;{startslow}

procedure slownrg(var p0,p1,v0,v1:real;var place:loc;
          var time,defl,energy:real;var nrg:nrgtype);

var
    xnew,
    deltax,
    volts,
    deltanrg        :       real;
```

```
begin
    while less(place,cursors[L]) do
    with disposables do
    begin
        ink(place);
        ink(xplace);
        volts := voltfac * abdec *
            (curve.page[place.page,place.point]
             - 128) / 128;
        p0 := (volts - zerovolts) * loadfacl / gain ;
        volts := voltfac * abdec *
              (curve.page[xplace.page,xplace.point]
               - 128) / 128;
        xnew := volts * velterm;
        deltax := xnew - xold;
        deltanrg := p0 * deltax;
        defl := defl + deltax;
        energy := energy + deltanrg;
        time := time + deltatee;
        xold := xnew;
    end;
    nrg[1,L] := p0;
    nrg[2,L] := time;
    nrg[3,L] := defl;
    nrg[4,L] := energy;
end;

procedure startdrop(var p0,p1,v0:real;var v1:real;
                    var place:loc);

begin{startdrop}
    ink(place);
    getload(p1);
end;{startdrop}

procedure dropnrg(var p0,p1,v0,v1:real;var place:loc;
          var time,defl,energy:real;
          var nrg:nrgtype);

var
    ea,e0,
    i1,i2,i3,
    p2,v2,v3        :       real;
    halt            :       loc;
```

```
    {-----------------------------------------------}
    {ea and e0 are as defined in the derivation of the}
    {Augland-Grumbach equation.  i1 is the impulse    }
    {after the first interval.  i2 is the impulse     }
    {after the second interval.  i3 is the impulse    }
    {after the third interval.                        }
    {-----------------------------------------------}

begin{dropnrg}
    halt := cursor[1];
    ink(halt);
    {-----------------------------------------------}
    {If this is not done, one interval at the end of  }
    {the calculation will not be calculated (trivial) }
    {and the proper load at the cursors will not be   }
    {reported (possibly very important).              }
    {-----------------------------------------------}
    while less(place,halt) do
    begin
        ink(place);
        getload(p2);
        i1 := (-p2 + 8*p1 + 5*p0) * deltatee / 12;
        i2 := (p2 + 4*p1 + p0) * deltatee / 3;
        i3 := (9*p2 + 3*p0) * deltatee / 4;
        with disposables do
        begin
            v1 := v0 - i1 / mass
                + gee * deltatee;
            v2 := v0 - i2 / mass
                + 2 * gee * deltatee;
            v3 := v0 - i3 / mass
                + 3 * gee * deltatee;
        end;
    {-----------------------------------------------}
    {The third term of each velocity calculation takes}
    {into account the effect of gravity but only on   }
    {velocity and deflection.  The energy correction  }
    {is handled below.                                }
    {-----------------------------------------------}
        defl := defl +
            (v3-5*v2+19*v1+9*v0) * deltatee / 24;
        ea := v0 * i1;
        e0 := mass * v0 * v0 / 2;
        energy := energy + ea - ((ea * ea)/(4 * e0));
        with disposables do
            energy := energy +
                (-p2 + 10*p1 + 3*p0) * gee *
                deltatee * deltatee / 24;
```

```
    {---------------------------------------------}
    {Thus the energy correction.                  }
    {---------------------------------------------}
        time := time + deltatee;
        p0 := p1;
        p1 := p2;
        v0 := v1;
    end;
end;{dropnrg}

procedure startpend(var p0,p1,v0,v1:real;var place:loc);

begin{startpend}
    ink(place);
    getload(p1);
end;{startpend}

procedure pendnrg(var p0,p1,v0,v1:real;var place:loc;
          var time,defl,energy:real;
          var nrg:nrgtype);

var
    ea,e0,
    i1,i2,i3,
    p2,v2,v3        :       real;
    halt            :       loc;
    {---------------------------------------------}
    {ea and e0 are as defined in the derivation of the}
    {Augland-Grumbach equation.  i1 is the impulse    }
    {after the first interval.  i2 is the impulse     }
    {after the second interval.  i3 is the impulse    }
    {after the third interval.                        }
    {---------------------------------------------}

begin{pendnrg}
    halt := cursor[1];
    ink(halt);
    {---------------------------------------------}
    {If this is not done, one interval at the end of  }
    {the calculation will not be calculated (trivial) }
    {and the proper load at the cursors will not be   }
    {reported (possibly very important).              }
    {---------------------------------------------}
```

```
    while less(place,halt) do
    while less(place,cursors[l]) do
    begin
        ink(place);
        getload(p2);
        i1 := (-p2 + 8*p1 + 5*p0) * deltatee / 12;
        i2 := (p2 + 4*p1 + p0) * deltatee / 3;
        i3 := (9*p2 + 3*p0) * deltatee / 4;
        v1 := v0 - i1 / mass;
        v2 := v0 - i2 / mass;
        v3 := v0 - i3 / mass;
        defl := defl +
            (v3-5*v2+19*v1+9*v0) * deltatee / 24;
        ea := v0 * i1;
        e0 := mass * v0 * v0 / 2;
        energy := energy + ea - ((ea * ea)/(4 * e0));
        time := time + deltatee;
        p0 := p1;
        p1 := p2;
        v0 := v1;
    end;
end;{pendnrg}

procedure startcalc(var p0,p1,v0,v1:real;var place:loc;
            var time,defl,energy,k1,k2,k3,k4:real;
            var l:integer);
begin{startcalc}
    place := cursors[6];
    with disposables do
    begin
        if loadfac2 <> 0 then
        begin
            k1 := -loadfac1 / (2 * loadfac2);
            k2 := (loadfac1 / loadfac2)
                   * (loadfac1 / loadfac2);
            k2 := (k2 / 4) - (zero / loadfac2);
            k3 := 1 / loadfac2;
            k4 := 1.0000;
            if k3 < 0 then
                k4 := - k4;
        end;
    end;
```

```
        l := 6;
        v0 := vell;
        getload(p0);
        case run.kind of
            DROP : startdrop(p0,p1,v0,v1,place);
            PEND : startpend(p0,p1,v0,v1,place);
            SLOW : startslow(p0,p1,v0,v1,place);
        end;
        energy := 0;
        time := 0;
        defl := 0;
    end;{startcalc}

    procedure getnextcursor(var l:byte);

    begin
        l := l + 1;
        while (l < 10) and (cursors[l].page = 255) do
        begin
            l := l + 1;
        end;
    end;

begin{calcnrg}
    startcalc(p0,p1,v0,v1,place,time,defl,energy,k1,k2,k3,k4,l);
    while l < 10 do
    begin
        getnextcursor(l);
        if cursors[l].page <> 255
        then begin
            case run.kind of
                DROP : dropnrg(p0,p1,v0,v1,place,
                           time,defl,energy,nrg);
                PEND : pendnrg(p0,p1,v0,v1,place,
                           time,defl,energy,nrg);
                SLOW : slownrg(p0,p1,v0,v1,place,time,
                           defl,energy,nrg);
            end;
        end;
    end;
end;{calcnrg}
```

```
   procedure calctotals(var totals:duo);

   begin
       totals[1] := test.dialnrg;
       totals[2] := -1;
       if (vcomp>0) and (vel2>0)
       then begin
           totals[2] := (1 / 2) * mass * ((vcomp * vcomp) -
                        (vel2 * vel2))
       end;
   end;

begin{astircalc}
(*$R transcend*)
   getdelta(deltatee);
   getchannel(channel);
   getabdec(abdec);
   fetchvels(vel1,vel2,vcomp,cursors);
   calcmass(mass);
   getzero(zero);
   calcnrg(nrg);
   calctotal(totals);
end;{astircalc}
```

A.6.8 ASTIRREPT

```
segment procedure
astirrept(curve:apfiletype;cursors:cursorstype;nrg:nrgtype;
                            heading:ident;totals:duo;var next:boolean;
                            vel1,vel2:real;filternum,scale:byte);

var
       ch              :       char;
       outprint        :       text;

procedure bill(x:real;noplaces,len:byte;var result:string);

       var
               i,id    :       integer;
               dec     :       real;
               si,sid  :       string;

       begin
               if abs(x) <= 32767 then
               begin
                       if noplaces > 4 then noplaces := 4;
                       i := trunc(x);
                       dec := (x - i + 1) * pwroften(noplaces);
                       id  := trunc(dec);
                       str(i,si);
                       str(id,sid);
                       delete(sid,1,1);
                       result := concat(si,'.',sid);
               end else begin
                       result := 'OVERFLOW';
               end;
               if length(result) > len then
                       result := 'TOO LONG';
               if ((result = 'TOO LONG') or (result = 'OVERFLOW')) and
                  (length(result) > len) then
                  result := copy(result,1,len);
               while length(result) < len do
               begin
                       result := concat(' ',result);
               end;
       end;
```

```
procedure table;

var
    l        :        byte;
    number   :        string;
begin
    {headings}
    write(outprint,'         FEATURE          LOAD        ');
    writeln(outprint,' TIME      DEFLECTION     ENERGY');
    write(outprint,'                          [Lb]        ');
    writeln(outprint,'[mSec]       [In]       [Ft-Lb]');
    write(outprint,'      ---------------------------------');
    writeln(outprint,'---------------------------------');
    writeln(outprint);
    {data}
    for l := 7 to 10 do
    begin
        if cursors[l].page <> 255
        then begin
            str((l - 6),number);
            write(outprint,'     ',number,
            ' ',labels[l],'      ');
            if l = 10
            then begin
                write(outprint,' -----       ');
            end else begin
                bill(nrg[1,l],0,6,number);
                write(outprint,number,'       ');
            end;
            bill((1000 * nrg[2,l]),3,6,number);
            write(outprint,number,'       ');
            bill((nrg[3,l] * 12),4,6,number);
            {-------------------------------------}
            { VEL IS IN FT/SEC SO DEFL IS IN FEET }
            { BUT IT IS WANTED IN INCHES          }
            {-------------------------------------}
            write(outprint,number,'      ');
            bill(nrg[4,l],1,6,number);
            writeln(outprint,number);
            writeln(outprint);
        end;
    end;
end;
```

```
procedure printtotals;

var
    number  :       string;

begin
    if run.kind in [DROP, PEND] then
    begin
        write(outprint,'                                              ');
        writeln(outprint,'TOTAL ENERGY');
        writeln(outprint);
        if run.kind = PEND then
        begin
            write(outprint,'                                      ');
            write(outprint,'   DIAL: ');
            if totals[1] >= 0
            then begin
                bill(totals[1],1,6,number);
                writeln(outprint,number,' [Ft-Lb]');
            end else begin
                writeln(outprint,'______ [Ft-Lb]');
            end;
        end;
        write(outprint,'                                       FLAG: ');
        if totals[2] >= 0
        then begin
            bill(totals[2],1,6,number);
            writeln(outprint,number,' [Ft-Lb]');
        end else begin
            writeln(outprint,'______ [Ft-Lb]');
        end;
        write(outprint,'                                        TUP: ');
        if nrg[4,10] < 0 then
        begin
            write('-');
            bill(-nrg[4,10],1,6,number)
        end else begin
            bill(nrg[4,10],1,6,number);
        end;
        writeln(outprint,number,' [Ft-Lb]');
        writeln(outprint);
    end;
end;
```

```
procedure printest;

var
    mumble      :       string;

begin
    writeln(outprint,'DESIGNATION: ',heading);
    write(outprint,'TEST METHOD: ');
    case run.kind of
        DROP    :    writeln(outprint,
                     'INSTRUMENTED DROP TOWER IMPACT');
        PEND    :    writeln(outprint,
                    'INSTRUMENTED PENDULUM IMPACT');
        SLOW    :    writeln(outprint,
                    'SLOW BEND');
    end;
    if test.testtemp >= -459.69
    then begin
        bill(test.testtemp,0,5,mumble);
        writeln(outprint,'TEMPERATURE: ',mumble,' [F]');
    end;
end;

procedure printall;

begin
    writeln(outprint);
    table;
    writeln(outprint);
    printtotals;
end;
```

```
   procedure makecopy;

      procedure hardcopy(var outprint:text);

      begin
         rewrite(outprint,'PRINTER:');
         printest;
         writeln(outprint,chr(25),'GLD');
         writeln(outprint);
         printall;
         close(outprint);
      end;

   begin
      moveto(0,10);
      wstring(cmd[0]);
      hardcopy(outprint);
      moveto(0,10);
      wstring(cmd[2]);
   end;

begin{astirrept}
   (*$R turtlegraphics*)
   rewrite(outprint,'CONSOLE:');
   printest;
   printall;
   close(outprint);
   gotoxy(2,23);
   write(cmd[1]);
   textmode;
   ch := chr(0);
   while not(ch in ['C','F']) do
   begin
      ch := chr(0);
      while not(ch in ['C','F','G','H','Q','T']) do
      begin
         gotoxy(0,23);
         getc(ch);
      end;
      case ch of
            'H'     :      makecopy;
            'G'     :      grafmode;
            'T'     :      textmode;
            'C'     :      next := true;
            'F'     :      textmode;
            'Q'     :      exit(astir);
      end;
   end;
end;{astirrept}
```

A.6.9 The ASTIR Libraries

A.6.9.1 ASTIRSEGS

```
segment procedure drawcurve(curve:apfiletype;start,finish:loc;
               nopoints:fraction);

var
   zeroypp,
   xpp,ypp              :       integer;
   place                :       loc;
   mag                  :       fract;
   temp                 :       integer[6];
   zerofloat            :       real;
   min,max              :       byte;

   procedure setmag(curve:apfiletype;start,finish:loc;
            var mag:fract;var min,max:byte;
            var zerofloat:real);
   var
       strt,expanse     :       integer;
       zerobyte         :       byte;
   begin
       expanse := diff(finish,start) + 1;
       strt := 256*start.page + start.point;
       minimax(curve,strt,expanse,min,max);
       zerofloat := (128 * disposables.zerovolts)/
                (voltfac * abdec) + 128;
       if zerofloat >= 255 then zerofloat := 255
       else if zerofloat <= 0 then zerofloat:= 0;
       zerobyte := trunc(zerofloat);
       if run.user = SUPER then
       begin
           textmode;
           writeln('ZEROFLOAT = ',zerofloat);
           writeln('ZEROBYTE =  ',zerobyte);
           writeln('MIN = ',min);
           writeln('MAX = ',max);
           writeln('ABDEC = ',abdec);
           readln;
           grafmode;
       end;
```

```
        if max < zerobyte then max := zerobyte;
        if min > zerobyte then min := zerobyte;
        if run.user = SUPER then
        begin
            textmode;
            writeln('ZEROFLOAT = ',zerofloat);
            writeln('ZEROBYTE =  ',zerobyte);
            writeln('MIN = ',min);
            writeln('MAX = ',max);
            readln;
            grafmode;
        end;
        mag.num := 126;
        mag.den := max - min;
        if mag.den = 0 then mag.den := 1;
    end;

(*$R-*)
begin{drawcurve}
    (*$R turtlegraphics*)
    initturtle;
    moveto(0,128);
    pencolor(white1);
    moveto(0,top+2);
    moveto(237,top+2);
    moveto(237,bot-2);
    moveto(0,bot-2);
    moveto(0,128);
    viewport(lft,rgt,bot,top);
    setmag(curve,start,finish,mag,min,max,zerofloat);
    zeroypp := round((mag.num * (zerofloat - min) / mag.den) + bot);
    pencolor(none);
    moveto(0,zeroypp);
    pencolor(green);
    moveto(237,zeroypp);
    place := start;
    temp := ((diff(place,start))*nopoints.den)
              div nopoints.num + lft;
    xpp := trunc(temp);
    ypp := (mag.num * (curve.page[place.page,place.point]
          - min) div (mag.den)) + bot ;
    pencolor(none);
    moveto(xpp,ypp);
    pencolor(white);
    while less(place,finish) do
```

```
   begin
       ink(place);
       temp := ((diff(place,start))*nopoints.den)
                  div nopoints.num + lft;
       xpp := trunc(temp);
       ypp := (mag.num * (curve.page[place.page,place.point]
        - min) div (mag.den)) + bot;
       moveto(xpp,ypp);
   end;
   pencolor(none);
   viewport(0,279,0,191);
end;{drawcurve}
(*$R+*)


segment procedure startgraf(var cmd:cmds;var labels:labelstype;
               var filternum,scale:byte;curve:apfiletype;
               var cursors:cursorstype;var channel:char);

   procedure curvo(curve:apfiletype);

   var
       curcurve                :       file of sector;
       l                       :       byte;

   begin
       rewrite(curcurve,'CURRENT');
       for l := 0 to 3 do
       begin
           curcurve^ := curve.page[l];
           put(curcurve);
       end;
       close(curcurve,lock);
   end;
```

