ANALYSIS OF GAMMA RAY SPECTRA FROM NEUTRON ACTIVATION STUDIES

by

H.Levent Akın

B.S. in A.E., Istanbul Technical University, 1982

Submitted to the Institute for Graduate Studies in Science and Engineering in partial fulfillment of the requirements for the degree of

Master of Science

in

Nuclear Engineering

Boğaziçi University

1984

ANALYSIS OF GAMMA RAY SPECTRA FROM NEUTRON ACTIVATION STUDIES

APPROVED BY

Doç.Dr. Vural Altın
(Thesis Supervisor)

Doç.Dr. A.Nezihi Bilge

Doç.Dr.Başar Civelek

Prof.Dr. Turan B. Enginol

DATE OF APPROVAL 13.8.1984

ACKNOWLEDGEMENTS

I would like to express my special thanks to Doç.Dr. Özer Çiftçioğlu whose continual interest helped in preparing the programs. I would also like to thank Doç.Dr.Vural Altın, my thesis advisor, for his invaluable and encouraging guidance and assistance throughout this work. Special thanks are extended to Prof.Dr.Turan B.Enginol and Doç.Dr.A.Nezihi Bilge for their constructive criticsms and suggestions.

Finally my thanks are due to OECD NEA DATA BANK for supplying program SAMP080 and GAMDAT-78 gamma ray data collection.

Levent Akın

ABSTRACT

Neutron activation analysis coupled with Gamma Ray Spectrometry is a powerful and highly precise, non-destructive elemental analysis method.

In this work a program package, BUCAASA, was developed for the analysis of gamma ray spectra.

The first program of the package BUCAASA/FIND, using an improved version of the method of the smoothed first derivative identifies the peaks in the spectrum.

The second program BUCAASA/FIT is then used for fitting a function to the peaks for the accurate determination of peak location and peak area. The function used for fitting was developed by Çiftçioğlu.

The third program BUCAASA/CALIBR performs energy and efficiency calibration.

The constituent nuclides together with their activities are identified by the fourth program BUCAASA/NUCLIDE which is an adapted version of SAMP080 PART III.

All of the programs are interactive and can be used for both routine and academic work. Since they are coded in standard FORTRAN 77 programming language their implementation on other computers is easy. Implementation on microcomputers requires overlaying however.

ÖZET

Gamma spektrometri ile birlikte yapılan nötron aktivasyon analizi güçlü, tahribatsız bir element analizi yöntemidir.

Bu çalışmada gamma spektrumlarının analizi için bir program paketi, BUCAASA geliştirilmiştir.

İlk program BUCAASA/FIND, düzgünleştirilmiş birinci türev yönteminin geliştirilmiş bir şeklini kullanarak spektrumdaki fotopiklerin tanınması işlevini yerine getirir.

İkinci program BUCAASA/FIT, hassas kanal ve fotopik alanı saptaması için fotopiklere eğri uydurmada kullanılır. Gözönüne alınan fonksiyon Çiftçioğlu tarafından geliştirilmiştir.

Üçüncü program BUCAASA/CALIBR, enerji ve verim kalibrasyonunu yapar.

Spektrumu meydana getiren nüklitlerin tanınması SAMPO80 PART III'ün uyarlanmış bir şekli olan dördüncü program BUCAASA/NUCLIDE tarafından yapılır.

Bütün programlar etkileşimlidir ve rutin ve akademik çalışmalar için kullanılabilir. Bütün programlar standard FORTRAN 77 programlama dilinde yazıldığı için başka sistemlere uyarlanmaları kolaydır. Mikrobilgisayarlara uyarlamak için örtme gereklidir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

A	Height of the Gaussian function, peak area, activity mass number
$\underline{\underline{A}}_i$	Matrix
A_G	Area of the Gaussian function
A_T	Area of the tailing function
A_g	Height of the Gaussian function
A_t	Coefficient of the tailing function
A_v	Avogadros number
B	Background, tailing function parameter
$\underline{\underline{B}}$	Interference matrix
C	Tailing function parameter
$C(x)$	Compton continuum polynomial
C_i	Counts in channel i
D	Tailing function parameter
$D(x)$	Distortion function
$\underline{D}$	Experimental intensity vector
E	Energy absorbed
E_B	Binding energy of the electron
E_c	Energy of the Compton edge
E_k	Kinetic energy of the electron
E_o	Energy of the incident photon
E_p	Energy of the light photons
E_{tol}	Energy tolerance in nuclide identification
E_1	Energy transferred to a charged particle
F	Fano factor
$F(x)$, $F(x,\underline{p})$	Photopeak function
FWHM	Full width at half maximum
G	Internal gain
$G(x)$	Faussian function
$\underline{\underline{G}}$	Hessian matrix
J	Transparency parameter
$\underline{\underline{H}}$	Inverse of the Hessian matrix
K	Proportionality constant, geometrical factor
M	Atomic mass
M_A	Mass
N	Number of disintegrations

N_m	Normalization factor in data convolution
Q	Total charge
R	Error correlation
S	Significance value
S^2	Variance of the energy calibration function
S(x)	Step function
S_i	Second difference
T	Time
T(x)	Tailing function
U	Overall amplification
V	Volume
$\underline{\underline{V}}$	Variance covariance matrix
Y	Conversion efficiency
Z	Atomic number
$\underline{a}$	Derivative vector
a_i	Coefficient of Compton polynomial
a_j	convolution integers
e_i	Photopeak energy
f	Isotopic abundance
$f(\underline{p},x)$	Photopeak function
$\underline{g}$	Gradient vector
h	Step function parameter
h_{max}	Pulse height
i,j,k,l	Index
m	Order of the Compton polynomial, number of points in data convolution, order of the energy calibration polynomial, number of channels in fit
n	Number of parameters in fit
p	Number of light photons
$\underline{p}$	Parameter vector
t	Time
$\underline{\underline{v}}$	Variance-covariance matrix
x	Independent variable (channels)
x_o	Peak channel
$\underline{x}$	Unknown activities vector
ΔE	Energy difference
$\Delta h_{1/2}$	Pulse height interval
$\Delta \bar{n}^2$	Mean square value of the statistical variations in the number of the total ionization events

$\Delta\underline{p}$	Increase in $\underline{p}$
Δt	Cooling time
χ^2	Independent variable of the Chi-square distribution
χ^2_R	Reduced χ^2
Σ_C	Macroscopic cross section for Compton interaction
Σ_{ph}	Macroscopic cross section for photoelectric effect
α	Coefficient of the step vector
$\underline{\underline{\alpha}}$	Matrij of second derivatives
$\underline{\beta}$	Derivative vector
$\delta(x)$	Dirac function
ε	Average energy necessary for an ionization energy
$\underline{\underline{\varepsilon}}$	Error matrix
λ	Dummy variable
ν	Degree of freedom
σ	Standard deviation
σ^2	Variance of the energy calibration function
σ_e	Standard deviation of the Gaussian noise
σ_g	Standard deviation of the Gaussian function
σ_t	Tailing function parameter
σ^2_{AG}	Variance of the area of the Gaussian function
σ^2_{AT}	Variance of the area of the failing function
$\sigma_{ii'}$	Covariance of counts in i th and i' th channels
$\sigma^2_{p_j}$	Variance of parameter p_j
τ	Dummy variable

1. INTRODUCTION

Activation analysis is a method for determining concentrations of elemental constituents in a given sample by measuring the characteristic radiations emitted by the radioactive nuclei resulting from nuclear transmutations as a consequence of the irradiation of samples with nuclear particles. Mainly all of its applications are of a technological nature: purity control of high purity materials and semiconductors, soil analysis, geological prospecting, analysis of biological material for medicinal purposes, criminological and archaeological studies, etc.

Neutron activation analysis (NAA) was first discovered in 1936 by Hevesy and Levi[1], who determined the contents of Dy and Eu in a rare-earth mixture. Later work until 1950's was based essentially on the use of post-irradiation chemical separations involving the use of carriers to separate the elements of interest. These chemical separations are generally time consuming and various problems, such as the necessity of ensuring the isotopic exchange of the carrier with the trace element in the sample and the determination of the chemical yield of the separation, exist.

The chemical separations were necessary because the samples were usually counted with Gerger-Mueller counters making qualitative identification difficult. Decay and absorption measurements were the only means of improving the selectivity of the measurement.

The development of NaI scintillation spectrometry provided the birth of Instrumental Neutron Activation Analysis (INAA) which is a non destructive analysis method. In 1960's INAA was restricted by the rather poor resolution obtained with NaI scintillation detectors, making the interpretation of complex spectra difficult.

After the introduction of semiconductor detectors such

as Ge(Li) accompanied with the improvement in the couppled electronic systems such as amplifiers, Analog to Digital Converters, etc., the resolution was greatly improved and INAA took great strides. But the amount of data produced was very extensive and computers had to be introduced to process the spectra. Later with the developments in electronics and computer hardware and software engineering, rather inexpensive and highly precise systems consisting of detectors, multichannel analyzers and microcomputers became available and INAA could be performed, in most cases automatically, in a routine manner[2-10].

A gamma ray spectrum is an information complex consisting of basically signals of interest and noise whose character depends on the electronics of the system and is also statistical. The relevant information is represented by shapes called peaks superposed over an underlying continuum. The location of the peak channel contains information about the identity of the element and the area under the peak is a measure of the amount present in the sample.

The human-computer interface interactively tries to separate the signal from the noise as accurately as possible. However as the quality of the results expected decreases, the analysis can be made in a more automated manner.

The subject of this thesis is a computer program package BUCAASA, based on an earlier program CAASA by Çiftçioğlu[3] developed with basic intentions of keeping human intervention as minimal and the analysis as accurate as possible.

2. GAMMA RAY SPECTROMETRY

Activation analysis is based on the principle that when a material is irradiated by the nuclear particles produced in a suitable source such as a nuclear reactor or particle accelerator, some of the atoms present in the material will interact with the incident particles and undergo nuclear transformations producing isotopes of the same element or a different element which are usually radioactive. The subsequent emission of radiations from these nuclei is a characteristic of the particular isotope. If each different induced radioactivity can be distinguished or separated from all other radioactivities produced, then the amount of each radioactivity is a measure of the quantity of the parent isotope present in the material[4-10].

A radionuclide can decay to a daughter product by various ways: (a) Beta or beta and gamma ray emission, (b) positron or positron and gamma ray emission, (c) electron capture, (d) gamma ray emission (isomeric transition), (e) internal conversion, (f) alpha particle emission.

In general, the majority of the radionuclides formed by the radiative capture (n,γ) process undergo beta decay, and in most cases beta decay is associated with the emission of one or more gamma rays. Positron decay and electron capture are more probable in radionuclides that have an excess number of protons. Electron capture and internal conversion result from interactions of the nucleus with orbital electrons of the atom and these processes in general lead to the emission of X rays. In contrast γ ray emission results from nucleon configurational changes within the nucleus. However, the electron capture decay of some radionuclides is accompanied by γ ray emissions along with the X ray emissions. Alpha particle emission is favourable only in elements with a high mass number ($Z>83$) and, although it is frequently accompanied by γ ray emission, the detection of alpha particles in NAA is extremely limited because of their very short range. Overall,

most of the radionuclides that undergo decay by alpha, beta or positron emission and by electron capture also emit γ rays as a result of readjustment of energy content in the radionuclides from excited states to more stable states[5-9].

The γ energy spectrum, although specific for a radionuclide, is continuous, thus its use in multielemental analysis by INAA in a sample matrix is impractical. On the other hand, the energies of X and γ ray photons are discrete and characteristic of the radionuclides. Because of this important property γ and X ray measurements are widely used in multielemental NAA. Gamma ray measurements have, in general much wider applications in NAA than X ray measurements because γ rays emitted from most radionuclides have a wider range of energies (40-3000 keV) and have greater penetrating range than X rays (2-90 keV) thus their absorption in a sample is minimal. This property, coupled with recent developments in high-resolution and high efficiency semiconductors such us Ge(Li) detectors and the availability of high output neutron sources makes NAA-gamma spectrometry a powerful multielemental analysis technique.

2.1. Interactions of Gamma Rays With Matter

The quantitative detection of gamma ray photons is based upon their interactions with matter.

There are several ways in which gamma rays interact with an absorbing material; three, namely the photoelectric effect, the Compton effect, and pair production are the most important ones for the utilization of gamma spectrometry in NAA[2-3,5,7-10].

a) Photoelectric effect: For photons of low energy, i.e. less than 1 MeV, the most important interaction is the photoelectric effect (see fig.2.1). In this process the gamma photon interacts with the atoms of the matter and transfers

the whole of its energy to an orbital electron, which is consequently ejected with kinetic energy E_k

$$E_k = E_o - E_B \tag{2.1}$$

here E_o is the energy of the incident photon, and E_B is the binding energy of the electron.

The extent of the photoelectric interaction depends on both the energy E_o of the gamma radiation and the atomic number Z of the absorbing material. Photoelectric macroscopic absorption cross section is proportional to Z^5/E_o^3[3,12]. For this reason gamma ray detector materials must have a high atomic number[3,9].

b) Compton effect (Scattering): In a Compton interaction a gamma ray photon makes an elastic collision with an electron of the absorbing material. Since, usually, the energy of the incident photon is much greater than the binding energy of the electron, the electron may be accepted as free. Only part of the energy of the incoming photon is transferred to the electron. And as a result of this process, the photon is deflected from its original path. The energy E_θ of a photon scattered at an angle θ can be approximately evaluated with the equation

$$E_\theta = \frac{E_o}{1 + \frac{E_o}{mc^2}(1-\cos\theta)} \tag{2.2}$$

where E_o is the initial energy of the photon and mc^2 the rest energy of the electron (0.511 MeV).

The scattered electron loses its excess energy in the normal ionization process, however, the scattered photon either interacts with matter again (almost in every case photoelectric interaction) or escapes from the material (detector). This incomplete absorption of energy produces a a disturbing effect called Compton continuum which makes the

detection of lower energy photons difficult.

Compton macroscopic cross section is roughly proportional to Z/E_o. Since Σ_{ph}/Σ_C increases in proportion to Z^4, detector material must have high Z[3].

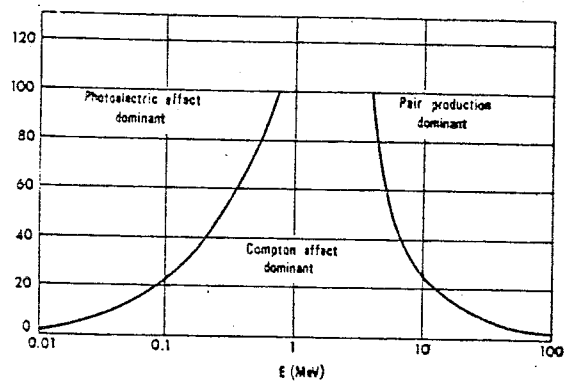

Fig.2.1. Regions of domain of photoelectric, Compton and pair production absorption processes[9].

c) Pair production: When a γ ray photon with energy in excess of 1.02 MeV passes near the nucleus of an atom, the photon can be annihilated in the strong electrical field with the formation of an electron-positron pair. The kinetic energy of the pair, i.e., the energy excess of 1.02 MeV, is absorbed as a result of the ionization process. The positron will be annihilated after an interaction with an electron, giving two 0.511 MeV gamma rays. The absorption probability of these γ ray photons, in general, is not high relatively, and depends on the detector geometry[3].

Pair production macroscopic croos section, increases over the threshold value of 1.02, and for energies greater than 4 MeV it is approximately proportional to ℓn E. For this reason this interaction forms the basis of detection of high energy photons.

2.2. Detection Systems

Any instrumentation for the detection of gamma photons

consists essentially of two parts: a detector and an amplifying-counting system[3,16]. The detector is the part where the interaction of photons with the matter occurs and an output pulse is generated, whereas the amplifying and counting system increases the detector's output pulse, counts the events and, eventually discriminates them according to their energies.

2.2.1. The Detectors

The main gamma ray detectors can be classified into the following groups (a) gas filled detectors (b) scintillation detectors (c) semiconductor detectors. However in NAA the most widely used detector types are scintillation and semiconductor detectors and only they will be examined.

Scintillation Detectors: Since 1896 some substances (called "phosphors") were known to have the property of emitting light flashes when bombarded with heavily ionizing particles[13,14,16]. Where the counting rates were not too high, the individual light flashes could be visually observed and counted under a microscope. These early scintillation detecting assemblies were called spintharoscopes[16]. The development of modern scintillation detectors was started in mid 1940s with the introduction of the photomultiplier tube to convert the light flashes into electrical signals[7,13,16].

The events leading to charge collection in a scintillation photomultiplier tube system can be summarized as follows. A gamma photon of energy E_o passing through gives an amount at E_1 of energy ($E_1 \leqq E_o$) to a charged particle, usually an electron. The electron dissipates a fraction AE_1 of its energy ($A \leqq 1$) in the phosphor the energy AE_1 is converted by the phosphor, with a conversion efficiency Y ($Y = pE_p/AE_1$), into p light photons, each having an average energy E_p, a fraction G of the emitted photons goes toward the photocathode of the photomultiplier. Moreover, because of the imperfect transparency J ($J < 1$) of the crystal to is own fluorescence radioation a fraction of them will be absorbed by the phosphor

itself[5] the photons reaching the photocathode are converted into photoelectrons with an efficiency P(P=number of photoelectrons per photon).

In conclusion, the total number of photoelectrons (n) produced at the photocathode as a consequence of the absorption of an energy E from a gamma ray is

$$n = \frac{E_1 AYJGP}{E_p} \qquad (2.3)$$

The photoelectrons are then multiplied by the photomultiplier and, if U is the overall amplification of the tube, the total charge Q at the final collector plate of the phototube is

$$Q = neU \qquad (2.4)$$

where e is the electronic charge. The proportionality between Q and E_1 allows the scintillation counter to be used as a spectrometer[18].

Scintillation detectors can be classified into two categories; organic and inorganic. Organic crystals can be subdivided into crystalline, liquid, plastic and glass scintillators, inorganic scintillators can be further subdivided into crystalline and noble gas detectors.

TABLE 2.1. Classification of Scintillation Detectors[15]

Organic				Inorganic	
Crystalline	Liquid	Plastic	Glass	Crystalline	Noble gas
Anthracene Transstilbene	(p-terpheny, PBO PDP, POPOP) in xylene, toluene Phenycyclohexane trimethylbenzene deceline	PBD, TP PP) in polypenyl- benzene polyvinyl toluene		NaI(TI) LiI(Eu) KI(Li)	Xe Kr

Among the desirable properties for good scintillators are high yield of fluorescent (immediately emitted) light, transparency for this emitted light, rapid light emission in combination with the lowest possible phosphorescence and finally, spectral distribution of the light suitable with regard to the sensitivity of existing photomultipliers[7,13].

The scintillator detector considered in this study is NaI crystal detector which contain 0.1 % thalium as active impurity and which have an emission band maximum at 4200 Å, decay time $0{,}25 \times 10^{-6}$ sec energy conversion efficiency 0.10-0.13 and is unusually transparent to its own scintillations[7,13,14,16]. They are relatively inexpensive and very stable during operation and their life span is fairly long ($\sim$10 years)[9]. Its comparatively high density and the presence of high atomic number iodine atoms (Z=53) give rise to large photoelectric, Compton and pair production absorption cross sections. For further information about NaI(TI) detectors see for example ref.7,13,14.

Semiconductor Detectors: Mc Kay[16] was the first to use a junction device for nuclear particle detection. He demonstrated that the p-n junction in germanium could be used to detect alpha particles, that the collection time for the charge was quite small, and that the energy required to produce an electron-hole pair in germanium was no more than about 3 eV.

The principle of operation of such detectors is basically the collection of the electric charge released in a solid by the absorption of photons or charged particles. The energy bands of the outer electrons in an insulator are schematically represented in Fig.(2.2a) all the electrons are accomodated in the valence and lower energy loands. The conduction band is usually empty and no movement of electric charge occurs. The valence and the conduction bands are separated by an energy gap usually wider than 3 eV. When an ionizing particle or radiation is absorbed by the insulator,

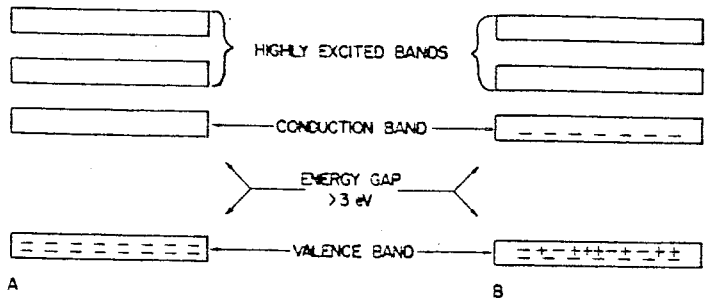

Fig.2.2. Schematic representation of the outer electronic bands of an insulator at rest (a) and after the absorption of an ionizing radiation (b)(16).

it can transfer to a single electron enough energy to raise it from the valence band to the highly excited bands; the removal of a negative charge from the valence band appears as a positive hole in the valence band. The highly excited states exist only about one picosecond after which the electrons are lowered to the conduction band, many more electron-hole pairs being formed in the process. The final situation is that of Fig.2.2b with positive holes at the valence band and an equal number of electrons in the bottom conduction band. When two electrodes are placed on two parallel faces of the crystal and a voltage is applied, the charge carries move to the appropriate electrodes and the induced charge can be detected.