```
procedure setlabels(var cmd:cmds;var labels:labelstype);

begin
    labels[1] := 'START FIRST FLAG';
    labels[2] := 'END FIRST FLAG';
    labels[3] := 'START SECOND FLAG';
    labels[4] := 'END SECOND FLAG';
    labels[5] := 'ZERO LEVEL';
    labels[6] := 'START';
    labels[7] :=  'GENERAL YIELD';
    labels[8] :=  'MAXIMUM LOAD ';
    labels[9] :=  'FAST FRACTURE';
    labels[10] := 'END OF EVENT ';
    labels[11] := 'START ZERORANGE';
    labels[12] := 'END ZERORANGE';
    cmd[0] := '                                              ';
    cmd[1] := 'A(CCEPT),D(ELETE),F(ILTER),E(XPAND),<,>,';
    cmd[2] := 'C(ONTRACT),S(CALE),R(IGHT),L(EFT),<-,-> ';
end;

procedure starttiming(var cursors:cursorstype;var channel:char);

var
    min,max,
    n               :       byte;
    start,finish    :       loc;
    nopoints        :       fraction;

    procedure setlimits(var start,finish:loc;channel:char);

        begin
            if ord(channel) >= 97
            then
                channel := chr(ord(channel)-32);
            case channel of
                'A' :   begin
                        start.page := 0;
                        finish.page :=
                          (curve.zeropage
                           div 2) - 1;
                    end;
```

```
              'B' :   begin
                      start.page :=
                        (curve.zeropage
                         div 2);
                      finish.page :=
                        (curve.zeropage)
                         - 1;
                  end;
              'C' :   begin
                      start.page := 0;
                      finish.page :=
                        (curve.zeropage)
                         - 1;
                  end;
          end;
          start.point := 0;
          finish.point := 255;
      end;

      procedure setpoints(start,finish:loc;
                  var nopoints:fraction);

      begin
          nopoints.num := diff(finish,start);
          nopoints.den := 236;
      end;

begin{starttiming}
(*$R turtlegraphics*)
initturtle;
pencolor(none);
textmode;
    channel := 'C';
    if curve.page[curve.zeropage,170] <> 12
    then begin
        for n := 1 to 4 do
        begin
            cursors[n].page := 255;
            { SINCE VELOCITY WILL BE BY       }
            { KEYBOARD, REGARDLESS OF RUN,    }
            {           CURSORS ARE REJECTED }
        end;
```

```
    end else begin
        cleartext;
        textmode;
        if run.velocity = curv then
        begin
            setlimits(start,finish,'C');
            setpoints(start,finish,nopoints);
            plotcurve(curve,start,finish,scale);
            moveto(36,175);
            wstring('CHANNEL A');
            moveto(163,175);
            wstring('CHANNEL B');
            while not(channel in ['A','B']) do
            begin
                moveto(0,8);
                wstring('PLEASE TYPE THE');
                moveto(16,8);
                wstring(' CHANNEL WHERE TIMING');
                moveto(0,0);
                wstring('INFORMATION CAN BE FOUND:  ');
                grafmode;
                getc(channel);
            end;
        end else begin
            for n := 1 to 4 do
            begin
                cursors[n].page := 255;
                { SINCE VELOCITY WILL BE BY      }
                { KEYBOARD, CURSORS ARE REJECTED }
            end;
        end;
    end;
end;
```

```
begin{startgraf}
(*$R turtlegraphics*)
   if run.user = SUPER then
   begin
       writeln('STARTED STARTGRAF');
       readln;
   end;
   setlabels(cmd,labels);
   filternum := 0;
   scale := 0;
   curvo(curve);
   starttiming(cursors,channel);
   initturtle;
   if run.user = SUPER then
   begin
       textmode;
       writeln('FINISHED STARTGRAF');
       readln;
       grafmode;
   end;
end;{startgraf}

segment procedure curvi(var curve:apfiletype);

var
   curcurve                  :       file of sector;
   l                         :       byte;

begin
   (*$I-*)
   reset(curcurve,'CURRENT');
   if ioresult = 0 then
   begin
       for l := 0 to 3 do
       begin
           curve.page[l] := curcurve^;
           get(curcurve);
       end;
       close(curcurve);
   end else begin
       close(curcurve);
       cleartext;
       textmode;
       writeln('CURVE LOST. PRESS RETURN TO ABORT.');
       readln;
       exit(astir);
   end;
   (*$I+*)
end;
```

```
segment procedure startrept(var cmd:cmds;scale:byte);

   procedure resetcmd(var cmd:cmds);

   begin
       cmd[0] := '                                                    ';
       cmd[1] := 'H(ARD COPY),G(RAPHICS),C(ONT),F(ACTORS),Q(UIT)';
       cmd[2] := 'H(ARD COPY),T(EXT),C(ONT),F(ACTS),Q(UIT)';
   end;

   procedure drawpic;

   var
       i                :       byte;
       nopoints         :       fraction;
       start,finish     :       loc;
       xpp              :       integer;
       mumble           :       string;

       procedure setpoints(start,finish:loc;var nopoints:fraction);

       begin
           nopoints.num := diff(finish,start);
           nopoints.den := 236;
       end;

       function xp(x,start:loc;nopoints:fraction):integer;

       var
           temp         :       integer[6];
           tmp          :       integer;

       begin
           temp:=((diff(X,start)) * nopoints.den)
               div nopoints.num
               + lft;
           xp := trunc(temp);
       end;
```

```
begin{drawpic*}
    initturtle;
    start := cursors[6];
    finish := cursors[10];
    dek(start);
    ink(finish);
    setpoints(start,finish,nopoints);
    drawcurve(curve,start,finish,nopoints);
    pencolor(none);
    chartype(14);
    for l := 7 to 9 do
    begin
        xpp := xp(cursors[l],start,nopoints);
        if (xpp>=0) and (xpp<=236)
        then begin
            pencolor(none);
            moveto(xpp,bot);
            pencolor(white);
            moveto(xpp,top);
            pencolor(none);
            str((l-6),mumble);
            moveto(xpp-3,50);
            wstring(mumble);
        end;
    end;
    chartype(10);
    pencolor(none);
    moveto(0,10);
    wstring(cmd[2]);
    str((2*filternum+1),mumble);
    mumble := concat('DATA AVERAGED OVER ',mumble,' POINT');
    if filternum > 0 then mumble := concat(mumble,'S  ');
    moveto(0,20);
    wstring(mumble);
    str((diff(cursors[10],cursors[6]) + 1),mumble);
    mumble := concat('USEFUL POINTS: ',mumble,'    ');
    moveto(0,30);
    wstring(mumble);
end;{drawpic}
```

```
begin{startrept}
(*$R turtlegraphics*)
   resetcmd(cmd);
   drawpic;
   cleartext;
   gotoxy(0,23);
end;{startrept}
```

A.6.9.2 TYPESTUFF

```
(*$S+*)
unit typestuff; intrinsic code 25;

interface

const
   maxpage          =        4;{room for 4 pages of curve & zero page}
   maxent           =       75;
   datadisc         =        5;
   bs               =        8;
   fs               =       21;
   jump             =       16;
   voltfac          =        9.08;
   rgt              =      257;
   lft              =        2;
   top              =      189;
   bot              =       62;
type
   byte             =      0..255;
   charimage        =      packed array[0..7] of byte;
   charset          =      packed array[0..127] of charimage;
   charfile         =      file of charset;
   loc              =      packed recordi
                    page    :       byte;
                    point   :       byte;
               end;
   sector           =      packed array[0..255] of byte;
   block            =      packed array[0..1] of sector;
   ident            =      packed array[0..29] of char;
   catent           =      packed record
                    trk,sct,filetype,length  :  byte;
                    name                     :  ident;
               end;
   entrytype        =      packed array[0..maxent] of catent;
   cattype          =      packed record
                    entries :       entrytype;
                    length  :       byte;
               end;
```

```
pg              =       packed array[0..maxpage] of sector;
apfiletype      =       packed record
                page    :       pg;
                zeropage:       byte;
            end;
labelstype      =       packed array[1..12] of string;
cursorstype     =       packed array[1..12] of loc;
fract           =       packed record
                num,den :       integer;
                end;
fraction        =       packed record
                num     :       integer[6];
                den     :       integer[6];
            end;
nrgtype         =       packed array[1..4,6..10] of real;
duo             =       packed array[1..2] of real;

cmds            =       packed array[0..2] of string;

methodtype      =       (PEND, DROP, SLOW);
specimentype    =       (BEAM,CANTILEVER);
usertype        =       (NORMAL,PRO,SUPER);
veltype         =       (KEYB,CURV);
printype        =       (EPSON,IMAGE);
factortype      =       packed record
                flag1,
                flag2,
                velterm,
                gee,
                weight,
                gain,
                loadfac1,
                loadfac2,
                zerovolts       :       real;
            end;
testtype        =       packed record
                kind            :       specimentype;
                initvel,
                finalvel,
                testtemp,
                dialnrg,
                length,
                width,
                thickness,
                notchdepth      :       real;
            end;
```

```
   runtype          =       packed record
                    kind    :       methodtype;
                    velocity:       veltype;
                    user    :       usertype;
                  end;

   procedure cleartext;
   procedure getc(var ch:char);
   procedure getline(var line:string;inn:string;x,y,limit:integer);
   procedure sttofp(var xx:real;var diag:string;s:string;
            var exponent:boolean);
   function  less(op1,op2:loc):boolean;
   function  diff(larger,smaller:loc):integer;
   procedure ink(var point:loc);
   procedure dek(var point:loc);

implementation

   procedure cleartext;

   begin
       write(chr(12));
   end;

   procedure getc;
   external;

   procedure getline;

   var
       o       :       integer;
       ch      :       char;
       cha     :       string[1];

       procedure rgt(var line,inn:string;var x:integer);

       begin
           if length(inn) >= 1 then
           begin
               line := concat(line,copy(inn,1,1));
               delete(inn,1,1);
               x := x + 1;
           end else begin
               write(chr(7));
           end;
       end;
```

```
    procedure lft(var line,inn:string;var x:integer);

    begin
        if length(line) >= 1 then
        begin
            inn := concat(copy(line,length(line),1),inn);
            delete(line,length(line),1);
            x := x - 1;
        end else begin
            write(chr(7));
        end;
    end;

begin{getline}
    while (length(inn) < limit) and (length(inn) <= 79) do
    begin
        inn := concat(inn,' ');
    end;
    inn := copy(inn,1,limit);
    gotoxy(x,y);
    write(inn);
    ch := chr(0);
    o := ord(ch);
    line := '';
    cha := ' ';
    while o <> 13 do
    begin
        gotoxy(x,y);
        getc(ch);
        if eoln then ch := chr(13);
        o := ord(ch);
        if (o<32) or (o>126) then
        begin
            if not(o in [8,13,21]) then
            begin
                write(chr(7));
            end else begin
                case o of
                    21      :   rgt(line,inn,x);
                    8       :   lft(line,inn,x);
                end;
            end;
```

```
        end else begin
            if length(inn) > 0 then
            begin
                x := x + 1;
                write(ch);
                cha[1] := ch;
                line := concat(line,cha);
                delete(inn,1,1);
            end else begin
                write(chr(7));
            end;
        end
    end;
end;{getline}

procedure sttofp;

var
    valid,digits,
    decimel,negetive        :       boolean;
    nodigits,
    digit,d                 :       integer;
    exp                     :       real;

    procedure makestart(var valid,decimel,negetive:boolean;
                var nodigits,digit,d:integer;
                var xx,exp:real);

    begin
        digits := false;
        valid := true;
        decimel := false;
        negetive := false;
        exp := 0;
        digit := 1;
        nodigits := 0;
        d := 0;
        xx := 0;
    end;

    procedure hndldig(var digits,valid:boolean;var xx:real;
              var nodigits:integer);
```

```
begin
    if (d + nodigits) >= 38 then
        valid := false
    else begin
        xx := xx * 10 + (ord(s[digit]) - ord('0'));
        digits := true;
        if not(decimel) then nodigits := nodigits + 1;
    end;
end;

procedure hndldec(var decimel,valid:boolean;var d:integer);

begin
    if decimel then
    begin
        valid := false;
    end else begin
        decimel := true;
        d := d - 1;
    end;
end;

procedure hndlneg(var negetive,valid:boolean);

begin
    if (not(negetive)) and (digit = 1) then
        negetive := true
    else
        valid := false;
end;

procedure hndlpls(var valid:boolean);

begin
    if digit <> 1 then valid := false;
end;

procedure hndlspc(var decimel:boolean;var d:integer);

begin
    if decimel then d := d - 1;
end;
```

```
procedure hndlexp(var exponent,valid:boolean;var xx,exp:real;
          var digit:integer;var s:string);
var
    diagx   :      string;

begin
    if exponent then
        valid := false
    else begin
        exponent := true;
        delete(s,1,digit);
        sttofp(exp,diagx,s,exponent);
        if trunc(exp) <> exp then
            valid := false;
        if diagx = 'ng' then
            valid := false;
        if not(digits) then
        begin
            xx := 1;
            digits := true;
        end;
        if abs(exp + nodigits - 1) > 37 then
            valid := false;
        digit := length(s) + 1;
        if (decimel) then
            d := d - 1;
    end;
end;
```

```
begin{sttofp}
    makestart(valid,decimel,negetive,nodigits,digit,d,xx,exp);
    while (valid) and (digit <= length(s)) do
    begin
        if s[digit] in ['0'..'9','.','-','+','E','e',' ',',']
        then begin
            if s[digit] in ['0'..'9'] then
                hndldig(digits,valid,xx,nodigits)
            else begin
                case s[digit] of
                    '.' : hndldec(decimel,valid,d);
                    '-' : hndlneg(negetive,valid);
                    '+' : hndlpls(valid);
                    'e' : hndlexp(exponent,
                              valid,xx,exp,
                              digit,s);
                    'E' : hndlexp(exponent,
                              valid,xx,exp,
                              digit,s);
                    ',' : hndlspc(decimel,d);
                    ' ' : hndlspc(decimel,d);
                end;
            end;
        end else begin
            valid := false;
        end;
        digit := digit + 1;
        if decimel then d := d + 1;
    end;
    if d > 37 then valid := false;
    if not(digits) then valid := false;
    if valid then
    begin
        diag := 'ok';
        xx := xx / pwroften(d);
        if negetive then xx := -xx;
        if exponent then
        begin
            if exp >= 0 then
                xx := xx * pwroften(trunc(exp))
            else
                xx := xx / pwroften(-trunc(exp));
        end;
    end else
        diag := 'ng';
end;{sttofp}
```

```
   function less;

   begin
       less := false;
       if op1.page < op2.page
       then
           less := true
       else
           if (op1.page = op2.page) and (op1.point < op2.point)
           then
               less := true;
   end;

   function diff;

   begin
       diff := (larger.page - smaller.page) * 256
            + (larger.point - smaller.point);
   end;

   procedure ink;
   external;

   procedure dek;
   external;

begin{initialization}
end;{typestuff}
```

A.6.9.3 PLOTSTUFF

```
unit plotstuff; intrinsic code 26;

interface
uses typestuff;

   procedure minimax(curve:apfiletype;start,expanse:integer;
             var min,max:byte);
   procedure plotc(curve:apfiletype;strt,expanse,scale,min:integer);
   procedure plotcurve(curve:apfiletype;start,finish:loc;scale:byte);

implementation
   procedure minimax;
   external;

   procedure plotc;
   external;

   procedure plotcurve;

   var
       min,max          :       byte;
       expanse          :       integer;
       strt             :       integer;

   begin{plotcurve}
       expanse := diff(finish,start) + 1;
       strt := 256 * start.page + start.point;
       minimax(curve,strt,expanse,min,max);
       plotc(curve,strt,expanse,scale,min);
   end;{plotcurve}

begin{initialization}
end.{plotstuff}
```

A.6.9.4 ASTIRSTUFF, the Machine Language Subroutines

```
;-----------
; MACRO POP
;-----------
        .MACRO POP
        PLA
        STA %1
        PLA
        STA %1+1
        .ENDM

;-------------
; MACRO DPP
;------------
        .MACRO DPP
        LDA %1+1
        PHA
        LDA %1
        PHA
        .ENDM

        .PROC DEK,1     ; ONE WORD OF PARAMETERS
;-----------------------------------
; PROCEDURE DEK(VAR POINT:LOC);
; DECREMENTS POINT.
;-----------------------------------
RETURN  .EQU    0       ;TEMP VAR FOR RETURN ADDR