Fig.2.3a shows the energy band of outer electrons in a semiconductor. It differs from the energy bands in an insulator in two aspects(1) The energy gap between conduction and valence bands is usually lower than 3 eV; and (2) impurity energy states are present close to the crystal's energy bands. The impurities, the presence of which allows a semiconductor to conduct, can be schematically gathered into two types (1) Donors who give electrons to the conduction band; and (2) Acceptors who accept electrons from the conduction band (Fig.2.3a). When a p-n junction is formed the distribution of free holes, conduction electrons, and impurity sites is that shown in Fig.(2.3b). When a reverse bias is applied, all the charge carriers move towards the electrodes

and a depleted region is formed where a space charge due to the ionized impurity sites exists (Fig.2.3c). This depleted region is the sensitive area of a p-n junction detector[3,16].

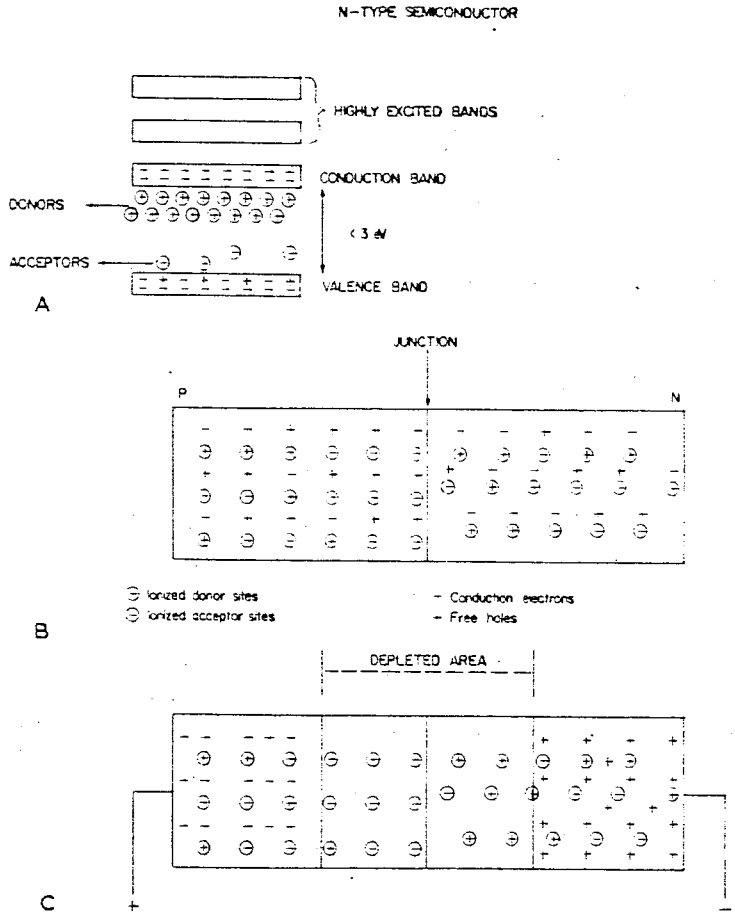

Fig.2.3. Schematic representation of the outer electronic bands of an n-type semiconductor (a) and of a p-n junction at rest (b) and after a reverse bias has been applied (c)[16].

In order to have good detection characteristics, the detector material should fulfill the following requirements as closely as possible

a) It should contain very few free electric charge carriers in the operating condition; such carriers may from a current and will tend to conceal or to spread the desired signal.

b) It should not contain a significant number of trapping centers capable of holding electrons or holes and

therefore giving rise to an incomplete charge collection. Furthermore, trapped electrons and holes produce an electric field in the crystal which, being opposite to the applied field, causes the current to decrease with time (polarization effect).

c) Recombination of holes and electrons during the charge collection process must be very small.

d) Charge collection must be as fast as possible; holes and electrons must therefore have high mobility.

e) In order to have a good resolution, the average energy to produce an electron-hole pair must be as low as possible.

Today the most commonly used semiconductor detectors are Ge(Li), Si(Li), and high purity Ge.

Since the depletion area is generally small in a pure semiconductor this region can be extended by adding Li to the semiconductor (Si and Ge). This process is called drifting. The drifted Li decreases the net density of holes in the depleted region and a wider depleted region is obtained in this region the free charge carriers formed as a result of ionization can move to p or n side of the system and thus form a current.

Since ΔE, the energy difference between the valence and condutcion band, decreases with increasing temperature, Ge(Li) dedectors can not be operated at room temperature, the reason for this is that the 0.6 eV energy is supplied with the available heat energy and a great number of hole-electron pairs are formed, in order to stop the diffusion of the drifted Li at the necessary level Ge(Li) dedectors must be kept and operated under $-200^{o}C$.

This problem can be avoided to some extent in high purity Ge dedectors which need cooling only under operation

conditions.

2.2.2. Comparison of Scintillation and Semiconductor Detectors

Here the basis of comparison will be their application to gamma spectrometry

a) Energy Resolution: The resolution of a detection system is a measure of the extent to which monoenergetic particles produce pulse heights of a single value. In gamma-ray spectrometry, the spread in the total energy peak is a measure of the resolution. The degree of the uniformity of the pulse heights is usually described by the quantity FWHM, the full width at half maximum[13]. The quantity FWHM is calculated as

$$FWHM = \frac{\Delta h_{1/2}}{h_{max}} \tag{2.5}$$

where h_{max} is the pulse height corresponding to the maximum in the curve while $\Delta h_{1/2}$ is the pulse height interval between the points at which one-half of the maximum value occurs. Since as it will be seen in the succeding chapters that the peak is generally represented by a Gaussian function, another definition of the FWHM can be given as

$$(FWHM)^2 = 4 \ln 4 \frac{\text{Variance}}{(\text{mean})^2} \tag{2.6}$$

The resolution is important because as the spread in the peak gets smaller, peaks whose average energies are very close can be distinguished and interferences will be minimized and even very complex spectra can be analyzed easily. Semiconductor detectors have a very definite advantage over scintillation detectors in this respect.

In both kinds of detectors, only part of the energy of the incident gamma ray photon is used for useful ionization leading to detection. This causes statistical variations in scintillation detectors in the number of electrons reaching

the first cathode and in semiconductor detectors in the number of electron hole pairs formed.

The ratio of the mean square value of the statistical variations observed in the number of total ionization events ($\Delta\bar{n}^2$) to the number of total ionization events is defined as Fano factor[3] and is given as

$$F = \frac{\Delta\bar{n}^2}{\bar{n}} = \frac{\Delta\bar{n}^2}{E}\,\varepsilon \qquad (2.7)$$

Here E is the energy absorbed and ε is the average energy necessary for an ionization event.

Combining eq.(2.6) and (2.7) a new definition

$$\Delta E = FWHM = 2\sqrt{\ell n\ 4}\ \varepsilon\ \sqrt{F,\bar{n}} \qquad (2.8)$$

is obtained.

Fano factor for scintillation detectors is 1, and for Ge(Li) detectors is less than 0.13[3].

TABLE 2.2. Comparison of Scintillation and Semiconductor Detectors[3]

		Type of Detector	
	Symbol	Semiconductor	Scintillation
Necessary energy for an ionization event (eV)		3	300
The number of electrons formed at energy E (eV)	$\bar{n}$	E/3	E/300
Standard deriation of $\bar{n}(\sigma=\sqrt{F\bar{n}}$		$\sqrt{F\,.\,E/3}$	$\sqrt{F\,.\,E/300}$
Resolution	FWHM	$2.335\sqrt{3\ F.E}$	$2.335\sqrt{300\ F.E}$
Internal gain	G	1	10^6
Charge (Coulomb)	Q	$E \times 0.5 \times 10^{-19}$	$E \times 0.5 \times 10^{-15}$

The resolution calculated from eq.(2.8) takes into consideration only the detector. However, since the detector

and the coupled electronic system are the sources of independent statistical variations that affect the total resolution obtained from gamma spectrometry they will be added as

$$(\Delta E)^2_{total} = (\Delta E)^2_{detector} + (\Delta E)^2_{electronics} \qquad (2.9)$$

In Ge(Li) detectors at low energies, the statistical spread in charge production is small and in most cases it is less than the spread contributed by the electronic noise. At higher energies, detectors frequently show insufficient charge collection due to trapping and or recombination. This results in a larger noise contribution from the detector than the estimated contribution, causing the FWHM vs photon energy characteristic to be linear rather than parabolic[17]. In semiconductor detectors, because of the vacancies and impurities in the crystal structure trapping regions are formed and some of the charge carriers are either lost from the current or they are retarded, this leads to a fluctuation in the current. The resolution decreases and in the lower energy side of the nearly Gaussian spectrum line shape an assymetric region is formed.

The other factors affecting the resolution are detector noise (Shot, Johnson, superficial leakage current noise) and detector capacity. For further information see for example ref[3].

b) Detector Efficiency: Another important parameter is detection efficiency. Although efficiency can be defined with respect to peaks produced in different interactions; in the range used in this study (40 keV-2 MeV) the photoelectric interaction cross section is considerably greater than Compton and pair production cross sections and only full energy photopeak efficiencies will be considered.

Every efficiency can be expressed as intrinsic, absolute or relative.

Total energy photopeak efficiency is the probabilty of counting a gamma ray photon of a given energy in the photopeak energy region.

Absolute total energy photopeak efficiency is defined as the probability of counting the photons from a soruce of known activity in the total energy photopeak region.

However the most frequently used definition of efficiency is the relative efficiency which is defined as the ratio of total energy photopeak efficiency to the efficiency in a different energy.

2.2.3. Auxillary Electronic Equipment

In general the amplifying-counting system consists of the high voltage supply, preamplifier, amplifier and the multichannel analyzer.

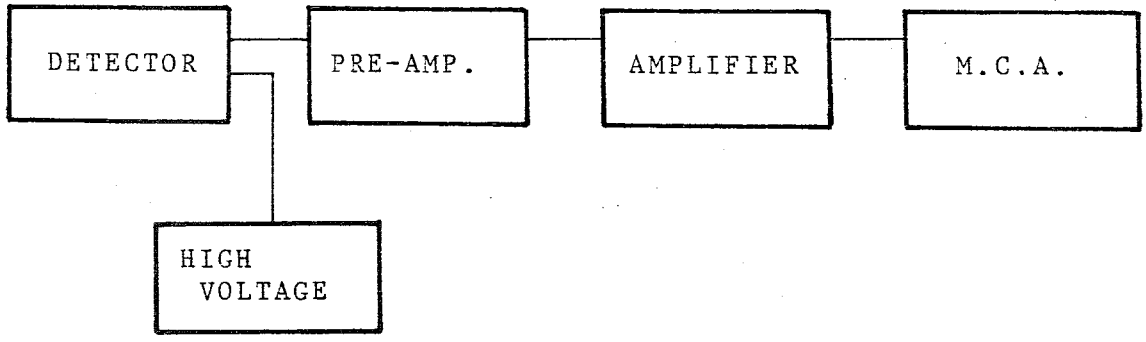

Fig.2.4. Schematic arrangement of a gamma spectrometry system.

Highvoltage power supply: Both scintillation and semiconductor detectors need high voltage for different reasons; while high voltage is used in the photomultiplier tube for the amplification of electron current, it is used in the semiconductor detector for the application of reverse bias.

Preamplifier: The signals obtained from detectors have very small amplitudes. For this reason they have to be amplified in, order to be analyzed. This can be performed in

three ways: (1) Voltage amplification (2) Current amplification (3) Charge amplification.

Amplifier: is used for obtaining variable voltage gain, reducing pulse pile up to a minimum.

Multichannel (pulse height) analyzer: is used for obtaining the frequency distribution histogram of the heights of the incoming pulses. The amplitude of each signal is measured, and with an Analog to Digital Converter, is transformed to an integer approximately proportional to the amplitude. And the information thus obtained is kept in the memory area in cells called channels. The counts in every channel shows the number of pulses whose amplitude value is in between the limits of each channel.

3. COMPUTER ANALYSIS OF GAMMA RAY SPECTRA

The phenomenon of a gamma photon dissipating all or part of its energy in the detector and eventually giving rise to a digital signal representing the γ-ray energy is called an event. The collection of numbers of events corresponding to a specific digital channel for a given radioactive source and measurement time is called the gamma ray spectrum[18] (GRS).

The term gamma ray spectrum is somewhat misleading because the detection of gammarays is indirect and the spectrum is actually the spectrum of the fast electrons produced by the interactions of the gamma photons with the detector material. As a result the spectrum contains continuous parts and escape peaks in addition to peaks corresponding to the mono energetie gamma rays.

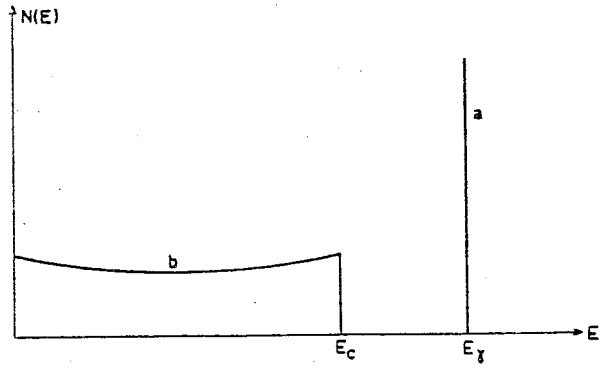

Fig.3.1. Theoretical gamma ray spectrum from a monoenergetic ray E_γ a) Dirac δ-functi , b) Compton distribution with Compton edge E_c[15,18].

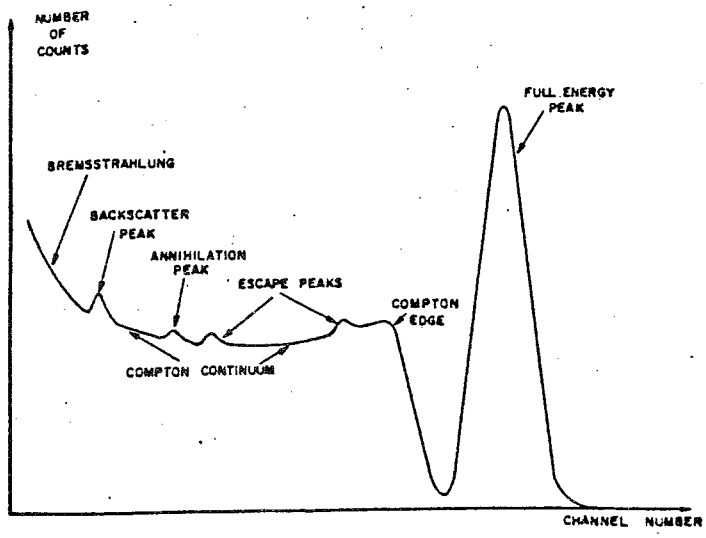

Fig.3.2. A typical Gamma ray spectrum[19].

In gamma ray spectra (GRS) two kinds of statistical phenomena have to be considered. The first is that due to the phenomenon of radioactivity itself because of which the number of events in each channel is governed by Poisson statistics.

$$P(m) = \frac{N^{M} e^{-N}}{m!} \tag{3.1}$$

where P is the probability of observing m events and N is the expected average of events. The magnitude of the statistical fluctuations is conventionally given by the estimate of the standard deviation on the Poisson distribution[3,19].

$$\Delta N = \sqrt{N} = \sqrt{m} \tag{3.2}$$

The second kind of statistics is related to the detectino system response. The gamma photons emitted by radioactive isotopes are monoenergetic. They have a very small spread in energy determined by Heisenberg's uncertainty relation. In channel units. They can be represented by a Dirac delta function at their specific energy E_{γ} as shown in Fig.3.1. Photons that do not dissipate all their energy are those interacting via the Compton effect with quasi-free electrons in the detector, giving the theoretical energy distribution of eq. (2.2). In addition, for energies greater than 1.02 MeV, those that produce pairs lead to single and double escape peaks.

The physical processes that govern the creation of light photons or electron-hole pairs in detectors, those that produce the electrical discharge and finally those that transform the latter into a digital binary number are all statistical in nature[3,19].

A GRS consists of a large senes of integer numbers, a few hundred to a few thousand depending on the kind of detector used, arranged in a certain sequence. The extraction of the information that is of value from the numbers and their sequence is the art of analysis of a spectrum. Throughou the text sought after information will be denoted as signal,

and noise (background), denotes the bits of data that obscures the recognition of the presence of the valuable information. In a GRS the noise has partly a composite statistical and partly a continuous nature which is due to Compton effect, pulse pile-up, Bremsstrahlung, etc .

In regions of the spectrum, hereafter denoted as peaks, the signal-to-noise ratio is stronger than in continuum parts, at least with respect to a given specimen. In the following the concept of complex will be used for a group of channels in an observed GRS that are related by the fact that several sources of information contribute to such a group of channels, e.g., true detector background, Compton distributions from several gamma lines, statistical fluctuations of different origin and one or more unique signals belonging to γ rays. If a complex contains several unique signals, each of them expenences the sum of others as part of the noise.

The signal has a qualitative as well as a quantitative aspect. The first is the information about the species (isotopes or elements) that have contributed to the complex and the second, about the species (isotopes or elements) that have contributed to the complex and the second, about the amount of each speciemen present. The exact position of the peak maximum determines the qualitative aspect, whereas the integrated area under the peak determines the quantitative aspect. Both these values can be determined only after the presence of a peak region has been recognized. The latter involves the concept of peak shape in numerical terms, such that signal and background can be accurately separated. The recognition of the peak shape is also important for a judgement of the uniqueness of the signal.

All numerical values extracted from the GRS are influenced by the statistical part of the noise. As a result only estimates of the true value can be given, accompanied by standard deviations representing a measure for the precision of the estimate. The statistical part of the noise also puts

a limit on the ability to recognize a peak region, which leads to the concept of limit of detection.

However, the amount of valuable information that can be extracted from a given GRS is determined by the nature and the quality of the measurement. The signal-to-noise ratio cannot be improved after the measurement, no mather how sophisticated or powerful the mathematical techniques used for the analysis[19].

Especially with the starting of wide spread use of Ge(Li) detectors, because of their high resolution; multichannel analyzers with 4096, even with 8192 channels became common place; such vast amount of data could not be manually analyzed in a reasonable time, thoroughly and in an efficient manner. With parallel advances in computer hardware and software, the introduction of computers and computer codes into the field of gamma ray spectra analysis was not only logical but became a must.

Many computer codes have been written for the analysis of NaI(TI) spectra these generally require a knowledge of the possible component isotopes in the spectrum and also the availability of standard pulse height spectra of these components[18,19]. However because of their high resolution Ge(Li) spectra need a special treatment which is, in most cases easier, faster, and more automatic than analyzing NaI(IL) spectra[3,19,22-26].

A computer code developed for the analysis of gamma ray spectra must have the following qualifications[3,26].

(a) The method must be programmable to be able to perform an automatic analysis.
(b) The code must be able to handle data obtained under various experimental conditions.
(c) The code must be able to analyze highly complex spectra.
(d) It must include information about error limits.

(e) The execution time and memory requirements should be kept to a minimum.

In general the steps in computer analysis are as follows:

a) Peak identification
b) Fitting a function to the peaks
c) Energy and efficiency calibration
d) Identification of nuclides producing the spectrum.

3.1. Peak Identification

Inorder to minimize statistical variations in countrate various filtering methods are used to smooth the spectral data the earliest and the most commonly used is the method of Savitzky and Golay[20] which is a least squares fitting of a polynomial to the data points carried out by their convolutio with properly chosen sets of integers.

$$C_i' = \frac{1}{N_m} \sum_{j=-m}^{+m} a_j C_{i+j} \qquad (3.3)$$

Here C_i represents counts in channel i a_j are the convolution integers, N_m is the normalisation factor (which are given tabulated form in ref[20]) and C_i' is the smoothed value of channel i. Here for the smoothing process 2m+1 points are used. Smoothing the whole spectrum can be carried out by taking channel i+m+1 and dropping channel i-m-1. In this way not only a smoothed spectrum but the derivatives of the spectrum can also be obtained.

The number of channels used in convolution and the order of polynomial are the important parameters. For optimum performance the number of channels included in smoothing should be equal to FWHM and a cubic polynomial for calculating the first denvatire for peak detection purposes must be used[3, 21,22].