        POP RETURN
        PLA
        STA PAGE+1
        STA POINT+1
        STA OUTPG+1
        PLA
        STA PAGE+2
        STA POINT+2
        STA OUTPG+2
        CLC
        LDA POINT+1
        ADC #01
        STA POINT+1
```

```
        STA OUTPT+1
        LDA POINT+2
        ADC #00
        STA POINT+2
        STA OUTPT+2
        SEC
POINT   LDA 01111
        SBC #01
OUTPT   STA 01111
PAGE    LDA 01111
        SBC #00
OUTPG   STA 01111
        DPP RETURN
        RTS

        .PROC INK,1     ; ONE WORD OF PARAMETERS
;-------------------------------------
; PROCEDURE INK(VAR POINT:LOC);
; INCREMENTS POINT.
;-------------------------------------
RETURN  .EQU    0       ;TEMP VAR FOR RETURN ADDR

        POP RETURN
        PLA
        STA PAGE+1
        STA POINT+1
        STA OUTPG+1
        PLA
        STA PAGE+2
        STA POINT+2
        STA OUTPG+2
        CLC
        LDA POINT+1
        ADC #01
        STA POINT+1
        STA OUTPT+1
        LDA POINT+2
        ADC #00
        STA POINT+2
        STA OUTPT+2
        CLC
POINT   LDA 01111
        ADC #01
```

```
OUTPT   STA 01111
PAGE    LDA 01111
        ADC #00
OUTPG   STA 01111
        DPP RETURN
        RTS

        .PROC GETC,1     ; ONE WORD OF PARAMETERS
;-------------------------------------
; PROCEDURE GETC(VAR CH:CHAR);
; (WAITS UNTIL KEYPRESS THEN RETURNS
;  THE CHARACTER ON KEYBOARD)
;-------------------------------------
RETURN  .EQU    0        ;TEMP VAR FOR RETURN ADDR

        POP RETURN
        PLA
        STA COUT+1
        STA COUTP1+1
        PLA
        STA COUT+2
        STA COUTP1+2
        CLC
        LDA COUTP1+1
        ADC #01
        STA COUTP1+1
        LDA COUTP1+2
        ADC #00
        STA COUTP1+2
START   LDA 0C000
        CMP #80
        BCC START
        SBC #80
        CMP #61
        BCC COUT
        SEC
        SBC #20
COUT    STA 01111
        LDA #00
COUTP1  STA 01111
        LDA 0C010
        DPP RETURN
        RTS

        .PROC MINIMAX,5
```

```
;-----------------------------------------------------------------
; PROCEDURE MINIMAX(A:CURVETYPE;START,EXPANSE:INTEGER;
;                   VAR MIN,MAX:INTEGER);
; OBTAINS THE MINIMUM AND MAXIMUM OF A CURVE IN THE RANGE EXPANSE
; STARTING WITH START.
;-----------------------------------------------------------------
RETURN  .EQU    0
MINN    .EQU    2
MAXX    .EQU    4
START   .EQU    6
EXPANSE .EQU    8

        POP RETURN
        PLA
        STA MAX+1
        STA MAXP1+1
        PLA
        STA MAX+2
        STA MAXP1+2
        PLA
        STA MIN+1
        STA MINP1+1
        PLA
        STA MIN+2
        STA MINP1+2
        POP EXPANSE
        POP START
        PLA
        STA ARRAY+1
        PLA
        STA ARRAY+2
        CLC
        LDA MINP1+1
        ADC #01
        STA MINP1+1
        LDA MINP1+2
        ADC #00
        STA MINP1+2
        CLC
        LDA MAXP1+1
        ADC #01
        STA MAXP1+1
        LDA MAXP1+2
```

```
        ADC #00
        STA MAXP1+2

        LDA #0FF
        STA MINN
        LDA #00
        STA MAXX
MINP1   STA 01111
MAXP1   STA 01111
        CLC
        LDA START+1
        ADC ARRAY+2
        STA ARRAY+2
        LDX START
;-----------------------
; WHILE EXPANSE <> 0 DO
;-----------------------
BEGIN   LDA EXPANSE+1
        CMP #00
        BNE ARRAY
        LDA EXPANSE
        CMP #00
        BEQ END
ARRAY   LDA 01111,X
        CMP MINN
        BCS MAXIMUM
        STA MINN
MAXIMUM CMP MAXX
        BCC NEXT1
        STA MAXX
NEXT1   CPX #0FF
        BCC NEXT2
        INC ARRAY+2
NEXT2   INX
        SEC
        LDA EXPANSE
        SBC #01
        STA EXPANSE
        LDA EXPANSE+1
        SBC #00
        STA EXPANSE+1
        JMP BEGIN
END     LDA MINN
```

```
MIN      STA 01111
         LDA MAXX
MAX      STA 01111
         DPP RETURN
         RTS

         .PROC PLOTC,5     ; FIVE WORDS OF PARAMETERS
;-------------------------------------------------
; PROCEDURE PLOTC(A:CURVE;START,EXPANSE,SCALE,MIN:INTEGER);
; PLOTS EXPANSE POINTS OF CURVE A ON HIRES
; SCREEN STARTING WITH POINT START
;-------------------------------------------------
RETURN  .EQU   00        ;TEMP VAR FOR RETURN ADDR
QUOTNT  .EQU   02
DIVISOR .EQU   04
DIVDND  .EQU   06
XX      .EQU   08
YY      .EQU   09
YYOLD   .EQU   0A
YYINT   .EQU   0B
YYTMP   .EQU   0C
COL     .EQU   0D
BITT    .EQU   0E
DISP    .EQU   0F
BAND    .EQU   10
BANDA   .EQU   12
ROW     .EQU   14
ROWA    .EQU   16
BOX     .EQU   18
BOXA    .EQU   1A
ADDR    .EQU   1C
CATCHAL .EQU   1E
EXPANSE .EQU   20
NOPTS   .EQU   22
PTS     .EQU   24
START   .EQU   26
SCALE   .EQU   28
MIN     .EQU   2A
COUNTER .EQU   2C
MM      .EQU   2E
MSG     .EQU   30
```

```
        POP RETURN
        POP MIN
        POP SCALE
        POP EXPANSE
        POP START
        PLA
        STA ARRAY+1
        PLA
        STA ARRAY+2

        JSR SETNOPT
        JSR INIT
FIRE1   JSR ARRAY         ;GET FIRST POINT
        JSR INY16
        JSR DECCTR
                          ;INITIALIZE ADDRESS
        JSR GETROW
        JSR GETADDR
        JSR PLOTYY
        LDA YY
        STA YYOLD
        LDA #01
        CMP NOPTS
        BCC FIRE2
        JSR ARRAY
        JSR INY16
        JSR DECCTR
        JMP MAIN5
FIRE2   JMP MAIN2
                          ;TOP LOOP OF MAIN PROGRAM STARTS HERE
MAIN2   LDA #00
        STA PTS
MAIN3   LDA PTS
        CMP NOPTS
        BCS MAIN5
        JSR ADRYY
        JSR PLOTYY
        LDA YY
        STA YYOLD
        LDA COUNTER+1
        CMP #00
        BNE MAIN4
        LDA COUNTER
```

```
         CMP #00
         BNE MAIN4
         JMP ENDMAIN
MAIN4    JSR ARRAY
         JSR INY16
         JSR DECCTR
         INC PTS
         JMP MAIN3
MAIN5    LDA YYOLD
         STA YYINT
         JSR CALCMM
         LDA MSG
         CMP #01
         BEQ MAIN575
         JSR INCBITT
         JSR PLOTYY
MAIN575 LDA #01
         STA PTS+1
MAIN6    LDA PTS+1
         CMP NOPTS+1
         BCS MAIN9
         LDA YYINT
         ADC MM
         STA YYINT
         LDA MM+1
         CMP #0
         BEQ MAIN8
         DEC MM+1
         LDA YYOLD
         CMP YY
         BCC MAIN7
         DEC YYINT
         JMP MAIN8
MAIN7    INC YYINT
MAIN8    LDA YYOLD
         CMP YY
         BCC MAIN85
         BNE MAIN825
         JSR PLOTINT
MAIN825 JSR INCBITT
         JSR PLOTINT
         LDA #01
         STA MSG
         JMP MAIN875
```

```
MAIN85  JSR PLOTINT
        JSR INCBITT
        LDA #00
        STA MSG
MAIN875 INC PTS+1
        JMP MAIN6
MAIN9   LDA MSG
        CMP #01
        BNE MAIN100
        JSR INCBITT
MAIN100 JMP MAIN2

ENDMAIN JSR ADRYY
        JSR PLOTYY
        LDA MSG
        CMP #01
        BEQ END2
        JSR INCBITT
        JSR PLOTYY
END2    LDA RETURN+1
        PHA
        LDA RETURN
        PHA
        RTS

CALCMM  SEC
        LDA YY
        SBC YYOLD
        BPL PLUS
        STA MM
        LDA #0
        SEC
        SBC MM
PLUS    STA DIVDND
        LDA NOPTS+1
        STA DIVISOR
        JSR DIVIDE2
        LDA QUOTNT
        STA MM
        LDA DIVDND
        STA MM+1
        LDA YY
        CMP YYOLD
```

```
        BPL MMEND
        SEC
        LDA #0
        SBC MM
        STA MM
MMEND   RTS

SETNOPT LDA EXPANSE+1
        CMP #01
        BCC REVERSE
        STA NOPTS
        LDA #01
        STA NOPTS+1
        JMP ENDPTS
REVERSE LDA EXPANSE
        STA DIVISOR
        LDA #0
        STA DIVISOR+1
        STA DIVDND
        LDA #01
        STA DIVDND+1
        JSR DIVIDE2
        LDA QUOTNT
        STA NOPTS+1
        LDA #01
        STA NOPTS
ENDPTS  RTS
                                ;REST OF THE VARIABLES ARE INITIALIZED HERE
INIT    JSR FRAME
        LDA #02
        STA BITT
        LDA #01
        STA MSG
        LDA #0
        STA XX
        STA PTS+1
        STA COL
        CLC
        LDA ARRAY+2
        ADC START+1
        STA ARRAY+2
        LDY START
        LDA EXPANSE
```

```
        STA COUNTER
        LDA EXPANSE+1
        STA COUNTER+1
        RTS

FRAME   LDX #00
        LDA #0FF
TOP     STA 02000,X
BOT     STA 02C50,X
        INX
        CPX #025
        BCC TOP
        LDA #00
        STA COL
        STA BITT
LFT     LDA #01
        STA YY
        STA YYOLD
        JSR GETROW
        JSR GETADDR
        LDA #082
        STA YY
        JSR ADRYY
        JSR PLOTYY
        LDA YY
        STA YYOLD
        INC YYOLD
        JSR INCADR
        LDA #025
        STA COL
        LDA #00
        STA BITT
        STA YY
        JSR ADRYY
        JSR PLOTYY
ENDFRM  RTS

DECCTR  SEC
        LDA COUNTER
        SBC #01
        STA COUNTER
        LDA COUNTER+1
        SBC #00
```

```
        STA COUNTER+1
        RTS

INY16   CPY #0FF
        BCC INY16B
        INC ARRAY+2
INY16B  INY
        RTS

PLOTINT LDA YY
        STA YYTMP
        LDA YYINT
        STA YY
        JSR ADRYY
        JSR SUMADDR
        JSR GETDISP
        JSR SCREEN
        LDA YYTMP
        STA YY
        RTS

PLOTYY  JSR SUMADDR
        JSR GETDISP
        JSR SCREEN
        RTS

                         ;NOW, FIND NEW ADDRESS
ADRYY   JSR PLOTYY
        LDA YY
        CMP YYOLD
        BEQ ENDYY
        BCS INCYY
DECYY   DEC YYOLD
        JSR DECADR
        JMP ADRYY
INCYY   INC YYOLD
        JSR INCADR
        JMP ADRYY
ENDYY   RTS
```

```
INCBITT INC XX
        INC BITT
        LDA BITT
        CMP #07
        BCC ENDBITT
        LDA #0
        STA BITT
        INC COL
        LDA COL
        CMP #28
        BCC ENDBITT
        LDA #0
        STA COL
ENDBITT RTS

INCADR  CLC
        LDA BOXA
        ADC #04
        STA BOXA
        CMP #01C
        BEQ ENDINC
        BCC ENDINC
        LDA #0
        STA BOXA
        CLC
        LDA ROWA
        ADC #080
        STA ROWA
        LDA #0
        ADC ROWA+1
        STA ROWA+1
        CMP #04
        BCC ENDINC
        LDA #0
        STA ROWA
        STA ROWA+1
        CLC
        LDA BANDA
        ADC #028
        STA BANDA
        CMP #050
        BEQ ENDINC
        BCC ENDINC
        LDA #0
        STA BANDA
ENDINC  RTS
```

```
DECADR  SEC
        LDA BOXA
        SBC #04
        STA BOXA
        BPL ENDDEC
        LDA #01C
        STA BOXA
        SEC
        LDA ROWA
        SBC #080
        STA ROWA
        LDA ROWA+1
        SBC #0
        STA ROWA+1
        BPL ENDDEC
        LDA #080
        STA ROWA
        LDA #03
        STA ROWA+1
        SEC
        LDA BANDA
        SBC #028
        STA BANDA
        BPL ENDDEC
        LDA #050
        STA BANDA
ENDDEC  RTS

ARRAY   LDA 01111,Y       ; GET A VALUE FROM THE ARRAY, ZERO AND SCALE IT
        SEC
        SBC MIN
        LDX SCALE
        CLC
        ROR A
SCALE1  CPX #0
        BEQ SKIP0
        CMP #0
        BMI SKIP0
        ASL A
        DEX
        JMP SCALE1
SKIP0   CMP #080
        BCC SKIP1
        LDA #07F
```

```
SKIP1   STA YY
        SEC
        LDA #07F
        SBC YY
        CLC
        ADC #02
        STA YY
        RTS


GETOLD  LDA YYOLD
        STA DIVDND
        LDA #0
                                ; YY MUST LIE IN [0..255].
        STA DIVDND+1
        LDA #040
        STA DIVISOR
        LDA #0
        STA DIVISOR+1
        JSR DIVIDE
        LDA QUOTNT
        STA BAND
        LDA #08
        STA DIVISOR
        LDA #0
        STA DIVISOR+1
        JSR DIVIDE
        LDA DIVDND
        STA BOX
        LDA QUOTNT
        STA ROW
        RTS

GETROW  LDA YY
        STA DIVDND
        LDA #0
                                ; YY MUST LIE IN [0..255].
        STA DIVDND+1
        LDA #040
        STA DIVISOR
        LDA #0
        STA DIVISOR+1
        JSR DIVIDE
```

```
        LDA QUOTNT
        STA BAND
        LDA #08
        STA DIVISOR
        LDA #0
        STA DIVISOR+1
        JSR DIVIDE
        LDA DIVDND
        STA BOX
        LDA QUOTNT
        STA ROW
        RTS

                        ;NOW CALCULATE THE ADDRESS
GETADDR JSR GTBANDA
        JSR GETROWA
        JSR GETBOXA
        JSR SUMADDR
        RTS

SUMADDR CLC
        LDA #0          ;SET BASE LOW BYTE
        ADC COL
        ADC BANDA
        ADC ROWA
        STA ADDR
        LDA #020        ;SET BASE HIGH BYTE
        ADC BOXA
        ADC ROWA+1
        STA ADDR+1
        RTS

GTBANDA LDX BAND
        LDA #0
        STA BANDA
ADDR1   CPX #00
        BEQ NEXT1
        CLC
        LDA #028
        ADC BANDA
        STA BANDA
        DEX
        JMP ADD        STA ROWA
        STA ROWA+1
```

```
ADDR3   CPX #00
        BEQ NEXT3
        CLC
        LDA #080
        ADC ROWA
        STA ROWA
        LDA #00
        ADC ROWA+1
        STA ROWA+1
        DEX
        JMP ADDR3
NEXT3   RTS

GETBOXA LDX BOX
        LDA #0
        STA BOXA
ADDR4   CPX #00
        BEQ NEXT4
        CLC
        LDA #04
        ADC BOXA
        STA BOXA
        DEX
        JMP ADDR4
NEXT4   RTS

                                ;NOW FIND NEW DISPLAY BYTE
GETDISP LDX BITT
        LDA #1
SETBIT  CPX #0
        BEQ NEXT5
        CLC
        ROL A
        DEX
        JMP SETBIT
NEXT5   STA DISP
        RTS
                                ;NOW CHANGE SCREEN
SCREEN  LDA ADDR
        STA STORE+1
        STA INSERT+1
        LDA ADDR+1
        STA STORE+2
        STA INSERT+2
        LDA DISP
```