Several authors have used Fourier transformations However, these are method relatively complex to be programmed need more execution time and memory and by no means much superior considering their complexity. For a discussion of these methods see ref[19].

In BUCAASA smoothing is used as an aid in determining the first derivative spectrum.

The methods of peak detection are classified in five groups i.e. those based on (1) The detection of a relative maximum in the spectrum, (2) The use of the first derivative, (3) The use of the second derivative, (4) cross correlation methods and (5) visual methods[24].

As a result of the IAEA G-1 Intercomparison of methods for processing Ge(Li) Gamma-ray Spectra[24], it was found that in the best hands, visual methods and the use of the second derivative are capable of marginally superior performance.

The performance of a peak detection method can be evaluated by considering its ability to detect true peaks, reject spurious ones, and detect multiplets.

3.1.1. Peak Location by Finding Relative Maxima

This is the simplest method. The spectrum is first scanned and the possible peaks are selected by finding those channels where the counts fulfill the conditions[19]:

$$C_{i-2} < C_i - K\sqrt{C_1} \text{ and } C_{1+c} < C_i - K\sqrt{C_i(i)} \qquad (3.4)$$

Where K is an experimentally determined constant.

This method is fast and can be programmed easily but often overlokes peaks, especially small ones, and cannot separate multiplets[19].

3.1.2. Method of Smoothed First Derivative

If the number of counts is regarded as a continuous function of the channel number, then the first derivative changes sign at the top of the peaks. In approaching a peak it is positive and after the peak it becomes negative in the next few adjacent channels. This can also be used for locating peaks. The computer looks for groups of channels such that the smoothed first derivative C' fulfills the following criteria

$$\begin{aligned} C'_i &\leq 0 \\ C'_{i+j} &< 0 \qquad j = 1,2,\ldots,r \\ C'_{i-j} &> 0 \qquad j = 1,2,\ldots,\ell \end{aligned} \tag{3.5}$$

Being able to recognize significant peaks but to disregard statistical fluctuations, r and ℓ must be chosen in accordance with energy resolution[19].

This method with a few extensions and checks for detecting multiplets is used in BUCAASA for peak detection.

3.1.3. Generalized Second Differences

In this method developed by Mariscotti[25], the peaks are assumed to be described by Gaussian functions and the background to be approximated by a linear function within short intervals:

$$Y(x) = G(x) + a_1 + a_2 x \tag{3.6}$$

where G(x) represents a Gaussian function if a peak is present and is zero otherwise, a_1 and a_2 are constants describing the background.

If it is assumed that Y(x) is a continuous function its second derivative Y"(x) becomes independent of the background and it vanishes for any interval in which there is no peak. Therefore a peak would be located wherever $Y''(x) \neq 0$.

Because of the discrete nature of data, the second derivative must be replaced by the second difference

$$S_i = C_{i+1} - 2C_i + C_{i-1} \tag{3.7}$$

which, like Y"(x) should be only different from zero in the neighborhood of a peak. However, since the data C_i are defined within a statistical error, The S_i values will fluctuate around the expected value according to their standard deviation and, if the expected value S_i is comparable to its standard deviation no peak searching can be attampted.

To detect weaker peaks S_i is replaced by a "smoothed" second derivative expressed by

$$S_i(w) = \sum_{x=i-m}^{1+m} S_j \tag{3.8}$$

where the normalization constant is omitted.

This process is repeated, the averages are averaged and the generalized second difference is defined as

$$S_i(z,w) = \sum_{j=1-m}^{i+m} \cdots \sum_{k=1-m}^{1+m} S_k \tag{3.9}$$

$$z$$

The optimum pair z,w is found to be $z_o = 5$ and $w_o = 0,6$ FWMM thus $S_i(z,w) \approx 0$ at the background and Compton continuum, differs significantly from zero around a peak and is proportional to the second derivative of a Gaussian, if the peak is a full energy peak superimposed on a linear background.

A peak is identified in channel i if $- S_i(z,w) > 2\Delta S_i(z,$ It is then tested to determine whether it is a real full energy peak or a Compton shoulder.

3.2. Detector Response Functions for Determination of Peak Position and Area

Inorder to assess precise values of energy and intensity. The spectrum line shape of a peak must be known, at least approximately. Although, theoretically an event corresponding to a given gamma ray energy is represented by a Dirac δ-function the statistical nature of radioactivity and the detection system as a whole, causes a widening of the line and thus forming the shape called peak as described in sub section 4.1. Here after until otherwise stated only detector response functions for Ge(Li) will be considered Generally, a response function is of the form

$$Y(x) = F(G(x)) + C(x) \tag{3.10}$$

Here Y(x) is the approximate detector response function G(x) is a Gauss function

$$G(x) = A\exp\left[\frac{(x-x_o)^2}{2\sigma^2}\right] \tag{3.11}$$

and C(x) is a polynomial:

$$C(x) = \sum_{i=1}^{n} a_i x^{i-1} \qquad n=1,2,3 \tag{3.12}$$

However, the main difference lies in the construction of F which is a function of the Gaussian G(x)[3]: Usually in order to compensate for trapping effects the Gaussian is modified with the addition of exponential tailing functions. For a general discussion of these methods see ref[3].

The requirements for a response function can be summarized as follows.

a) The physical events that affect the shape of the peak must as a whole be represented.

b) Since the resultant function is nonlinear nearly in every case it must have as few parameters as possible inorder for the minimisation method (generally a nonlinear least squares fitting) to work reliably and fast. These methods are iterative, so initial guesses must be supplied, as the number of parameters increase this becomes not only tedious but sometimes impossible.

c) And the method must be programmable easily.

In CAASA[3] and in BUCAASA an analytical approximation of the form given in eq (3.10) is constructed.

3.2.1. The Analytical Approach for Detector Response Function[3]

Generally in experiments investigating physical events, the data obtained is distorted due to the effects originating both from the system used and from the event itself. In such a case the variation of a physical quantity can be represented by a convolution integral

$$F(x)=\int_{o}^{x} G(x-\tau)D(\tau)d\tau = \int_{o}^{x} G(\tau)\, D(x-\tau)d\tau \qquad (3.13)$$

Here G(x) is the function representing the pure data and D(x) is the distortion function, and x is an independent variable. The ideal detector response in the detection of monoenergetic gammarays is a Dirac δ-function. Assuming that D(x) is a Gaussian function (eq.3.11) where x the independent variable representing channels, the detector response can be found as;

$$F_G(x)=A\int_{o}^{x_o} \delta(\tau-x_o)\exp\left[-\frac{1}{2}\left(\frac{x-\tau}{\sigma_g}\right)^2\right] d\tau = A_{exp}\left[-\frac{1}{2}\left(\frac{x-x_o}{\sigma_g}\right)^2\right] \qquad (3.14)$$

Where A is the amplitude, σ_g is the standard deviation being a function of energy (see Table 2.2) and x_o is the channel where the Gaussian is a maximum.

The distortion due to trapping effects can be represented by the convolution integral of an exponential function with the Gaussian noise:

$$f_1(x) = A_1 \exp\left[B(x-x_o)\right] \tag{3.15}$$

$$f_2(x) = A_2 \exp\left(-\frac{1}{2\sigma_e^2}\, x^2\right) \tag{3.16}$$

$$T(x)=f_1(x)*f_2(a) = A_2 \int_{-\infty}^{x_o} \exp\left[B(\tau-x_o)\right].\exp\left[-\frac{1}{2}\left(\frac{x-\tau}{\sigma_e}\right)^2\right]d\tau \tag{3.17}$$

where B is a constant, $A=A_1.A_2$ and σ_e is the standard deviation of the Gaussian noise.

$$\text{Defining } U = \frac{\tau-\left(\frac{x}{\sigma_e^2} + B\right)\sigma_e^2}{\sigma_e} \tag{3.18}$$

and

$$\text{erf}(x) = \frac{1}{(2\pi)^{1/2}} \int_o^x \exp\left(-\frac{1}{2}\, t^2\right) dt \tag{3.19}$$

and

$$\text{erfc}(x) = \frac{1}{2} - \text{erf}(x) = \frac{1}{2} - \frac{1}{(2\pi)^{1/2}} \int_o^x \exp(-\tfrac{1}{2}t^2)dt \tag{3.20}$$

$$T(x)=A(2\pi)^{1/2}\, \sigma_e \exp\left[B(x-x_o)+\frac{1}{2}\, B^2\sigma_e^2\right]\text{erfc}\left(\frac{x-x_o}{\sigma_e} + B\sigma_e\right) \tag{3.21}$$

is obtained

However, in BUCAASA instead of T(x) a new tailing function

$$D(x) = A_t \exp\left[B(x-x_o)\right].\{1-\exp\left(-\frac{1}{2\sigma_t^2}\, (x-Cx_o)^2\right)\}.\delta \tag{3.22}$$

is considered as can be seen from fig.3.3 D(x) is very similar in nature to T(x) and easier to implement in the program and better error estimates for D(x) can be obtained. Furthermore the time required for the minimisation process is less than the time required for minimising a function containing T(x).

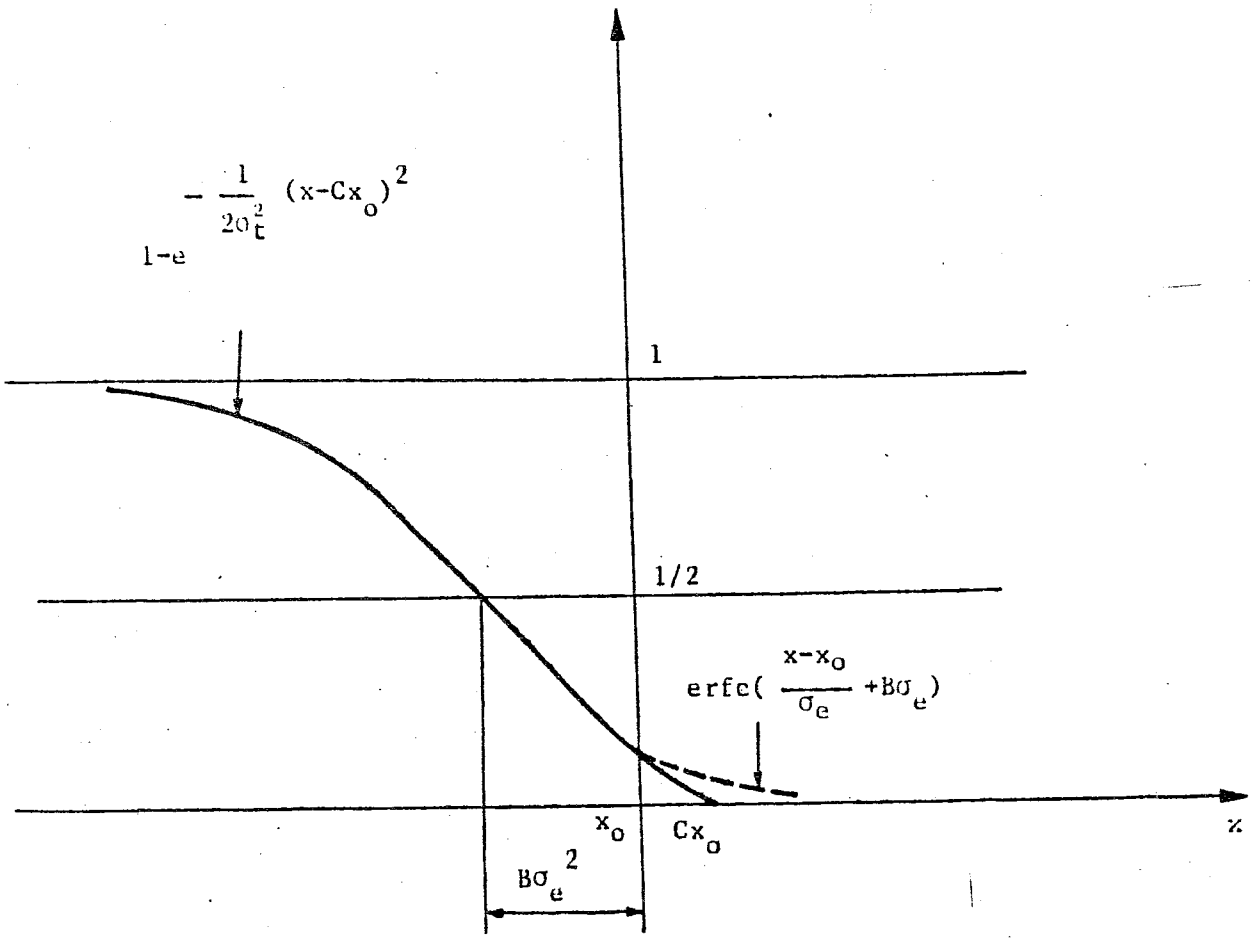

Fig.3.3. Comparison of T(x) and D(x)

Here C>1 and

$$A_t = A(2\pi)^{1/2}\,\sigma_e \exp\left(\tfrac{1}{2}B^2\sigma_e^2\right) \qquad (3.23)$$

$$\delta = \begin{cases} 1 , & x < Cx_o \\ 0 , & x \geq Cx_o \end{cases}$$

In GRS, although the ideal response is a Dirac δ-function the statistical variations, which are assumed to have a Poisson distribution in the number of ion-pairs cause the line shape to spread and be more or less a Gaussian. However, the probability of detection of monoenergetic photons in the lower energy side of the Gaussan is greater than the

probability of detection in the higher energy side. Because the photon whose initial energy is E_o and which corresponds to an energy E in the Gaussian, can be detected at any energy level less than or equal to E and however this is not true for energies greater than E. For this reason if it is assumed that the distribution of photoelectron energies is a Gaussian function and the probability of detection of a photon for all energise less than or equal to E is constant equal than a step function component over the Compton continuum is formed see fig.3.4.

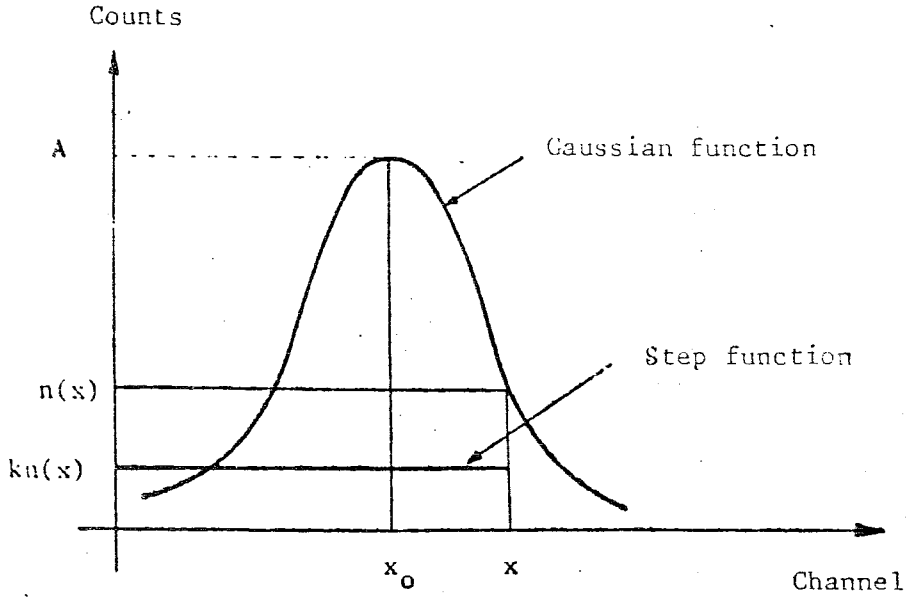

Fig.3.4. Schematical representation of step effect[3].

$$S(x) = A \int_{x}^{+\infty} k \exp\left|-\frac{1}{2\sigma_e^2}(\lambda - x_o)^2\right| d\lambda \qquad (3.24)$$

and evaluation of the integral gives;

$$S(x) = h \cdot \operatorname{erfc}\left(\frac{x-x_o}{\sigma_e}\right) \qquad (3.25)$$

where

$$h = (2\pi)^{1/2} A k \sigma_e \qquad (3.26)$$

Here σ_e is taken to be equal to σ_g because the correlation between S(x) and D(x) must be minimum[3].

The Compton continuum is represented by a quadratic

polynomial.

$$C(x) = \sum_{i=1}^{3} a_i x^{i-1} \qquad (3.27)$$

Finally gathering all terms, the detector response function is obtained as

$$F(x) = \sum_{i=1}^{3} a_i x^{i-1} + \sum_{j=1}^{n} \{h_j \operatorname{erfc}\left(\frac{x-x_{oj}}{\sigma_g}\right) + A_{tj} \exp\left|B(x-x_{oj})\right| \left(1-\exp\left|-\frac{1}{2}\left(\frac{x-Cx_{oj}}{\sigma_t}\right)^2\right|\right)\delta + A_{gj} \exp\left|-\frac{1}{2}\left(\frac{x-x_{oj}}{\sigma_g}\right)^2\right|\} \qquad (3.28)$$

where n, is the number of photopeaks in a photopeak group h the amplitude of the step function, A_g the amplitude of the Gauss function and x_o is the peak channel position.

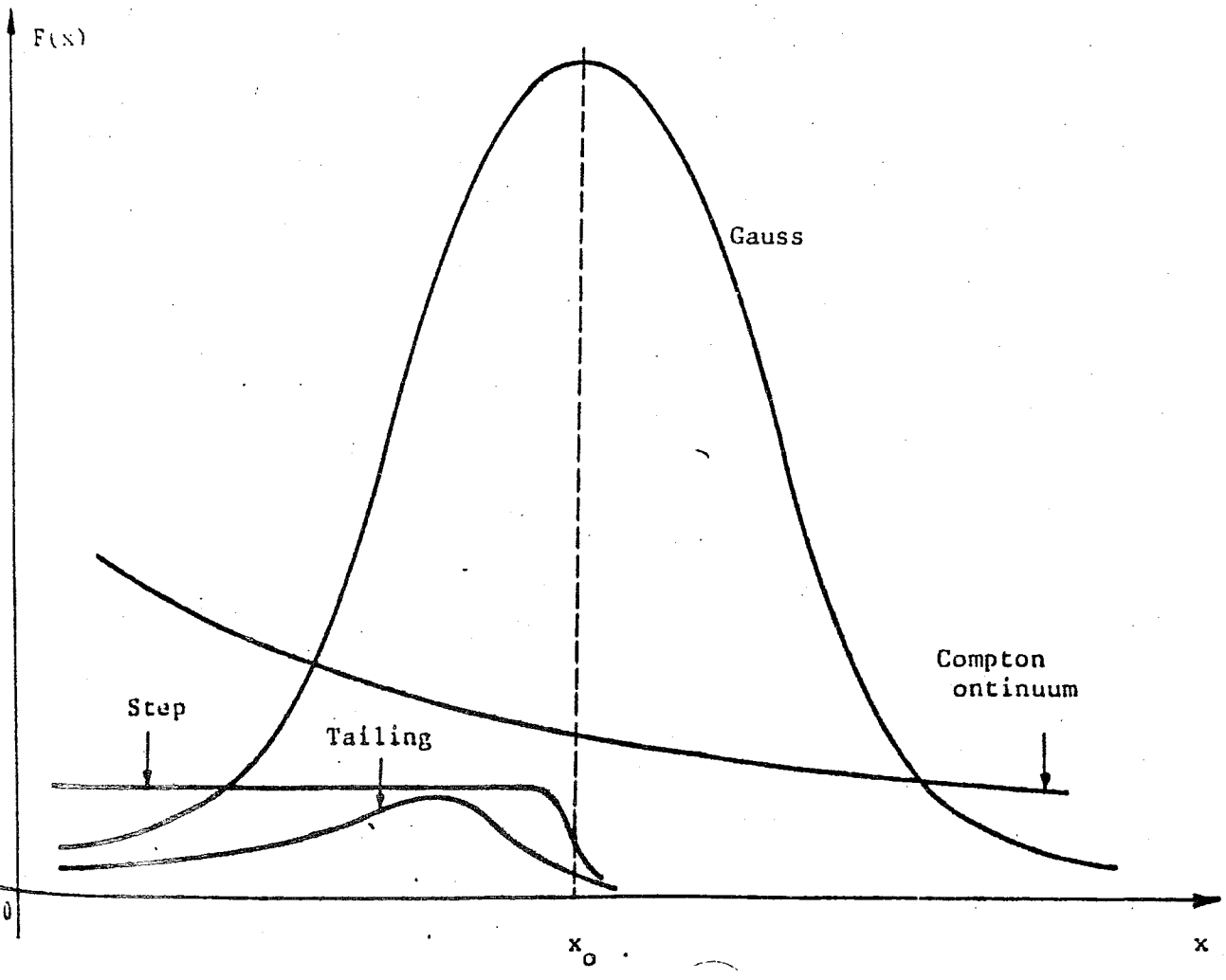

Fig.3.5. Schematic representation of the components of the photopeak function F(x) for a singlet[3].