```
INSERT  ORA 01111
STORE   STA 01111
        RTS

DIVIDE  LDA #0
        STA QUOTNT
DIV     JSR CMP16
        BCC FINDIV
        INC QUOTNT
        SEC
        LDA DIVDND
        SBC DIVISOR
        STA DIVDND
        LDA DIVDND+1
        SBC DIVISOR+1
        STA DIVDND+1
        JMP DIV
FINDIV  RTS

DIVIDE2 LDA #0
        TAX
        STA CATCHAL
REPDIV  LDA DIVISOR
        CMP #02
        BCC FINDIV2
        INX
        CLC
        ROR DIVDND+1
        ROR DIVDND
        ROR CATCHAL
        CLC
        ROR DIVISOR
        JMP REPDIV
FINDIV2 LDA DIVDND
        STA QUOTNT
        LDA DIVDND+1
        STA QUOTNT+1
        LDA #0
        STA DIVDND
        STA DIVDND+1
FINDIV3 CPX #0
        BEQ FINDIV4
        CLC
        ROL CATCHAL
        ROL DIVDND
        DEX
        JMP FINDIV3
```

```
FINDIV4 RTS

CMP16   LDA DIVDND+1
        CMP DIVISOR+1
        BEQ REST
        JMP OUTPUT
REST    LDA DIVDND
        CMP DIVISOR
OUTPUT  RTS

GETCOL  LDA XX
        STA DIVDND
        LDA XX+1
        STA DIVDND+1
        LDA #7
        STA DIVISOR
        LDA #0
        STA DIVISOR+1
        JSR DIVIDE
        LDA DIVDND
        STA BITT
        LDA QUOTNT
        STA COL
        RTS

        .END
```

APPENDIX B

STATS

B.1 INTRODUCTION

STATS is a family of statistical subroutines written by Ward C. Stevens in fortran 77. The original thrust of stats was to provide a simple method of calculating calibration curves which could NOT be assumed to be linear. The objective, therefore was to perform a univariant polynomial curve fit and test the goodness of each fit through analysis of variance. Subsequently, simple F and Student's t tests have been added.

A user must write a short fortran program, compile it with the unix fortran 77 compiler, f77 so that it will be linked to the stats library during compilation and run the object program. The user must be familiar with analysis of variance for the output of the polynomial analysis of variance routines to be meaningful. The user must understand the F and Student's t tests and their limitations to make meaningful use of the F and t test subroutines.

The analysis of variance subroutines contained in stats and the F and t tests are based on Statistical Methods in Research and Production by Owen L. Davies and Peter L. Goldsmith (eds.). [63].

The matrix manipulation subroutines come from Computer Solution of Linear Algebraic Systems by George E. Forsythe and Cleve B. Moler, from Computer Methods for Mathematical Computations by George E. Forsythe, Michael A. Malcolm and Cleve B. Moler, and from the fmm library collected by Cleve B. Moler and modified by him for use on ogcvax, a DEC VAX-11/780.

B.2 EXAMPLES

Two program examples follow. The first program calculates F and t ratios among six groups of data. The second program calculates first through fifth order calibration curves for calibration data which must be entered through the standard input (for example by redirecting a text file).

B.2.1 Example of F and Student's t tests

```
      PROGRAM COMPARE
C
      INTEGER I,NHPD,NHPT,NHDT,NLPD,NLPT,NLDT
      DOUBLE PRECISION HPD(5),HPT(5),HDT(5)
      DOUBLE PRECISION LPD(5),LPT(5),LDT(6)
      DATA (HPD(I),I=1,5) /55.5,53.0,49.5,52.0,54.0/
      DATA (HPT(I),I=1,5) /55.1,53.6,49.3,53.1,55.1/
      DATA (HDT(I),I=1,5) /57.7,59.3,66.1,60.9,71.6/
      DATA (LPD(I),I=1,5) /12.5,12.0,12.5,12.0,12.5/
      DATA (LPT(I),I=1,5) /12.7,12.2,13.0,12.4,12.7/
      DATA (LDT(I),I=1,6) /16.1,17.6,17.8,16.5,15.4,16.1/
      DATA NHPD,NHPT,NHDT,NLPD,NLPT,NLDT /5,5,5,5,5,6/
C
      WRITE(6,100)
      WRITE(6,200) "PENDULUM DIAL NRG"
      WRITE(6,300) "PENDULUM TUP NRG"
      CALL TWOSID(HPD,HPT,NHPD,NHPT)
C
      WRITE(6,100)
      WRITE(6,200) "PENDULUM TUP NRG"
      WRITE(6,300) "DROP TOWER TUP NRG"
      CALL TWOSID(HPT,HDT,NHPT,NHDT)
C
      WRITE(6,100)
      WRITE(6,200) "PENDULUM DIAL NRG"
      WRITE(6,300) "DROP TOWER TUP NRG"
      CALL TWOSID(HPD,HDT,NHPD,NHDT)
C
      WRITE(6,100)
      WRITE(6,200) "PENDULUM DIAL NRG"
      WRITE(6,300) "PENDULUM TUP NRG"
      CALL TWOSID(LPD,LPT,NLPD,NLPT)
C
```

```
C
      WRITE(6,100)
      WRITE(6,200) "PENDULUM TUP NRG"
      WRITE(6,300) "DROP TOWER TUP NRG"
      CALL TWOSID(LPT,LDT,NLPT,NLDT)
C
      WRITE(6,100)
      WRITE(6,200) "PENDULUM DIAL NRG"
      WRITE(6,300) "DROP TOWER TUP NRG"
      CALL TWOSID(LPD,LDT,NLPD,NLDT)
C
      STOP
C
  100 FORMAT(" ")
  200 FORMAT("MU1:   ",16A)
  300 FORMAT("MU0:   ",16A)
      END
```

B.2.2 Example of Calibration Curve Calculation

```
      PROGRAM CALIBRATE
C
      INTEGER N,I,IPVT(9),MAXORD,ORDER,REPS,DF(6),NREPS
      DOUBLE PRECISION X1REPS(9),Y1REPS(9,3),POLY1(27,9)
      DOUBLE PRECISION X(27),Y(27),MUP(9),MUY,MUU(9),MUZ
      DOUBLE PRECISION INFO(9,9),COPY(9,9),CORR(9,9),SYY,SYP(9),SZU(9)
      DOUBLE PRECISION WORK(9),COND,BEE0,BEE(9),B10,B1(9),BS0,BS(9)
      DOUBLE PRECISION VAR(6),MS(6),INFOLU(9,9),INFINV(9,9),BB(9),XX(9)
      DOUBLE PRECISION BT0,BT(9),F
C
C INITIALIZE:
C
      VAR(1) = -1.0D0
      N = 9
      REPS = 3
      NREPS = N * REPS
      MAXORD = 9
      ORDER = 1
C
```

```
C LOAD THE DATA AND DISPLAY IT:
C
      CALL GETXY(N,REPS,XREPS,YREPS)
      CALL PUTXY(N,REPS,XREPS,YREPS)
C
C
      DO 20 ORDER = 1,5
          CALL PANOVA(MAXORD,ORDER,N,REPS,X1REPS,Y1REPS,X,Y,POLY1,MUP,
     &      MUY,MUU,MUZ,INFO,COPY,CORR,SYY,SYP,SZU,IPVT,WORK,COND,BEE0,
     &      BEE,B10,B1,BS0,BS,VAR,DF,MS,NREPS,INFOLU,INFINV,BB,XX,BT0,
     &      BT,F)
          CALL DISPLA(MAXORD,ORDER,B10,B1,BS0,BS,VAR,DF,MS,BT0,BT,F)
   20 CONTINUE
      STOP
  200 FORMAT()
      END
```

B.3 DESCRIPTIONS OF THE SUBROUTINES IN STATS

Each description below consists of three parts:

Part 1 is an example call.
Part 2 is a brief description of what the subroutine does.
Part 3 is a description of inputs and outputs.

If a more detailed explanation of the statistical procedures is required, the reader is referred to Davis and Goldsmith, Chapters 4, 7, & 8. If a more detailed discussion of the matrix routines is needed, the reader is referred to Forsythe and Moler, especially Chapters 16-18 (pp. 58-79).

B.3.1 MATRIX MANIPULATION SUBROUTINES

B.3.1.1 SUBROUTINES DECOMP AND SOLVE

Decomp and solve are NOT described here since they are used exactly as found in the fmm library and are thoroughly described elsewhere. [64];[65].

B.3.1.2 SUBROUTINE INVERT

1. CALL INVERT(MAXORD,N,A,LU,AINV,WORK,BB,XX,IPVT)

2. INVERT is described here since this implementation is a translation by the current author and it contains an important exception to the algol 60 routine on p. 78 of Forsythe and Moler [66] which was the source for the translation. Unlike the original algol 60 routine, this implementation DOES NOT USE ITERATIVE IMPROVEMENT. Note that the matrix invert is useful in and of itself. Consequently, the invert is calculated, rather than using another approach.

   INVERT finds the matrix invert, AINV of a square matrix, A.

3. Inputs:

A(N,N):	the original matrix
N:	the dimension of the square matrix
MAXORD:	the dimension of the square array in which the the matrix is stored

   Outputs:

AINV(N,N):	the matrix invert of A

   Side effects:

LU(N,N):	Upper diagonal decomposition product of A(N,N)
WORK(N) & IPVT(N):	Data flows used for communication between decomp and solve
BB(N) & XX(N):	work space for invert

B.3.2 GENERAL PURPOSE STATISTICAL SUBROUTINES

B.3.2.1 DOUBLE PRECISION FUNCTION MEAN

1. X = MEAN(A, N)

2. MEAN calculates the mean of a column array

3. Inputs:

A:	the column array
N:	the dimension of the column array

Output:

MEAN:	the mean of the column array

B.3.2.2 SUBROUTINE MU

1. CALL MU(MAXORD,X,N,M,MUX)

2. MU calculates a row array which contains the means of the columns of a two dimensional array.

3. Inputs:

X:	the two dimensional array whose means are to be calculated
N:	the first dimension of X
MAXORD:	the second dimension of X
M:	the number of columns which will be averaged

Output:

MUX:	the row array containing the means

B.3.2.3 SUBROUTINE TWOSID

1. CALL TWOSID(X1,XO,N1,NO)

2. TWOSID calculates the averages and variances of two column arrays, and calculates the F ratio between their variances and the t ratio between their means. The number of degrees of freedom for the F and t ratios are also calculated.

3. Inputs:

N1 & NO:	the number of elements in each array
X1(N1) & XO(NO):	the two column arrays

Outputs:

The only output is a text display sent to the standard output displaying the means and variances of the two column arrays, the F ratio between their variances the t ratio between their means and the degrees of freedom associated with the F and t ratios.

B.3.3 SUBROUTINES DIRECTLY USED IN POLYNOMIAL ANALYSIS OF VARIANCE

B.3.3.1 SUBROUTINE VARIANCE

1. CALL VARIANCE(MAXORD,X,N,M,Y,MUX,MUY,INFO,SYX,SYY)

2. VARIANCE calculates covariances.

3. Inputs:

X:	an N by MAXORD array of N observations each of M independent variables. MAXORD adds flexibility.
MAXORD:	a second dimension added to X to allow for higher powers of X also the dimension of the array used to store info, MUX and SYX
Y:	an array of N observations of a dependent variable
N & M:	as described in MU

Outputs:

MUX:	an array of the M averages of the columns of X
MUY:	the average of the Y observations
INFO:	an M by M array containing the covariances, i.e., SXX, of the X columns, the information matrix
SYX:	an array containing the M covariances between each of the N columns of X and Y
SYY:	the variance of Y

B.3.3.2 SUBROUTINE GETB

1. CALL GETB(MAXORD,INFO,COPY,IPVT,WORK,M,SYX,MUX,MUY,COND,BO,B)

2. Subroutine GETB calculates the regression coefficients from the information matrix, the covariances of the dependent variable with the dependent variables, and the averages of the dependent and independent variables.

3. Inputs:

M:	conceptually, the dimension of the square and column matricies used in GETB actually a limit on the do loops
MAXORD:	conceptually the maximum order of the curve fit which the calling program will attempt actually the dimension of the arrays
INFO:	the covariances of the independent variables
SYX:	the covariance of the dependent variable with the independent variable
MUX:	the averages of the independent variables
MUY:	the average of the dependent variable

Outputs:

COND:	the condition of the information matrix
BO:	the regression constant
B:	the regression coefficients

Side effects:

COPY:	a copy of INFO(M,M) which gets scrambled
IPVT:	a data flow which lets decomp comunicate with solve - generally useless otherwise
WORK:	a data flow used only by decomp

B.3.3.3 SUBROUTINE SCALE

1. CALL SCALE(MAXORD,INFO,SYX,SYY,M,CORR,SZU,MUU,MUZ)

2. SCALE scales the information matrix creating the correlation matrix and similarly scales the SYX vector. The transformations used are equivalent to:

$$u_{ji} = x_{ji} - x_{jav} / \mathrm{dsqrt}(S_{jj}/n-1) \text{ and}$$
$$z_i = y_i - y_{av} / \mathrm{dsqrt}(S_{yy}/n-1)$$

NOTE THAT THIS IS A LINEAR TRANSFORMATION

The importance of scaling is that it permits multivariate regression analysis when the independent variables are of different magnitude as, for example, when polynomial fits are being attempted.

3. Inputs:

MAXORD & M:	as defined in GETB
INFO:	the information matrix
SYX:	the covariance vector for the dependent variable with the independent variables
SYY:	the variance of the dependent variable

Outputs:

CORR:	the correlation matrix, i.e., the scaled information matrix
SZU:	the scaled covariance vector

Side effects:

MUU:	the averages of the transformed independent variables - always identically zero but GETB needs this as a double precision vector
MUZ:	the average of the transformed dependent variable - always identically zero, but again GETB needs a double precision value

B.3.3.4 SUBROUTINE DESCALE

1. CALL DESCALE(MAXORD,BEEO,BEE,M,SYY,INFO,MUY,MUX,BO,B)

2. DESCALE produces the uscaled regression constant and coefficients from the scaled regression constant and coefficients.

3. Inputs:

MAXORD & M:	as defined in GETB
BEEO:	the scaled regression constant
BEE:	the scaled regression coefficients
SYY:	the variance of Y
INFO:	(or SXX) the information matrix
MUY:	the average of the dependent variable
MUX:	the averages of the independent variables

Outputs:

BO:	the unscaled regression constant
B:	the unscaled regression coefficients

B.3.3.5 SUBROUTINE MAKREP

1. CALL MAKREP(REPS,N,NREPS,XREPS,YREPS,X,Y)

2. MAKREP transforms a one dimensional array of N independent variable values and a two dimensional array of REPS replicate measurements of N corresponding values of a dependent variable (each column contains REPS replications of the test and thus corresponds to a single value of the independent variable) into a one dimensional array containing each of the independent variable values reps times and a one dimensional array containing all the values of the dependent variable in such an order that every dependent variable value with index I corresponds to the depdendent variable value with index I.

3. Inputs:

REPS:	the number of repetitions of each determination.
N:	the number of determinations.
NREPS:	N * REPS. It is passed instead of calculated to simplify array dimensioning.
XREPS(N):	the N values of the independent variable.
YREPS(N,REPS):	the REPS replications of the N dependent variable determinations.

Outputs:

X(NREPS):	REPS copies of each of the N values of the independent variable.
Y(NREPS):	all the dependent variable data in a one dimensional array.

B.3.3.6 DOUBLE PRECISION FUNCTION PURERR

1. X = PURERR(REPS,N,YREPS)

2. PURERR calculates the pure error in a set of replicated dependent variable data, i.e., it calculates an estimate of the variance due to the test method itself and not due to goodness of fit.

   PURERR calculates the variances of the replicated dependent variable data at each value of the independent variable, sums them and returns the result.

   It is not possible to calculate lack of fit if there is no replication. The host program must recognise that -1.0 indicates failure to do so.

3. Inputs:

N:	the number of values of the independent variable.
REPS:	the number of replications at each value of the independent variable.
YREPS(N,REPS):	the dependent variable data.

   Output:

   The only output is the double precision number returned by PURERR. If there was no replication, it is impossible to calculate pure error, but PURERR returns the value -1 in that case rather than crashing the program or complaining.

B.3.3.7 SUBROUTINE POLYNO

1. CALL POLYNO(MAXORD,X,N,ORDER,Y,POLY,MUP,MUY,MUU,MUZ,INFO,COPY,
            CORR,SYY,SYP,SZU,IPVT,WORK,COND,BEEO,BEE,BO,B)

2. POLYNO performs a polynomial curve fit. POLYNO does NOT perform multivariate polynomial regression.

3. Inputs:

Y:	the observations of the dependent variable
X:	the observations of ONE independent variable
N:	the number of observations also as described under GETB
ORDER:	the order of the polynomial fit
MAXORD:	the maximum order expected by the calling program also as described under GETB

Outputs:

POLY:	the 1st through the M^{th} powers of the independent variable in an N by MAXORD array. MAXORD allows flexibility.
INFO:	the variances of the powers of the independent variable - i.e., the information matrix.
CORR:	the scaled information mayrix i.e., the correlation matrix.
BO:	the regression constant.
B:	the regression coefficients.
BEEO:	the scaled regression constant
BEE:	the scaled regression coefficients
MUP:	the averages of the powers of the independent variables.
MUY:	the average of the dependent variable.
MUU:	the averages of the scaled powers of the independent variable.
MUZ:	the average of the scaled dependent variable.
SYY:	the variance of the dependent variable
SYP:	the covariance between the dependent variable and the powers of the independent variable.
SZU:	the covariances between the scaled dependent variable and the scaled powers of the independent variable.
COND:	the condition of matrix CORR(ORDER,ORDER)

Side effects:

COPY:	a copy of INFO(ORDER,ORDER) scrambled by decomp.
IPVT & WORK:	two arrays needed by decomp and solve for work space. Usually, they are otherwise useless.

B.3.3.8 SUBROUTINE PANOVA

1. CALL PANOVA(MAXORD,ORDER,N,REPS,XREPS,YREPS,X,Y,POLY,MUP,
MUY,MUU,MUZ,INFO,COPY,CORR,SYY,SYP,SZU,IPVT,
WORK,COND,BEEO,BEE,BO,B,BSO,BS,VAR,DF,MS,NREPS,
INFOLU,INFINV,BB,XX,BTO,BT,F)

2. PANOVA is the heart and purpose of the stats package. It performs polynomial analysis of variance for one independent and one dependent variable. IT DOES NOT PERFORM MULTIVARIATE ANALYSIS OF VARIANCE.

   If the calling program has used PANOVA before, VAR(1) contains the last value of the variance explained by regression. It is assumed in that case that the last case was for degree 1 less than this call. Hence, the improvement in variance explained will have degree of freedom of 1. If the calling program has stored -1 in VAR(1), the calculations based on improvement in variance explained will be supressed.

   It is not possible to calculate lack of fit if there is no replication. Nonetheless, the rest of PANOVA may be possible.

3. Inputs:

MAXORD:	the largest order the calling program will call
ORDER:	the current order
N:	the number of determinations
REPS:	the number of replications of each determination
NREPS:	the product of N and REPS because of array dimensioning problems it must be pased instead of calculated in PANOVA
Y:	the observations of the dependent variable

X:	the observations of ONE independent variable
XREPS(N):	the values of the independent variable
YREPS(N,REPS):	the REPS replications of the N determinations of the dependent variable

Outputs:

BO,B(MAXORD):	the regression constant and the regression coefficients
BEEO:	the scaled regression constant
BEE:	the scaled regression coefficients
BSO,BS(MAXORD):	the standard errors of the regression constant and the regression coefficients
BTO,BT(MAXORD):	the t significance ratios of the regression constant and the regression coefficients
VAR(1):	variance explained by the regression.
VAR(2):	the pure error in determination of the independent variable. It is set to -1 if it was impossible to calculate it and is then not displayed.
VAR(3):	variance caused by lack of fit. It is set to -1 if it is impossible to calculate it and is then not displayed.
VAR(4):	variance about regression.
VAR(5):	total variance.
VAR(6):	improvement in variance explained due to this order.
DF(1..6):	degrees of freedom in variances above.
MS(1..6):	mean squares of variances above.
F:	the F ratio for the improvement due to the current order
POLY:	the 1st through the mth powers of the independent variable in an N by MAXORD array. MAXORD allows flexibility.
INFO:	the variances of the powers of the independent variable - i.e., the information matrix.

CORR:	the scaled information matrix, i.e., the correlation matrix.
MUP:	the averages of the powers of the independent variables.
MUY:	the average of the dependent variable.
MUU:	the averages of the scaled powers of the independent variable.
MUZ:	the average of the scaled dependent variable.
SYY:	the variance of the dependent variable
SYP:	the covariance between the dependent variable and the powers of the independent variable.
SZU:	the covariances between the scaled dependent variable and the scaled powers of the independent variable.
COND:	the condition of matrix CORR

Side effects:

COPY:	a copy of INFO scrambled by decomp.
IPVT and WORK:	two arrays needed by DECOMP and SOLVE for work space. Usually, they are otherwise useless.
INFOLU:	the upper diagonal decomposition product of INFO
INFINV:	the inverse of INFO
BB and XX:	workspace for INVERT

B.3.4 SUBROUTINES USED TO DISPLAY RESULTS OR RETRIEVE DATA

B.3.4.1 SUBROUTINE DISPLA

1. CALL DISPLA(MAXORD,ORDER,BO,B,BSO,BS,VAR,DF,MS,BTO,BT,F)

2. DISPLA produces a display of the results of polynomial analysis of variance.

3. Inputs:

MAXORD:	the maximum order fit the calling program attempted to find the dimension of the arrays actually stored
ORDER:	the order polynomial which was found.
BO:	the regression constants.
B(MAXORD):	the regression coefficients.
BSO:	the standard error of the regression constant.
BS(MAXORD):	the standard errors of the regression coefficients.
BTO:	the two sided t ratio for the significance of BO.
BT(MAXORD):	the two sided t ratios for the significance of B.
VAR(1):	variance explained by the regression.
VAR(2):	the pure error in determination of the independent variable. It is set to -1 if it was impossible to calculate it and is then not displayed.
VAR(3):	variance caused by lack of fit. It is set to -1 if it is impossible to calculate it and is then not displayed.
VAR(4):	variance about regression.
VAR(5):	total variance.
VAR(6):	improvement in variance explained due to this order.
DF(1..6):	degrees of freedom in variances above.
MS(1..6):	mean squares of variances above.
F:	F ratio: MS(6) / MS(4). Obviously the degrees of freedom are: DF(6) and DF(4).

B.3.4.2 SUBROUTINE ANVTBL

1. CALL ANBTBL(VAR,DF,MS,F,ORDER)

2. ANVTBL forms an analysis of variance table

3. Inputs:
   The meaning of the inputs has been explained under SUBROUTINE DISPLA above.

   Outputs:
   The only output is a text display sent to the standard output.

B.3.4.3 SUBROUTINE EQN

1. CALL EQN(MAXORD,ORDER,B0,B)

2. EQN displays the regression equation.

3. Inputs:
   The meaning of the inputs has been explained under SUBROUTINE DISPLA above.

   Outputs:
   The only output is a text display sent to the standard output.

B.3.4.4 SUBROUTINE BSETBL

1. CALL BSETBL(MAXORD,ORDER,B0,B,BS0,BS,BT0,BT,DF)

2. BSETBL displays the regression constant, the regression coefficients, their standard errors and t ratios.

3. Inputs:
   The meaning of the inputs has been explained under SUBROUTINE DISPLA above.

   Outputs:
   The only output is a text display sent to the standard output.

B.3.4.5 SUBROUTINE PUTXY

1. CALL PUTXY(N,REPS,XREPS,YREPS)

2. PUTXY writes the input data to the standard output in x, y pairs

3. Inputs:

REPS:	the number of repetitions of each determination.
N:	the number of determinations.
NREPS:	N * REPS. It is passed instead of calculated to simplify array dimensioning.
XREPS(N):	the N values of the independent variable.
YREPS(N,REPS):	the REPS replications of the N dependent variable determinations.

Outputs:

The only output is a text display which is sent to the standard output.

B.3.4.6 SUBROUTINE GETXY

1. CALL GETXY(N,REPS,XREPS,YREPS)

2. GETXY reads the input data from the standard input. Of course the data could be redirected from a file.

3. Inputs:

REPS:	the number of repetitions of each determination.
N:	the number of determinations.
NREPS:	N * REPS. It is passed instead of calculated to simplify array dimensioning.

Outputs:

XREPS(N):	the N values of the independent variable.
YREPS(N,REPS):	the REPS replications of the N dependent variable determinations.

B.3.4.7 SUBROUTINE CALC

1. CALL CALC(MAXORD,ORDER,N,REPS,XREPS,YREPS,B0,B)

2. CALC displays the values of the independent variable, the corresponding values of the averages of the dependent variable and the corresponding predictions for the values of the dependent variable based on the polynomial curve fit.

3. Inputs:

   The meaning of the inputs is described under GETXY and DISPLA above.

   Outputs:

   The only output is a text display sent to the standard output.

B.4 SOURCE CODE

What follows is the source code for the stats library. Since the special features of f77 are NOT used, stats should be portable to any system with a fortran 77 compiler:

```
=========================================================================
B.4.1  Matrix Manipulation Subroutine INVERT
=========================================================================

      SUBROUTINE INVERT(MAXORD,N,A,LU,AINV,WORK,BB,XX,IPVT)
C INVERT FINDS THE MATRIX INVERT, AINV OF A SQUARE MATRIX, A.
C THIS IMPLEMENTATION IS A TRANSLATION OF THE ALGOL 60 ROUTINE ON P
C OF THE BOOK BY FORSYTHE AND MOLER.  AN IMPORTANT EXCEPTION IS:
C IT DOES NOT USE ITERATIVE IMPROVEMENT.
C
C   INPUTS:
C            A(N,N):  THE ORIGINAL MATRIX
C   OUTPUTS:
C            AINV(N,N):  THE MATRIX INVERT OF A
C   SIDE EFFECTS:
C                  LU(N,N):  UPPER DIAGONAL DECOMPOSITION PRODUCT OF
C                            A(N,N)
C        WORK(N) & IPVT(N):  DATA FLOWS USED FOR COMUNICATION BETWEEN
C                            DECOMP AND SOLVE
C            BB(N) & XX(N):  WORK SPACE FOR INVERT
      INTEGER N,I,J,MAXORD,IPVT(MAXORD)
      DOUBLE PRECISION A(MAXORD,MAXORD),LU(MAXORD,MAXORD)
      DOUBLE PRECISION AINV(MAXORD,MAXORD),WORK(MAXORD),BB(MAXORD)
      DOUBLE PRECISION XX(MAXORD),COND
C
      DO 20 J=1,N
        DO 10 I=1,N
            LU(I,J) = A(I,J)
   10   CONTINUE
   20 CONTINUE
C SINCE DECOMP WILL LEAVE THE INPUT MATRIX IN UPPER DIAGONAL FORM
C AND A(N,N) WILL BE NEEDED LATER, IT IS NECESSARY TO MAKE A COPY OF
C A(N,N).  IT IS CONVENIENT TO DO SO ON LU(N,N).
```

```
C
      CALL DECOMP(MAXORD,N,LU,COND,IPVT,WORK)
      DO 50 J=1,N
        DO 30 I=1,N
          IF (I.EQ.J) THEN
            BB(I) = 1.0D0
          ELSE
            IF (I.NE.J) BB(I) = 0.0D0
          ENDIF
          XX(I) = BB(I)
   30   CONTINUE
        CALL SOLVE(MAXORD,N,LU,XX,IPVT)
C THE LATEST VERSION OF SOLVE USES DOUBLE PRECISION ARITHMETIC.  MAYBE
C ITERATIVE IMPROVEMENT IS UNNECESSARY.
        DO 40 I=1,N
          AINV(I,J) = XX(I)
   40   CONTINUE
   50 CONTINUE
      RETURN
      END
C
C
C
```

===

B.4.2 General Purpose Statistical Subroutines

B.4.2.1 MEAN

===

```
      DOUBLE PRECISION FUNCTION MEAN(A, N)
C
C THIS FUNCTION CALCULATES THE MEAN OF A COLUMN ARRAY
C
      INTEGER N, I
      DOUBLE PRECISION A(N), SUM
C
      SUM = 0.0D0
      DO 10 I = 1, N
        SUM = SUM + A(I)
   10 CONTINUE
      MEAN = SUM / N
      RETURN
      END
C
C
C
```

```
========================================================================
B.4.2.2  MU
========================================================================

C
      SUBROUTINE MU(MAXORD,X,N,M,MUX)
C
C THIS SUBROUTINE CALCULATES A ROW ARRAY WHICH CONTAINS THE MEANS
C OF THE COLUMNS OF A TWO DIMENSIONAL ARRAY.
C
      INTEGER M,MAXORD,N,I,J
      DOUBLE PRECISION X(N,MAXORD), MUX(MAXORD), SUM
C
      DO 20 I = 1, M
        SUM = 0.0D0
        DO 10 J = 1, N
          SUM = SUM + X(J,I)
   10   CONTINUE
        MUX(I) = SUM / N
   20 CONTINUE
      RETURN
      END
C
C
C
```

```
========================================================================
B.4.2.3  TWOSID
========================================================================

      SUBROUTINE TWOSID(X1,X0,N1,N0)
C
      EXTERNAL MEAN
      INTEGER N1,N0
      INTEGER DFF(2),DFT
      INTEGER I
      DOUBLE PRECISION X1(N1),X0(N0),MEAN
      DOUBLE PRECISION F,T
      DOUBLE PRECISION MU1,MU0,SIGMA1,SIGMA0,S1SQ,S0SQ
      DOUBLE PRECISION RPHI,TERM1,TERM0
C
```

```
C COMPUTING AND REPORTING THE MEANS:
C
      MU1 = MEAN(X1,N1)
      MU0 = MEAN(X0,N0)
      WRITE(6,100) MU1,MU0
C
C COMPUTING THE SUMS OF SQUARES AND VARIANCES
C   SIGMAS ARE SUMS OF SQUARES.
C   SNSQS ARE VARIANCES (IE S SQUARED VALUES OR THE SQUARES OF STANDARD
C                           DEVIATION)
C
C
      SIGMA1 = 0.0D0
      SIGMA0 = 0.0D0
      DO 10 I = 1,N1
        SIGMA1 = SIGMA1 + (X1(I)-MU1)*(X1(I)-MU1)
   10 CONTINUE
      DO 20 I = 1,N0
        SIGMA0 = SIGMA0 + (X0(I)-MU0)*(X0(I)-MU0)
   20 CONTINUE
      S1SQ = SIGMA1 / (N1-1)
      S0SQ = SIGMA0 / (N0-1)
C
C CALCULATING THE F RATIO AND THE DEGREES OF FREEDOM OF THE VARIANCES:
      DFF(1) = N1-1
      DFF(2) = N0-1
      F = S1SQ / S0SQ
      IF (F.LT.1.0D0) F = 1/F
C
C NOW, F CONTAINS THE F RATIO AND DFF CONTAINS THE DEGREES OF FREEDOM.
C
      IF (S1SQ.GT.S0SQ) WRITE(6,200) F,DFF(1),DFF(2)
      IF (S1SQ.LE.S0SQ) WRITE(6,200) F,DFF(2),DFF(1)
C
C COMPUTING THE T RATIO:
C
      T = (MU1 - MU0) / DSQRT((S1SQ/N1) + (S0SQ/N0))
      IF (T.LT.0.0D0) T = -T
C
```

```
C DO NOT ASSUME THAT THE TWO VARIANCES ARE EQUAL SO CALCULATE THE
C INVERSE OF THE NUMBER OF DEGREES OF FREEDOM, RPHI
C OF THE T RATIO AS FOLLOWS:
C
      TERM1 = (S1SQ/N1) / ((S1SQ / N1) + (S0SQ/N0))
      TERM1 = TERM1 * TERM1
      TERM1 = TERM1 / (N1 - 1)
      TERM0 = (S0SQ/N1) / ((S1SQ / N1) + (S0SQ/N0))
      TERM0 = TERM0 * TERM0
      TERM0 = TERM0 / (N0 - 1)
      RPHI = TERM1 + TERM0
C
C THE INVERSE OF RPHI WILL BE A FLOATING POINT NUMBER, SO ROUND TO GET
C THE NUMBER OF DEGREES OF FREEDOM OF THE T RATIO:
C
C
      DFT = (1.0D0 / RPHI) + 1.0D0 / 2.0D0
C
C
C THE DOUBLE PRECISION NUMBERS ON THE RIGHT HAND SIDE FORCE THE RIGHT
C HAND SIDE TO BE DOUBLE PRECISION.  ADDING ONE HALF CHANGES TRUNCATION
C TO ROUNDING UP.
C
C NOW, T CONTAINS THE TWO SIDED T RATIO AND DFT CONTAINS THE
C CORRESPONDING NUMBER OF DEGREES OF FREEDOM.  ALL THAT REMAINS IS TO
C REPORT THEM:
C
      WRITE(6,300) T,DFT
      RETURN
  100 FORMAT("MU1: ",F12.8," MU0: ",F12.8)
  200 FORMAT("F RATIO:  ",F12.8," DEGREES OF FREEDOM:  ",I4," & ",I4)
  300 FORMAT("T RATIO:  ",F12.8," DEGREES OF FREEDOM:  ",I4)
      END
C
C
C
```

===

B.4.3 Subroutines Directly Used in Polynomial Analysis of Variance

B.4.3.1 VARIANCE

===