The fitting of the function F(x) given by eq. (3.28) to the experimental data points is done with the variable metric minimisation method (see Appendix A).

The peak areas are determined by calculating the area under the fitted function:

The area of the Gaussian is

$$A_G = A_g \int_{-\infty}^{+\infty} \exp\left[-\frac{1}{2}\left(\frac{x-x_o}{\sigma_g}\right)^2\right] dx = (2\pi)^{1/2} A_g \sigma_g \qquad (3.30)$$

and the area of the tailing function is given by

$$A_T = \int_{o}^{Cx_o} A_t \exp\left[B(x-x_o)\right]\left\{1-\exp\left[-\frac{1}{2}\left(\frac{x-Cx_o}{\sigma_t}\right)^2\right]\right\} dx$$

$$= \frac{A_t}{B} \exp(-Bx_o)\left[\exp(BCx_o)-1\right] +$$

$$+ (2\pi)^{1/2} A_t \sigma_t \exp\left[\frac{1}{2} B^2 \sigma_t^2 + Bx_o(C-1)\right]\left[\operatorname{erf}(B\sigma_t) - \operatorname{erf}\left(\frac{Cx_o}{\sigma_t}+B\sigma_t\right)\right] \qquad (3.31)$$

Where the index j is omitted for clarity and erf(x) is defined in eq. (3.19).

The peak location which is an actual parameter of F(x) is directly determined by the fitting process.

The goodness of fit is calculated by

$$\chi^2_\nu = \frac{1}{n-m} \sum_i \frac{1}{Y_i}\left[Y_1 - F(x_i)\right]^2 \qquad (3.32)$$

where Y_i are the experimental data points, n the number of channels in this fit and m the number of parameters in the fitting process.

In BUCAASA another test developed by Aarnio et al[27] the error correlation is also used:

$$R = \frac{1}{(n-m)} \sum_{i=1}^{n} \Delta_i \Delta_{i+1} \quad , \quad (\Delta_{n+1} \overset{\text{def}}{=\!=} \Delta_1) \tag{3.33}$$

where

$$\Delta_i = \frac{Y_i - F(x_i)}{\sqrt{F(x_i)}} \tag{3.34}$$

A good fit is indicated by $\chi^2_R < 10$ and $|R| \leq 1$.

The estimation of errors due to the parameters in the fit is done by employing the inverse of the Hessian matrix: The diagonal elements give the variances of the parameters whereas the off diagonal elements give the covariances (see Appendix B).

3.3. Energy and Efficiency Calibration

Experimental values of channels and corresponding energy and efficiency values obtained under identical conditions are supplied to the computer by the user.

In general the energy-channel relation can be represented by a parabolic polynomial[3,22,26] and the energy efficiency relation by a function of the form;

$$F = pE^q \tag{3.35}$$

where p and q are constants for a given detector and source geometry, over the range 100 keV to several MeV[22].

The energy calibration function is given by

$$f(\underline{P},x) = \sum_{i=1}^{3} p_i x^{i-1} \tag{3.36}$$

where $\underline{p}$ is the parameter vector, x is the channel number and $f(\underline{p},x)$ is the corresponding photon energy. For the fitting process the linear least squares method is employed. The uncertainties in the determination of energy is mainly due to the spectromoter system therefore the uncertainties in

system therefore the uncertainties in channel may be taken to be equal to

$$\sigma_i^2 = \sigma^2 \cong S^2 = \frac{1}{M-(m+1)} \sum_{i=1}^{M} \{e_i - f(\underline{P}, x_i)\}^2 \qquad (3.37)$$

here, M-(m+1) is the degree of freedom related to fitting M data points to a polynomial of (m+1) parameters e_i is the photopeak energy, $\sigma_i = \sigma$, is the standard deviation corresponding to photopeak energy e_i, and S^2 is the variance in the fitting process.

The errors in the determination of the energies is given by

$$\sigma_f(x_i) = \left(\sum_{\ell} \sum_{k} \frac{\partial f(\underline{P}, x_i)}{\partial P_\ell} , \frac{\partial f(\underline{P}, x_i)}{\partial P_\ell'} \varepsilon_{\ell k} S^2\right)^{1/2} \qquad (3.38)$$

where $\underline{\underline{\varepsilon}}$ is the error matrix calculated by taking $\sigma_i = 1$ (see Appendix B)

3.4. Nuclide Identification

The total number of known X and γ ray emissions include more than 48000 lines emitted by more than 2000 nuclides[28]. In any practical analysis, the totality of these is impractical and therefore not required, specially selected reference libraries should be used in different applications. Such libraries should contain all nuclides which may appear in the spectra under analysis, and also a minimum of those which will not appear in any case[29].

Accordingly, different libraries have been developed for different applications using ZZ-GAMDAT-78[28] data collection as a data base.

There are basically two approaches to computer aided nuclide identification: a) Simple line matching, b) identification

with associated lines.

3.4.1. Simple Line Matching

Simple line matching is performed by comparing the calculated peak energies with the gamma-ray reference library one at a time.

The simple line matching is a tentative identification of possible peaks. Generally the computer codes accept a certain maximum energy difference between the identified peak energies and the gamma-ray reference library energies the value of which is supplied by the user.

The final result is obtained by inspecting the printout of the program. For an experienced worker this printout provides a fairly accurate and reliable method of identifying the component nuclides in the measured spectrum without the chore of manual computation.

3.4.2. Identification With Associated Lines

This method is based on the method suggested by Gunnink and Niday, however, the present implementation[29] has several improvements such as the tests to insure that only the right nuclides are accepted.

In this method a working matrix, which contains the branching intensities, is formed by comparing the computed peak energies in the analyzed spectrum with those in the reference library. If the computed energy of the first peak differs by less than a tolerance value, E_{tol}, from a line in the library, the program starts forming the first column of the working matrix. As the nuclide in the library may also have other gamma energies besides the first one, all those that lie in the comparison interval are compared with the measured peaks. For every library energy of this nuclide, the smallest difference between this energy and the fitting peak energies that determine the row of the working matrix where the branching intensity is written, is computed.

The process is then repeated for the next nuclide until all the peaks in the spectrum have been compared with the library. The resulting working matrix has the dimensions $N_X M$, where N is the number of fitted peaks in the spectrum and M the number of possible nuclides.

Along with the working matrix three vectors are formed. They contain the symbols of the possible nuclides, the confidence values for each nuclide, and the numbers which are used to write the accepted nuclides into the output matrices in the order of increasing proton number and atomic weight.

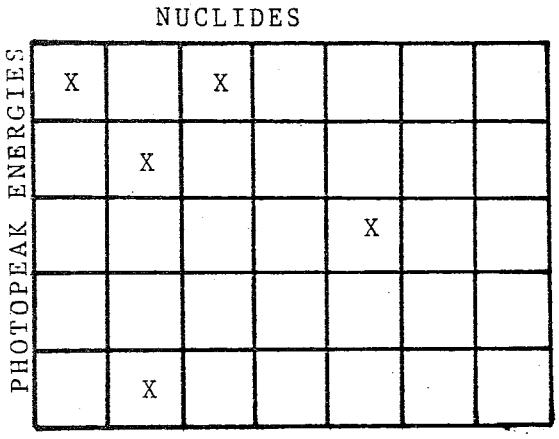

Fig.3.6. The structure of the working matrix. Non zero elements of the matrix are marked with crosses[29].

The confidence value of a nuclide is an index computed to discriminate against probable nuclides. The initial value of the confidence index is 1. For each considered library gamma energy it is multiplied by the function.

$$f(\Delta E) = \exp\left[-(0.16/E_{tol})(\Delta E)^2\right] \qquad (3.39)$$

where ΔE is the smallest computed difference between the library energy considered and the peaks in the spectrum. To account for small intensities a reference activity for each nuclide is computed. This reference activity is used to check whether such a small intensity should be seen in the spectrum or not.

After all the gamma energies of a nuclide that lie in

the interval of comparison is checked, the confidence value is multiplied by the function.

$$g(\Delta t) = \exp\left[-5.116856\times10^{-3}\ (\Delta t/T_{1/2})\right] \qquad (3.40)$$

Where Δt is cooling time and $T_{1/2}$ is the half life of the nuclei. This assures the removal of nuclei with too short a half life to exist in the sample at the time of counting.

Finally, the confidence value is compared with a given limit. If it is smaller, the matrix column under consideration is cleared.

The above process is repeated for all nuclei and an interference matrix is formed, the unknown activities of the nuclides are determined by solving the matrix equation

$$\underline{\underline{B}}\ \underline{x} = \underline{D} \qquad (3.41)$$

for the unknown activities $\underline{x}$; $\underline{\underline{B}}$ is the interference matrix containing the branching intensities of the nuclides that were found to interfere with each other, and $\underline{D}$ is a vector that contains the intensities of the peaks in the experimental spectrum that are assumed to be due to these nuclides. The equation is solved using the weighted linear least square technique. The weighing factors are the reciprocals of the errors of the calculated peak intensities. This takes into account the fact that stronger peaks can usually be calculated more accurately and therefore give better results than smaller peaks.

There are two acceptance criteria: none of the activities of the nuclides can be negative and none of the errors of the calculated activities more than 50 %. If one or both of these conditions are not fulfilled for some nuclide that nuclide is removed and the interference matrix is formed a new.

This process is repeated for all interference sets[29].

4. PROGRAM PACKAGE BUCAASA

The starting point of BUCAASA* is CAASA program developed by Ciftcioğlu[3], which was coded in FORTRAN-IV and ran on a PDP 11 microcomputer. The major shortcomings of CAASA were its lack of efficiency calibration and nuclide identification routines and to some extent the rather poor way of supplying initial guesses for the parameters in the fitting process.

The main aim in preparing the package was to make the programs user friendly-interactive- and more precise and accurate. For a general user although the principles behind the analysis are important, fast, automatic and easy to use programs are of greater value, for such programs are mainly used in routine work.

The package consists of four main programs, although it could be put into a single program format which was possible on CDC Cyber 170/815 system at Boğaziçi University where it was developed, however in order to make its adaptation to microcomputers easier it was decided to separate the self consistent parts. Though hierarchial in order, each program can be used separately and even for a mainframe computer considering the costs of occuppying central memory area this division is highly beneficial and it allows the user to evaluate the results of each part and perform additional runs towards a more accurate analysis especially in academic work. For routine work, excepting cases of important multiplets even part two (BUCAASA/FIT) may not be used at all and a single consecutive run of parts one, three and four would be sufficient.