```
      SUBROUTINE VARIANCE(MAXORD,X,N,M,Y,MUX,MUY,INFO,SYX,SYY)
C
C THIS SUBROUTINE CALCULATES COVARIANCES.
C INPUTS:
C         X : AN N BY MAXORD ARRAY OF N OBSERVATIONS EACH OF M
C             INDEPENDENT VARIABLES.  MAXORD ADDS FLEXIBILITY.
C         Y : AN ARRAY OF N OBSERVATIONS OF A DEPENDENT VARIABLE
C    N AND M: AS DESCRIBED ABOVE
C OUTPUTS:
C           MUX:  AN ARRAY OF THE M AVERAGES OF THE COLUMNS OF X
C           MUY:  THE AVERAGE OF THE Y OBSERVATIONS
C          INFO:  AN M BY M ARRAY CONTAINING THE COVARIANCES,
C                 IE SXX, OF THE X COLUMNS, THE INFORMATION MATRIX
C           SYX:  AN ARRAY CONTAINING THE M COVARIANCES BETWEEN EACH OF
C                 THE X COLUMNS AND Y
C           SYY:  THE VARIANCE OF Y
C
      INTEGER N,M,MAXORD,I,J,K
      DOUBLE PRECISION X(N,MAXORD), INFO(MAXORD,MAXORD)
      DOUBLE PRECISION Y(N), MUX(MAXORD), SYX(MAXORD)
      DOUBLE PRECISION MEAN, MUY, SYY
C
C
C
C FIRST, CALCULATE MEANS:
C
      MUY = MEAN(Y,N)
      CALL MU(MAXORD,X,N,M,MUX)
C
C NOW CALCULATE INFO:
C
      DO 30 I = 1, M
        DO 20 J = I, M
          INFO(I,J) = 0.0D0
          DO 10 K = 1, N
            INFO(I,J) = INFO(I,J) + (X(K,I)-MUX(I)) * (X(K,J)-MUX(J))
   10     CONTINUE
          INFO(J,I) = INFO(I,J)
   20   CONTINUE
   30 CONTINUE
C
```

```
C NOW CALCULATE SYX:
C
      DO 50 I = 1, M
        SYX(I) = 0.0D0
        DO 40 J = 1, N
          SYX(I) = SYX(I) + ((X(J,I) - MUX(I)) * (Y(J) - MUY))
   40   CONTINUE
   50 CONTINUE
C
C NOW CALCULATE SYY:
C
      SYY = 0.0D0
      DO 60 I = 1, N
        SYY = SYY + (Y(I)-MUY) * (Y(I)-MUY)
   60 CONTINUE
C
      RETURN
      END
C
C
C
```

==
B.4.3.2 GETB
==

```
      SUBROUTINE GETB(MAXORD,INFO,COPY,IPVT,WORK,M,SYX,MUX,MUY,COND,BO,
     &                B)
C
C SUBROUTINE GETB CALCULATES THE REGRESSION COEFFICIENTS FROM THE
C INFORMATION MATRIX, THE COVARIANCES OF THE DEPENDENT VARIABLE WITH
C THE DEPENDENT VARIABLES, AND THE AVERAGES OF THE DEPENDENT AND
C INDEPENDENT VARIABLES.
C INPUT:
C         INFO(M,M):  THE COVARIANCES OF THE INDEPENDENT VARIABLES
C         COPY(M,M):  A COPY OF INFO(M,M) WHICH GETS SCRAMBLED
C         IPVT(M):    A DATA FLOW WHICH LETS DECOMP COMUNICATE
C                     WITH SOLVE - GENERALLY USELESS OTHERWISE
C         SYX(M):     THE COVARIANCE OF THE DEPENDENT VARIABLE WITH
C                     THE INDEPENDENT VARIABLE
C         MUX(M):     THE AVERAGES OF THE INDEPENDENT VARIABLES
C         MUY:        THE AVERAGE OF THE DEPENDENT VRIABLE
C         WORK:       A DATA FLOW USED ONLY BY DECOMP
C OUTPUT:
C          COND:  THE CONDITION OF THE INFORMATION MATRIX
C          BO:    THE REGRESSION CONSTANT
C          B(M):  THE REGRESSION COEFFICIENTS
C
      INTEGER M,MAXORD,I,J,IPVT(MAXORD)
      DOUBLE PRECISION INFO(MAXORD,MAXORD),SYX(MAXORD),MUX(MAXORD)
      DOUBLE PRECISION COPY(MAXORD,MAXORD),WORK(MAXORD),MUY
      DOUBLE PRECISION COND,BO,B(MAXORD)
      DOUBLE PRECISION CONDP1
C
      DO 20 I = 1,M
        B(I) = SYX(I)
        DO 10 J = 1,M
           COPY(J,I) = INFO(J,I)
   10   CONTINUE
   20 CONTINUE
C
```

```
      CALL DECOMP(MAXORD,M,COPY,COND,IPVT,WORK)
      CONDP1 = COND + 1.0D0
      IF (COND .EQ. CONDP1) GOTO 50
      CALL SOLVE(MAXORD,M,COPY,B,IPVT)
      B0 = MUY
      DO 50 I = 1,M
        B0 = B0 - B(I) * MUX(I)
   50 CONTINUE
      RETURN
      END
C
C
C
```

===
B.4.3.3 SCALE
===

```
      SUBROUTINE SCALE(MAXORD,INFO,SYX,SYY,M,CORR,SZU,MUU,MUZ)
C
C SCALE SCALES THE INFORMATION MATRIX CREATING THE CORRELATION
C MATRIX AND SIMILARLY SCALES THE SYX VECTOR.  THE TRANSFORMATIONS
C USED ARE EQUIVALENT TO:
C  UIJ = XJI - XJAV / DSQRT(SJJ/N-1) AND
C  ZI  = YI - YAV / DSQRT(SYY/N-1)
C NOTE THAT THIS IS A LINEAR TRANSFORMATION
C INPUT:
C          INFO(M,M):  THE INFORMATION MATRIX
C          SYX(M):     THE COVARIANCE VECTOR FOR THE DEPENDENT
C                      VARIABLE WITH THE INDEPENDENT VARIABLES
C          SYY:        THE VARIANCE OF THE DEPENDENT VARIABLE
C OUTPUT:
C           CORR(M,M):  THE CORRELATION MATRIX, IE THE SCALED
C                       INFORMATION MATRIX
C           SZU(M):     THE SCALED COVARIANCE VECTOR
C           MUU(M):     THE AVERAGES OF THE TRANSFORMED INDEPENDENT
C                       VARIABLES - ALWAYS IDENTICALLY ZERO BUT
C                       GETB NEEDS THIS AS A DOUBLE PRECISION
C                       VECTOR
C           MUZ:        THE AVERAGE OF THE TRANSFORMED DEPENDENT
C                       VARIABLE - ALWAYS IDENTICALLY ZERO BUT
C                       AGAIN GETB NEEDS A DOUBLE PRECISION VALUE
C THE IMPORTANCE OF SCALING IS THAT IT PERMITS MULTIVARIATE
C REGRESSION ANALYSIS WHEN THE INDEPENDENT VARIABLES ARE OF
C DIFFERENT MAGNITUDE AS, FOR EXAMPLE, WHEN POLYNOMIAL FITS
C ARE BEING ATTEMPTED.
C
```

```
      INTEGER M,MAXORD,I,J
      DOUBLE PRECISION INFO(MAXORD,MAXORD),SYX(MAXORD),SYY
      DOUBLE PRECISION CORR(MAXORD,MAXORD),SZU(MAXORD),MUU(MAXORD)
      DOUBLE PRECISION MUZ
C
      DO 20 I = 1,M
        SZU(I) = SYX(I) / DSQRT(SYY * INFO(I,I))
        MUU(I) = 0.0D0
        DO 10 J = 1,M
          CORR(J,I) = INFO(J,I) / DSQRT(INFO(J,J) * INFO(I,I))
   10   CONTINUE
   20 CONTINUE
      MUZ = 0.0D0
      RETURN
      END
C
C
C
```

==
B.4.3.4 DESCALE
==

```
      SUBROUTINE DESCALE(MAXORD,BEEO,BEE,M,SYY,INFO,MUY,MUX,BO,B)
C
C DESCALE PRODUCES THE USCALED REGRESSION CONSTANT AND COEFFICIENTS
C FROM THE SCALED REGRESSION CONSTANT AND COEFFICIENTS.
C   INPUTS:
C             BEEO:    THE SCALED REGRESSION CONSTANT
C             BEE(M):  THE SCALED REGRESSION COEFFICIENTS
C             SYY:     THE VARIANCE OF Y
C             INFO:    (OR SXX) THE INFORMATION MATRIX
C             MUY:     THE AVERAGE OF THE DEPENDENT VARIABLE
C             MUX(M):  THE AVERAGES OF THE INDEPENDENT VARIABLES
C   OUTPUTS:
C              BO:    THE UNSCALED REGRESSION CONSTANT
C              B(M):  THE UNSCALED REGRESSION COEFFICIENTS
      INTEGER M,MAXORD,I
      DOUBLE PRECISION BEEO,BEE(MAXORD),SYY,INFO(MAXORD,MAXORD),MUY
      DOUBLE PRECISION MUX(MAXORD),BO,B(MAXORD)
C
      DO 10 I=1,M
        B(I) = BEE(I) * DSQRT(SYY/INFO(I,I))
   10 CONTINUE
      BO = MUY
      DO 20 I=1,M
        BO = BO - B(I) * MUX(I)
   20 CONTINUE
      RETURN
      END
C
C
C
```

B.4.3.5 MAKREP

```
      SUBROUTINE MAKREP(REPS,N,NREPS,XREPS,YREPS,X,Y)
C MAKREP TRANSFORMS A ONE DIMENSIONAL ARRAY OF N INDEPENDENT VARIABLE
C VALUES AND A TWO DIMENSIONAL ARRAY OF REPS REPLICATE MEASUREMENTS  N
C CORRESPONDING VALUES OF A DEPENDENT VARIABLE (EACH COLUMN CONTAINS
C REPS REPLICATIONS OF THE TEST AND THUS CORRESPONDS TO A SINGLE VALUE
C OF THE INDEPENDENT VARIABLE.) INTO A ONE DIMENSIONAL ARRAY CONTAINING
C EACH OF THE INDEPENDENT VARIABLE VALUES REPS TIMES AND A ONE
C DIMENSIONAL ARRAY CONTAINING ALL THE VALUES OF THE DEPENDENT VARIABLE
C IN SUCH AN ORDER THAT EVERY DEPENDENT VARIABLE VALUE WITH INDEX I
C CORRESPONDS TO THE DEPDENDENT VARIABLE VALUE WITH INDEX I.
C   INPUTS:
C            REPS:  THE NUMBER OF REPETITIONS OF EACH DETERMINATION.
C               N:  THE NUMBER OF DETERMINATIONS.
C           NREPS:  N * REPS.  IT IS PASSED INSTEAD OF CALCULATED
C                   TO SIMPLIFY ARRAY DIMENSIONING.
C        XREPS(N):  THE N VALUES OF THE INDEPENDENT VARIABLE.
C   YREPS(N,REPS):  THE REPS REPLICATIONS OF THE N DEPENDENT VARIABLE
C                   DETERMINATIONS.
C   OUTPUTS:
C               X(NREPS):  REPS COPIES OF EACH OF THE N VALUES OF
C                           THE INDEPENDENT VARIABLE.
C               Y(NREPS):  ALL THE DEPENDENT VARIABLE DATA IN A ONE
C                           DIMENSIONAL ARRAY.
      INTEGER N,REPS,NREPS,I,J,INDEX
      DOUBLE PRECISION XREPS(N),YREPS(N,REPS),X(NREPS),Y(NREPS)
C
      INDEX = 1
      DO 20 J=1,N
        DO 10 I=1,REPS
          X(INDEX) = XREPS(J)
          Y(INDEX) = YREPS(J,I)
          INDEX = INDEX + 1
   10   CONTINUE
   20 CONTINUE
      RETURN
      END
C
C
C
```

==
B.4.3.6 PURERR
==

```
      DOUBLE PRECISION FUNCTION PURERR(REPS,N,YREPS)
C PURERR CALCULATES THE PURE ERROR IN A SET OF REPLICATED
C DEPENDENT VARIABLE DATA.
C
C   INPUTS:
C             N:  THE NUMBER OF VALUES OF THE INDEPENDENT VARIABLE.
C          REPS:  THE NUMBER OF REPLICATIONS AT EACH VALUE OF THE
C                 INDEPENDENT VARIABLE.
C YREPS(N,REPS):  THE DEPENDENT VARIABLE DATA.
C
C   OUTPUT:
C            PURERR CALCULATES THE VARIANCES OF THE REPLICATED
C            DEPENDENT VARIABLE DATA AT EACH VALUE OF THE
C            INDEPENDENT VARIABLE, SUMS THEM AND RETURNS THE RESULT.
C
      INTEGER N,REPS,I,J
      DOUBLE PRECISION YREPS(N,REPS),ERR,MUY
C
      IF (REPS.LT.2) THEN
C IT IS NOT POSSIBLE TO CALCULATE LACK OF FIT IF THERE IS NO
C REPLICATION.  THE HOST PROGRAM MUST RECOGNISE THAT -1.0 INDICATES
C FAILURE TO DO SO.
        ERR = -1.0D0
      ELSE
        ERR = 0.0D0
        DO 30 J=1,N
          MUY = 0.0D0
          DO 10 I=1,REPS
            MUY = MUY + YREPS(J,I)
   10     CONTINUE
          MUY = MUY/REPS
          DO 20 I=1,REPS
            ERR = ERR + (YREPS(J,I) - MUY) * (YREPS(J,I) - MUY)
   20     CONTINUE
   30   CONTINUE
      ENDIF
      PURERR = ERR
      RETURN
      END
C
C
C
```

```
=================================================================
B.4.3.7  POLYNO
=================================================================
```

```
      SUBROUTINE POLYNO(MAXORD,X,N,ORDER,Y,POLY,MUP,MUY,MUU,MUZ,INFO,
     &  COPY,CORR,SYY,SYP,SZU,IPVT,WORK,COND,BEEO,BEE,BO,B)
C
C POLYNO PERFORMS A POLYNOMIAL CURVE FIT
C   INPUTS:
C            Y(N):  THE OBSERVATIONS OF THE DEPENDENT VARIABLE
C            X(N):  THE OBSERVATIONS OF ONE INDEPENDENT VARIABLE
C                   POLYNO DOES NOT PERFORM MULTIVARIATE POLYNOMIAL
C                   REGRESSION.
C               N:  THE NUMBER OF OBSERVATIONS
C           ORDER:  THE ORDER OF THE POLYNOMIAL FIT
C          MAXORD:  THE MAXIMUM ORDER EXPECTED BY THE CALLING PROGRAM.
C   OUTPUTS:
C            POLY(N,MAXORD):  THE 1ST THROUGH THE MTH POWERS OF THE
C                             INDEPENDENT VARIABLE IN AN N X MAXORD
C                             ARRAY.  MAXORD ALLOWS FLEXIBILITY.
C         INFO(ORDER,ORDER):  THE VARIANCES OF THE POWERS OF THE
C                             INDEPENDENT VARIABLE - IE THE INFORMATION
C                             MATRIX.
C         COPY(ORDER,ORDER):  A COPY OF INFO(ORDER,ORDER) SCRAMBLED BY
C                             DECOMP.
C         CORR(ORDER,ORDER):  THE SCALED INFORMATION MATRIX IE THE
C                             CORRELATION MATRIX.
C                        BO:  THE REGRESSION CONSTANT.
C                  B(ORDER):  THE REGRESSION COEFFICIENTS.
C                MUP(ORDER):  THE AVERAGES OF THE POWERS OF THE
C                             INDEPENDENT VARIABLES.
C                       MUY:  THE AVERAGE OF THE DEPENDENT VARIABLE.
C                MUU(ORDER):  THE AVERAGES OF THE SCALED POWERS OF THE
C                             INDEPENDENT VARIABLE.
C                       MUZ:  THE AVERAGE OF THE SCALED DEPENDENT
C                             VARIABLE.
C                       SYY:  THE VARIANCE OF THE DEPENDENT VARIABLE
C                SYP(ORDER):  THE COVARIANCE BETWEEN THE DEPENDENT
C                             VARIABLE  AND THE POWERS OF THE
C                             INDEPENDENT VARTIABLE.
C                SZU(ORDER):  THE COVARIANCES BETWEEN THE SCALED
C                             DEPENDENT VARIABLE AND THE SCALED
C                             POWERS OF THE INDEPENDENT VARIABLE.
C                      COND:  THE CONDITION OF CORR(ORDER,ORDER)
C IPVT(ORDER) & WORK(ORDER):  ARRAYS NEEDED BY DECOMP AND SOLVE.
C                             USUALLY, THEY ARE USELESS.
```

```
C
      INTEGER MAXORD,ORDER,N,I,J,IPVT(MAXORD)
      DOUBLE PRECISION X(N),Y(N),POLY(N,MAXORD),MUP(MAXORD),MUY
      DOUBLE PRECISION MUU(MAXORD),MUZ,INFO(MAXORD,MAXORD)
      DOUBLE PRECISION COPY(MAXORD,MAXORD),CORR(MAXORD,MAXORD),SYY
      DOUBLE PRECISION SYP(MAXORD),SZU(MAXORD),WORK(MAXORD),COND,BEEO
      DOUBLE PRECISION BEE(MAXORD),BO,B(MAXORD)
C
C
C
C FIRST, FILL IN POLY:
C
      DO 30 I=1,N
        POLY(I,1) = X(I)
        DO 20 J=2,ORDER
          POLY(I,J) = POLY(I,1) * POLY(I,(J-1))
   20   CONTINUE
   30 CONTINUE
      CALL VARIANCE(MAXORD,POLY,N,ORDER,Y,MUP,MUY,INFO,SYP,SYY)
      CALL SCALE(MAXORD,INFO,SYP,SYY,ORDER,CORR,SZU,MUU,MUZ)
      CALL GETB(MAXORD,CORR,COPY,IPVT,WORK,ORDER,SZU,MUU,MUZ,COND,BEEO,
     &          BEE)
      CALL DESCALE(MAXORD,BEEO,BEE,ORDER,SYY,INFO,MUY,MUP,BO,B)
      RETURN
      END
C
C
C
```

```
==============================================================
B.4.3.8  PANOVA
==============================================================


      SUBROUTINE PANOVA(MAXORD,ORDER,N,REPS,XREPS,YREPS,X,Y,POLY,MUP,
     &MUY,MUU,MUZ,INFO,COPY,CORR,SYY,SYP,SZU,IPVT,WORK,COND,BEEO,BEE,
     &BO,B,BSO,BS,VAR,DF,MS,NREPS,INFOLU,INFINV,BB,XX,BTO,BT,F)
C
C PANOVA PERFORMS POLYNOMIAL ANALYSIS OF VARIANCE FOR ONE INDEPENDENT
C AND ONE DEPENDENT VARIABLE.  IT DOES NOT PERFORM MULTIVARIATE
C ANALYSIS OF VARIANCE.
C
C  INPUTS:
C           MAXORD:  THE LARGEST ORDER THE CALLING PROGRAM WILL CALL
C            ORDER:  THE CURRENT ORDER
C                N:  THE NUMBER OF DETERMINATIONS
C             REPS:  THE NUMBER OF REPLICATIONS OF EACH DETERMINATION
C         XREPS(N):  THE VALUES OF THE INDEPENDENT VARIABLE
C    YREPS(N,REPS):  THE REPS REPLICATIONS OF THE N DETERMINATIONS OF
C                    THE DEPENDENT VARIABLE
C  OUTPUTS:
C     BO,B(MAXORD):  THE REGRESSION COEFFICIENTS
C   BSO,BS(MAXORD):  THE STANDARD ERRORS OF THE REGRESSION COEFFICIENTS
C   BTO,BT(MAXORD):  THE T SIGNIFICANCE RATIOS OF THE REGRESSION
C                    COEFFICIENTS
C                F:  THE F RATIO FOR THE IMPROVEMENT DUE TO THE CURRENT
C                    ORDER
C
C
      INTEGER N,I,J,MAXORD,IPVT(MAXORD),ORDER,REPS,DF(6),NREPS,M
      DOUBLE PRECISION XREPS(N),YREPS(N,REPS)
      DOUBLE PRECISION X(NREPS),Y(NREPS),POLY(NREPS,MAXORD),MUP(MAXORD)
      DOUBLE PRECISION MUY,MUU(MAXORD),MUZ,INFO(MAXORD,MAXORD)
      DOUBLE PRECISION COPY(MAXORD,MAXORD),CORR(MAXORD,MAXORD),SYY
      DOUBLE PRECISION SYP(MAXORD),SZU(MAXORD)
      DOUBLE PRECISION WORK(MAXORD),COND,BEEO,BEE(MAXORD),BO,B(MAXORD)
      DOUBLE PRECISION BSO,BS(MAXORD),VAR(6)
      DOUBLE PRECISION MS(6),PURERR,INFOLU(MAXORD,MAXORD)
      DOUBLE PRECISION INFINV(MAXORD,MAXORD),BB(MAXORD),XX(MAXORD)
      DOUBLE PRECISION BTO,F,OLD,BT(MAXORD)
```

```
C
      IF (VAR(1).LT.0.0D0) THEN
        OLD = -1.0D0
      ELSE
        OLD = VAR(1)
      END IF
C IF THE CALLING PROGRAM HAS USED PANOVA BEFORE, VAR(1) CONTAINS THE
C LAST VALUE OF THE VARIANCE EXPLAINED BY REGRESSION.  IT IS ASSUMED
C IN THAT CASE THAT THE LAST CASE WAS FOR DEGREE 1 LESS THAN THIS CALL.
C HENCE, THE IMPROVEMENT IN VARIANCE EXPLAINED WILL HAVE DEGREE OF
C FREEDOM OF 1.
C IF THE CALLING PROGRAM HAS STORED -1 IN VAR(1), THE CALCULATIONS
C BASED ON IMPROVEMENT IN VARIANCE EXPLAINED WILL BE SUPRESSED.
      CALL MAKREP(REPS,N,NREPS,XREPS,YREPS,X,Y)
      IF (ORDER.GT.(N-1)) THEN
        M = N-1
      ELSE
        M = ORDER
      END IF
C N POINTS ARE NECESSARY TO FIT AN (N-1)TH ORDER POLYNOMIAL.
C USING M IN THE POLYNO CALL IS IMPORTANT BECAUSE IT ALLOWS
C PRESERVATION OF THE VALUE OF ORDER IN THE MAIN PROGRAM.
      CALL POLYNO(MAXORD,X,NREPS,M,Y,POLY,MUP,MUY,MUU,MUZ,INFO,COPY,
     &CORR,SYY,SYP,SZU,IPVT,WORK,COND,BEE0,BEE,B0,B)
      CALL INVERT(MAXORD,M,INFO,INFOLU,INFINV,WORK,BB,XX,IPVT)
C NOW, PANOVA HAS EVERYTHING IT NEEDS FOR THE CALCULATIONS.
      VAR(2) = PURERR(REPS,N,YREPS)
      IF (VAR(2).LT.0.0D0) VAR(3) = -1.0D0
C IT IS NOT POSSIBLE TO CALCULATE LACK OF FIT IF THERE IS NO
C REPLICATION.  PURERR WILL RETURN -1.0D0 IF REPS <= 1. NONE-THE-
C LESS, THE REST OF PANOVA MAY BE POSSIBLE.
C
```

```
C
      VAR(1) = 0.0D0
      DO 30 I = 1,M
        VAR(1) = VAR(1) + B(I) * SYP(I)
   30 CONTINUE
      VAR(4) = SYY - VAR(1)
      VAR(3) = VAR(4) - VAR(2)
      VAR(5) = SYY
      DF(1) = M
      DF(2) = (REPS-1) * N
      DF(4) = REPS * N - M - 1
      DF(3) = DF(4) - DF(2)
      DF(5) = REPS * N - 1
      IF (OLD.LT.0.0D0) THEN
        VAR(6) = -1.0D0
        DF(6) = -1
      ELSE
        VAR(6) = VAR(1) - OLD
        DF(6) = 1
      END IF
      DO 35 I=1,5
        IF (DF(I).EQ.0.0D0) DF(I) = -1.0D0
   35 CONTINUE
C ASSUMING OF COURSE THAT THE LAST CALCULATION WAS FOR ORDER = M - 1.
      DO 40 I=1,6
        IF (DF(I).LE.0.OR.VAR(I).LT.0.0D0) THEN
          MS(I) = -1.0D0
        ELSE
          MS(I) = VAR(I) / DF(I)
        END IF
   40 CONTINUE
      F = MS(6)/MS(4)
      DO 50 I=1,M
        BS(I) = DSQRT(MS(4) * INFINV(I,I))
```

```
C
C MS(4) IS AN ESTIMATE OF THE RESIDUAL VARIANCE SQUARED.  IT IS ALSO
C USED IN THE ESTIMATION OF BSO.
   50 CONTINUE
      BSO = 1 / NREPS
      DO 70 J=1,M
        DO 60 I=1,M
          BSO = BSO + INFINV(I,J) * MUP(I) * MUP(J)
   60   CONTINUE
   70 CONTINUE
      BSO = BSO * MS(4)
      BSO = DSQRT(BSO)
      BTO = DABS(BO/BSO)
      DO 80 I=1,M
        BT(I) = DABS(B(I)/BS(I))
   80 CONTINUE
      RETURN
      END
C
C
C
```

```
=================================================================
B.4.4  Subroutines Used to Display Results or Retrieve Data
B.4.4.1  DISPLA
=================================================================
```

```
      SUBROUTINE DISPLA(MAXORD,ORDER,BO,B,BSO,BS,VAR,DF,MS,BTO,BT,F)
C DISPLA PRODUCES A DISPLAY OF THE RESULTS OF POLYNOMIAL ANALYSIS
C OF VARIANCE.
C   INPUTS:
C           ORDER:  THE ORDER POLYNOMIAL WHICH WAS FOUND.
C                   DEPENDENT VARIABLE.
C              BO:  THE REGRESSION CONSTANTS.
C       B(MAXORD):  THE REGRESSION COEFFICIENTS.
C             BSO:  THE STANDARD ERROR OF THE REGRESSION CONSTANT.
C      BS(MAXORD):  THE STANDARD ERRORS OF THE REGRESSION COEFFICIENTS.
C             BTO:  THE TWO SIDED T RATIO FOR THE SIGNIFICANCE OF BO.
C      BT(MAXORD):  THE TWO SIDED T RATIOS FOR THE SIGNIFICANCE OF B.
C          VAR(1):  VARIANCE EXPLAINED BY THE REGRESSION.
C          VAR(2):  THE PURE ERROR IN DETERMINATION OF THE INDEPENDENT
C                   VARIABLE.  IT IS SET TO -1 IF IT WAS IMPOSSIBLE TO
C                   CALCULATE IT AND IS THEN NOT DISPLAYED.
C          VAR(3):  VARIANCE CAUSED BY LACK OF FIT.  IT IS SET TO -1 IF
C                   IT IS IMPOSSIBLE TO CALCULATE IT AND IS THEN NOT
C                   DISPLAYED.
C          VAR(4):  VARIANCE ABOUT REGRESSION.
C          VAR(5):  TOTAL VARIANCE.
C          VAR(6):  IMPROVEMENT IN VARIANCE EXPLAINED DUE TO THIS
C                   ORDER.
C        DF(1..6):  DEGREES OF FREEDOM IN VARIANCES ABOVE.
C        MS(1..6):  MEAN SQUARES OF VARIANCES ABOVE.
C               F:  F RAITIO: MS(6) / MS(4).  OBVIOUSLY THE DEGREES OF
C                   FREEDOM ARE:  DF(6) AND DF(4)
```

```
C
      INTEGER MAXORD,ORDER,DF(6)
      DOUBLE PRECISION BO,B(MAXORD),BSO,BS(MAXORD),VAR(6),MS(6)
      DOUBLE PRECISION BTO,BT(MAXORD),F
C
      WRITE(6,100)
      CALL EQN(MAXORD,ORDER,BO,B)
      CALL ANVTBL(VAR,DF,MS,F,ORDER)
      CALL BSETBL(MAXORD,ORDER,BO,B,BSO,BS,BTO,BT,DF)
      WRITE(6,100)
C ANVTBL FORMS AN ANALYSIS OF VARIANCE TABLE
      RETURN
  100 FORMAT("")
      END
C
C
C
```

```
=======================================================================
B.4.4.2  ANVTBL
=======================================================================


      SUBROUTINE ANVTBL(VAR,DF,MS,F,ORDER)
C ANVTBL FORMS AN ANALYSIS OF VARIANCE TABLE
      INTEGER I,DF(6),ORDER
      DOUBLE PRECISION VAR(6),MS(6),F
      CHARACTER TITLES(6)*24,FRAMES(5)*67
C
      FRAMES(1) = "
---------------------------------------------------
     &------------- "
      FRAMES(2) = "
|-------------------|---------------|------------|--
     &------------| "
      FRAMES(4) = "  |       SOURCE        |     SUM OF     |  DEGREES
|
     &   MEAN      | "
      FRAMES(5) = "  |                     |    SQUARES     | OF FREEDOM
|
     &  SQUARE     | "
      TITLES(1) = "  | DUE TO REGRESSION | "
      TITLES(2) = "  |   PURE ERROR      | "
      TITLES(3) = "  |   LACK OF FIT     | "
      TITLES(4) = "  | ABOUT REGRESSION  | "
      TITLES(5) = "  | TOTAL             | "
      TITLES(6) = "  | DUE TO THIS ORDER | "
C
```

```
C
      WRITE(6,110) FRAMES(1)
      WRITE(6,110) FRAMES(4)
      WRITE(6,110) FRAMES(5)
      WRITE(6,110) FRAMES(2)
      DO 10 I=1,6
        IF ((VAR(2).LT.0.0D0).AND.(I.EQ.2)) I=I+2
        IF (VAR(I).LT.0.0D0) GOTO 10
        WRITE(6,100) TITLES(I),VAR(I),DF(I),MS(I)
        IF (I.EQ.6.OR.(I.EQ.5.AND.VAR(6).LT.0.0D0)) GOTO 10
        WRITE(6,110) FRAMES(2)
   10 CONTINUE
      WRITE(6,110) FRAMES(1)
      IF (VAR(6).GE.0.0D0) THEN
        WRITE(6,120) ORDER,F
        WRITE(6,110) FRAMES(1)
        WRITE(6,130) DF(6),DF(4)
        WRITE(6,110) FRAMES(1)
      END IF
      RETURN
  100 FORMAT(A24,F12.4," |   ",I8,"  | ",F12.4," | ")
  110 FORMAT(A67)
  120 FORMAT("  | F RATIO FOR IMPROVEMENT DUE TO ORDER ",I2,":
",E11.5,
     &8X,"|")
  130 FORMAT("  | DEGREES OF FREEDOM:   ",I4," & ",I4,29X,"|")
      END
C
C
C
```

```
===========================================================================
B.4.4.3  EQN
===========================================================================

      SUBROUTINE EQN(MAXORD,ORDER,BO,B)
C EQN DISPLAYS THE REGRESSION EQUATION.
      INTEGER MAXORD,ORD,ORDER,I,LAST
      DOUBLE PRECISION BO,B(MAXORD)
C
      ORD = ORDER
C THE LAST LINE ALLOWS PRESERVATION OF THE VALUE OF ORDER.
      IF (ORD.LE.0) GOTO 90
      IF (ORD.GT.4) THEN
        LAST = 4
        ORD = ORD - 4
        WRITE(6,100) (I,I=2,4)
        WRITE(6,110) BO,(B(I),I=1,4)
      ELSE
        WRITE(6,100) (I,I=2,ORD)
        WRITE(6,110) BO,(B(I),I=1,ORD)
        ORD = ORD - 4
      END IF
   10 IF (ORD.LE.0) GOTO 90
      IF (ORD.GT.4) THEN
        WRITE(6,120) (I,I=(LAST+1),(LAST+4))
        WRITE(6,130) (B(I),I=(LAST+1),(LAST+4))
        ORD = ORD - 4
        LAST = LAST + 4
      ELSE
        WRITE(6,120) (I,I=(LAST+1),(LAST+ORD))
        WRITE(6,130) (B(I),I=(LAST+1),(LAST+ORD))
        ORD = -1
      END IF
      GOTO 10
   90 RETURN
  100 FORMAT(2X,"^",29X,4(14X,I1))
  110 FORMAT(SP,2X,"Y = ",E11.5,X,4(E11.5," X  "))
  120 FORMAT(5X,4(14X,I1))
  130 FORMAT(SP,6X,4(E11.5," X  "))
      END
C
C
C
```

```
==========================================================================
B.4.4.4  BSETBL
==========================================================================


      SUBROUTINE BSETBL(MAXORD,ORDER,BO,B,BSO,BS,BTO,BT,DF)
C BSETBL DISPLAYS THE REGRESSION CONSTANT, THE REGRESSION COEFFICIENTS,
C THEIR STANDARD ERRORS AND T RATIOS.
      INTEGER MAXORD,ORD,ORDER,I,LAST,DF(6)
      DOUBLE PRECISION BO,B(MAXORD),BSO,BS(MAXORD),BTO,BT(MAXORD)
      CHARACTER TITLES(7)*24,FRAMES(2)*14
C
      FRAMES(1) = "--------------"
      FRAMES(2) = "-------------|"
      TITLES(1) = "  ----------------------"
      TITLES(2) = "  |     COEFFICIENT     |"
      TITLES(3) = "  |        VALUE        |"
      TITLES(4) = "  |   STANDARD ERROR    |"
      TITLES(5) = "  | TWO SIDED T RATIO   |"
      TITLES(6) = "  | DEGREES OF FREEDOM: "
      TITLES(7) = "  |--------------------|"
C
```

```
      ORD = ORDER
C THE LAST LINE ALLOWS PRESERVATION OF THE VALUE OF ORDER.
      IF (ORD.LE.0) GOTO 90
      IF (ORD.GT.2) THEN
        LAST = 2
        ORD = ORD - 2
        WRITE(6,100)
        WRITE(6,110) TITLES(1),(FRAMES(1),I=1,3)
        WRITE(6,120) TITLES(2),(I,I=0,2)
        WRITE(6,110) TITLES(7),(FRAMES(2),I=1,3)
        WRITE(6,130) TITLES(3),B0,(B(I),I=1,2)
        WRITE(6,110) TITLES(7),(FRAMES(2),I=1,3)
        WRITE(6,130) TITLES(4),BS0,(BS(I),I=1,2)
        WRITE(6,110) TITLES(7),(FRAMES(2),I=1,3)
        WRITE(6,130) TITLES(5),BT0,(BT(I),I=1,2)
        WRITE(6,110) TITLES(1),(FRAMES(1),I=1,3)
        WRITE(6,140) TITLES(6),DF(4)
        WRITE(6,150) TITLES(1),FRAMES(1)
      ELSE
        WRITE(6,100)
        WRITE(6,110) TITLES(1),(FRAMES(1),I=1,(ORD+1))
        WRITE(6,120) TITLES(2),(I,I=0,ORD)
        WRITE(6,110) TITLES(7),(FRAMES(2),I=1,(ORD+1))
        WRITE(6,130) TITLES(3),B0,(B(I),I=1,ORD)
        WRITE(6,110) TITLES(7),(FRAMES(2),I=1,(ORD+1))
        WRITE(6,130) TITLES(4),BS0,(BS(I),I=1,ORD)
        WRITE(6,110) TITLES(7),(FRAMES(2),I=1,(ORD+1))
        WRITE(6,130) TITLES(5),BT0,(BT(I),I=1,ORD)
        WRITE(6,110) TITLES(1),(FRAMES(1),I=1,(ORD+1))
        WRITE(6,140) TITLES(6),DF(4)
        WRITE(6,150) TITLES(1),FRAMES(1)
        ORD = -1
      END IF
```

```
   10 IF (ORD.LE.0) GOTO 90
      IF (ORD.GT.3) THEN
        WRITE(6,100)
        WRITE(6,110) TITLES(1),(FRAMES(1),I=1,3)
        WRITE(6,120) TITLES(2),(I,I=(LAST+1),(LAST+3))
        WRITE(6,110) TITLES(7),(FRAMES(2),I=1,3)
        WRITE(6,130) TITLES(3),(B(I),I=(LAST+1),(LAST+3))
        WRITE(6,110) TITLES(7),(FRAMES(2),I=1,3)
        WRITE(6,130) TITLES(4),(BS(I),I=(LAST+1),(LAST+3))
        WRITE(6,110) TITLES(7),(FRAMES(2),I=1,3)
        WRITE(6,130) TITLES(5),(BT(I),I=(LAST+1),(LAST+3))
        WRITE(6,110) TITLES(1),(FRAMES(1),I=1,3)
        WRITE(6,140) TITLES(6),DF(4)
        WRITE(6,150) TITLES(1),FRAMES(1)
        ORD = ORD - 3
        LAST = LAST + 3
      ELSE
        WRITE(6,100)
        WRITE(6,110) TITLES(1),(FRAMES(1),I=1,ORD)
        WRITE(6,120) TITLES(2),(I,I=(LAST+1),(LAST+ORD))
        WRITE(6,110) TITLES(7),(FRAMES(2),I=1,ORD)
        WRITE(6,130) TITLES(3),(B(I),I=(LAST+1),(LAST+ORD))
        WRITE(6,110) TITLES(7),(FRAMES(2),I=1,ORD)
        WRITE(6,130) TITLES(4),(BS(I),I=(LAST+1),(LAST+ORD))
        WRITE(6,110) TITLES(7),(FRAMES(2),I=1,ORD)
        WRITE(6,130) TITLES(5),(BT(I),I=(LAST+1),(LAST+ORD))
        WRITE(6,110) TITLES(1),(FRAMES(1),I=1,ORD)
        WRITE(6,140) TITLES(6),DF(4)
        WRITE(6,150) TITLES(1),FRAMES(1)
        ORD = -1
      END IF
      GOTO 10
   90 RETURN
C
  100 FORMAT()
  110 FORMAT(A24,3A14)
  120 FORMAT(A24,3(6X,"B",I1,5X,"|"))
  130 FORMAT(A24,3(" ",E11.5," |"))
  140 FORMAT(A24,4X,I5,4X"|")
  150 FORMAT(A24,A14)
      END
C
C
C
```

```
=====================================================================
B.4.4.5  PUTXY
=====================================================================


      SUBROUTINE PUTXY(N,REPS,XREPS,YREPS)
C
C PUTXY WRITES THE OUTPUT DATA TO THE STANDARD OUTPUT OR A FILE
C IF A REDIRECT HAS BEEN USED.
C
      INTEGER I,J,N,REPS
      DOUBLE PRECISION XREPS(N),YREPS(N,REPS)
C
      DO 10 I=1,N
        WRITE(6,100) XREPS(I),(YREPS(I,J),J=1,REPS)
   10 CONTINUE
      RETURN
  100 FORMAT(" ",15(F11.4,", "))
      END
C
C
C
```

```
=====================================================================
B.4.4.6  GETXY
=====================================================================


      SUBROUTINE GETXY(N,REPS,XREPS,YREPS)
C
C GETXY READS THE INPUT DATA FROM THE STANDARD INPUT OR A FILE
C IF A REDIRECT HAS BEEN USED.
C
      INTEGER I,J,N,REPS
      DOUBLE PRECISION XREPS(N),YREPS(N,REPS)
C
      DO 10 I=1,N
        READ(5,100) XREPS(I),(YREPS(I,J),J=1,REPS)
   10 CONTINUE
      RETURN
  100 FORMAT(15F13.4)
      END
C
C
C
```

```
==========================================================================
B.4.4.7  CALC
==========================================================================


      SUBROUTINE CALC(MAXORD,ORDER,N,REPS,XREPS,YREPS,B0,B)
C
C CALC DISPLAYS THE VALUES OF THE INDEPENDENT VARIABLE,
C THE CORRESPONDING VALUES OF THE AVERAGES OF THE DEPENDENT VARIABLE
C AND THE CORRESPONDING PREDICTIONS FOR THE VALUES OF THE DEPENDENT
C VARIABLE BASED ON THE POLYNOMIAL CURVE FIT.
C
C
      INTEGER MAXORD,ORDER,N,REPS,I,J
      DOUBLE PRECISION XREPS(N),YREPS(N,REPS),B0,B(MAXORD),AVG,CALCY
C
      DO 90 I=1,N
        AVG = 0.0D0
        DO 70 J=1,REPS
          AVG = AVG + YREPS(I,J)
   70   CONTINUE
        AVG = AVG / REPS
        CALCY = 0.0D0
        DO 80 J=0,(ORDER-1)
          CALCY = XREPS(I) * (CALCY + B(ORDER-J))
   80   CONTINUE
        CALCY = CALCY + B0
        WRITE(6,100) XREPS(I),CALCY,AVG
   90 CONTINUE
      RETURN
  100 FORMAT("X:   ",F12.4," YHAT:   ",F12.4," YAVG:   ",F12.4)
      END
C
C
C
```

APPENDIX C

THE AMPLIFIER-POWER SUPPLY

C.1 Design of the Amplifier

The amplifier-power supply used for the experiment in this dissertation was designed and built by the OGC electronics shop. Its response is reported and discussed in section 3.1.4 herein. Figure C-1 is a schematic of the amplifier-power supply design. Figure C-2 is a photograph of the amplifier-power supply.

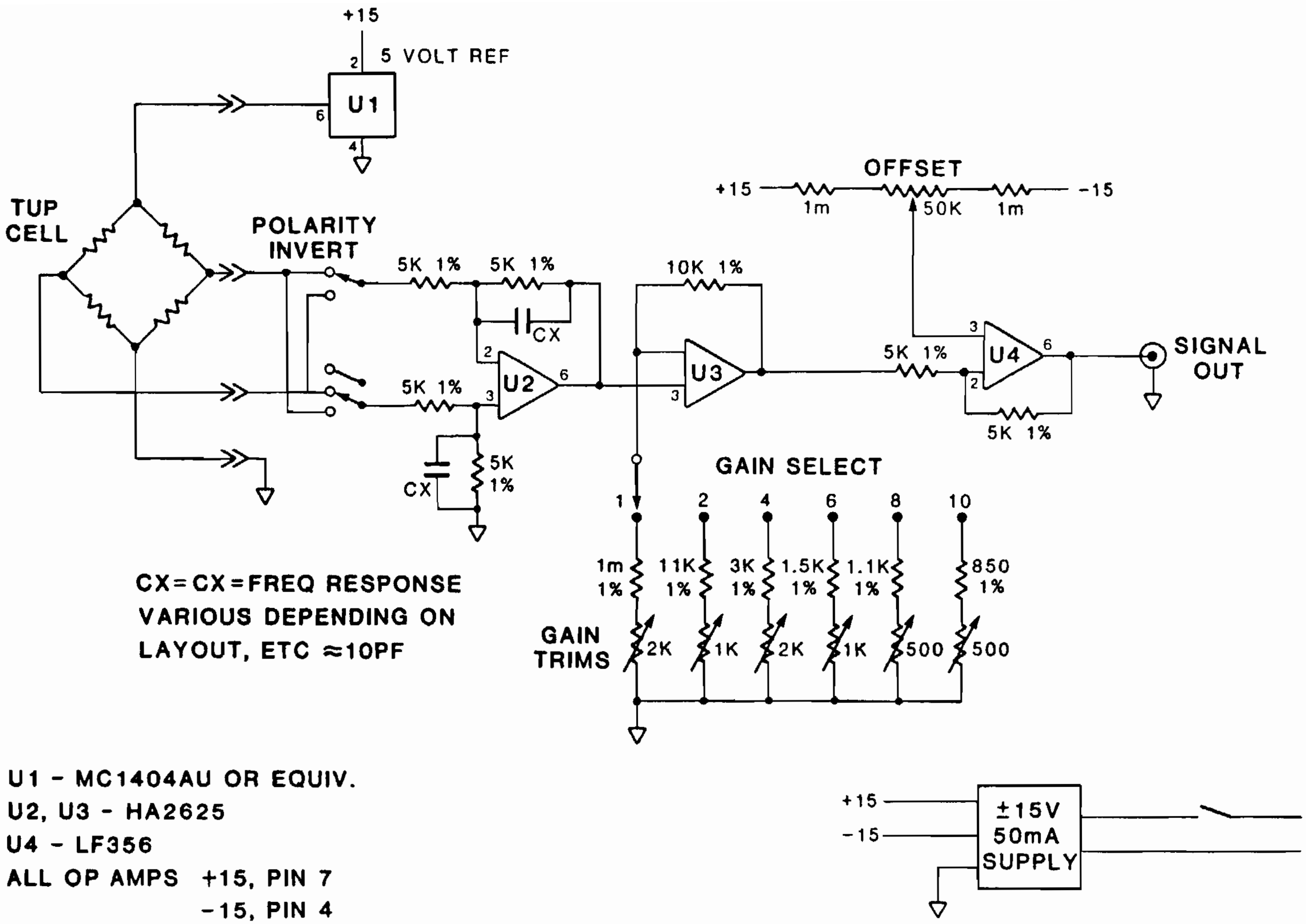


Figure C-1. Schematic of the Amplifier-Power Supply

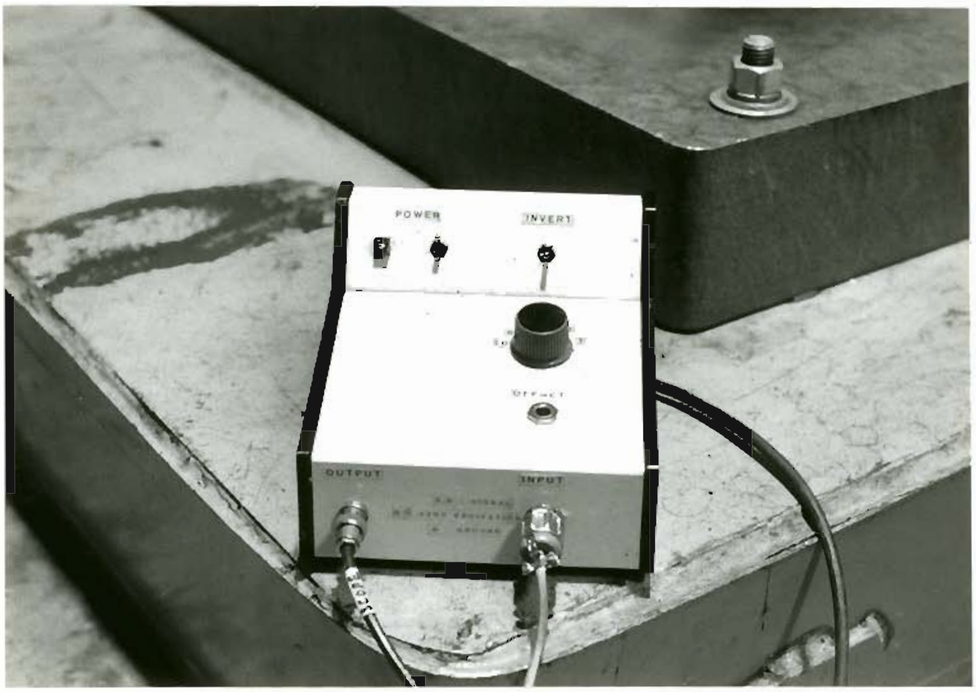


Figure C-2. Photograph of the Amplifier-Power Supply

BIOGRAPHICAL NOTE

The author was born and reared in White Plains, New York, where he attended White Plains High School. He graduated with an academic diploma in 1965. He then entered Case Institute of Technology in Cleveland, Ohio, where he received a B.S. in Metallurgy in January 1969. After graduating from college, the author enlisted in the United States Army. From January 1970 to November 1971 he served in the II Corps region of South Vietnam as a combat construction apprentice, combat demolitions apprentice, combat demolitions specialist, and soils analyst. He received the Bronze Star for distinguished service while serving with delta company of the Fourth Engineer Batallion. In January 1972, the author accepted a position with Titech International Inc. in Pomona, California where he served as a chemist and later as a project metallurgist until April 1975. He left Titech to take a position as a parts engineer with Precision Castparts Corp. in Milwaukie, Oregon. In June 1981, the author accepted a research fellowship in Materials Science at the Oregon Graduate Center. He received his M.S. in Materials Science in October 1987. He completed all course work and examinations for the degree of Ph.D. and also this dissertation, thus satisfying all the requirements for a Ph.D. in Materials Science.