In deciding for the programming language to be employed FORTRAN 77 was chosen due to the familiarity of general users with earlier versions of FORTRAN language and its possibility of allowing structural programming thus leading to programs

---

* BUCAASA; Boğaziçi University Computer Aided Automatic Spectrum Analysis.

which are easily understood and can be maintained and modified with minimum cost. The present implementation has no extensions to the language meaning that it will run on any system which has a FORTRAN 77 compiler with no major modifications. However the fact that CDC Cyber 170/815 uses 60 bit words, hence its single precision is double precision for other systems must be taken into consideration, but this aspect is important only in part two: BUCAASA/FIT.

4.1. Organisation of BUCAASA

BUCAASA consists of four programs

a) BUCAASA/FIND: performs peak identification and calculates peak areas, and runs an estimate of some of the necessary parameters for the fitting process such as peak channel, right and left limits of peaks, FWHM and coefficients of the Compton polynomial.

b) BUCAASA/FIT: Performs the fitting of a function to the peak data points; the function can be a Gaussian, plus tailing or a Gaussian plus tailing plus a step function with a polynomial Compton background which are chosen by changing the values of a single parameter. As a result of this fitting, exact channel location of the peak and its area together with relevant error estimates are calculated.

c) BUCAASA/CALIBR: Performs the necessary energy and efficiency calibrations using the results of part II and calibration data supplied by the user.

d) BUCAASA/NUCLIDE: Through the associated lines technique[29] using purpose built libraries identifies the component isotopes and their activities.

The programs pass data to each other with standard FORTRAN 77 random access files. The usage of random access files permit individual runs for each photopeak group thus

leading to improvement in the results.

Each program needs very few input parameters which direct the flow of the program, i.e. the thorougness of the analysis and which are supplied interactively. Although the programs can run batchwise, especially in BUCAASA/FIT, interactive usage helps to produce better results.

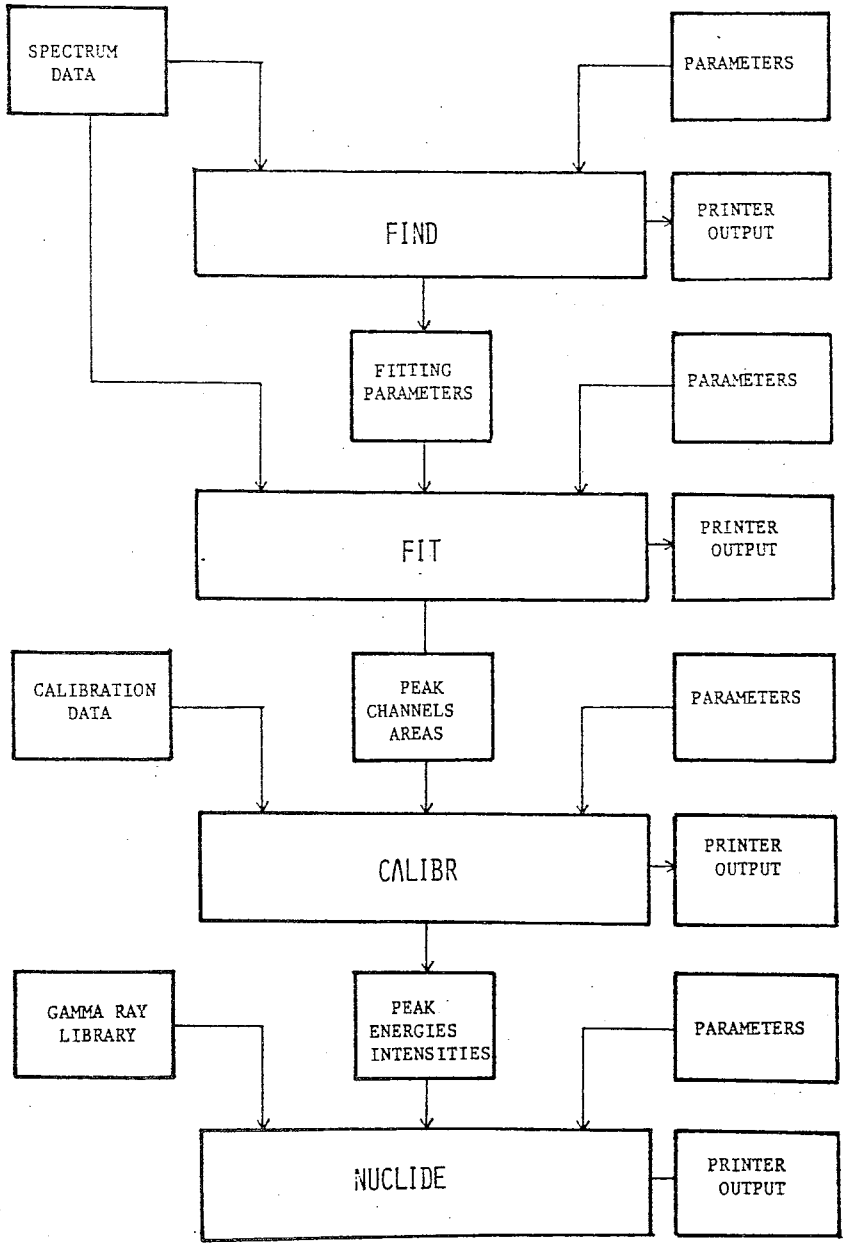

Fig.4.1. Schematical organization of BUCAASA with associated input and output files.

4.2. Program BUCAASA/FIND

BUCAASA/FIND consists of one main program and two subroutines.

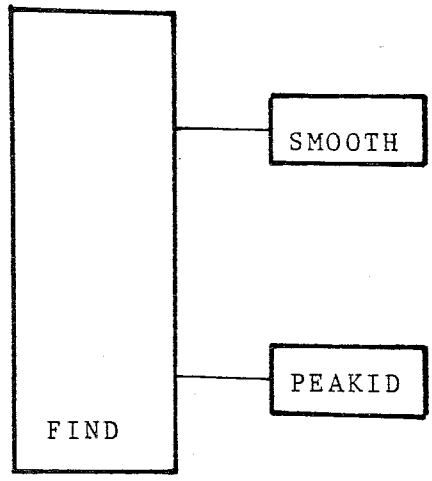

Fig.4.2. Subroutine structure of BUCAASA/FIND.

The main program controls input, output and communication with the subroutines, performs a tentative calculation of the peak areas by adding counts in the channels between the right and left limits of the peak and substracts background which is computed by fitting a line through the right and left channels. A significance value for each peak is computed as

$$S = \frac{(A+2B)^{1/2}}{A} \times 100 \qquad (4.1)$$

where A is the peak area and B is the background. Assuming that the peak can be represented by a Gaussian its area is given by

$$A_G = (2\pi)^{1/2}\sigma A \qquad (4.2)$$

Therefore an initial guess for can be computed as

$$\sigma = \frac{A_G}{(2\pi)^{1/2} A} \qquad (4.3)$$

Subroutine SMOOTH performs smoothing by the method of Savitzky and Golay (see subsection 3.1). Three, five, seven nine and eleven point smoothing is possible. Using the smoothed data the first derivative spectrum is calculated.

Subroutine PEAKID performs the peak identification using the first derivative spectrum (subsection 3.1.2). The right and left limits of the peaks, peak channels and multiplets are identified.

4.3. Program BUCAASA/FIT

BUCAASA/FIT consists of one main program, five subroutines and one function.

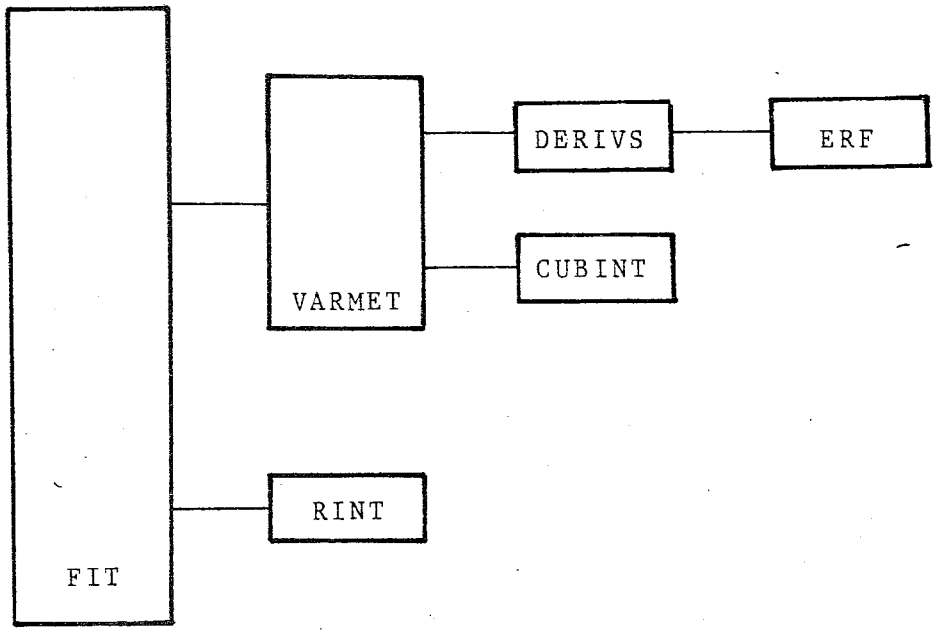

Fig.4.3. Subroutine structure of BUCAASA/FIT

Main program FIT controls input, output and performs the error estimation calculations (Appendix B). By assigning a value to the parameter IVAR the kind of fitting function can be chosen, the options being Gaussian plus polynomial background, Gaussian plus tailing and polynomial background, and Gaussian plus tailing plus the step function given by eq.(3.28) the necessary parameters are either supplied by the user or default and the values calculated by BUCAASA/FIT are used.

Subroutine VARMET is an implementation of the variable metric minimization method (Appendix A) for non linear least squares fitting. The iterations terminate when either 100 iterations have been completed or five consecutive values of

all parameters are found to be the same or all of the gradients are less than 10^{-11} or when one of the photopeak parameters become negative.

Subroutine DERIVS calculates the derivatives of the fitting function with respect to the parameters of the fit which are given by eqs.(A.35-43).

Subroutine CUBINT performs the cubic interpolation in order to calculate the optimum step length in the minimization (see Appendix A).

Subroutine RINT calculates the terms of the error equation for the tailing area (see Appendix B).

Subrautine ERF is used to compute the error function given by eq.(3.19).

Function ETS is used for to compensate underflow effects in the calculation of exponential functions.

4.4. Program BUCAASA/CALIBR

BUCAASA/CALIBR consists of one main program and four subroutines

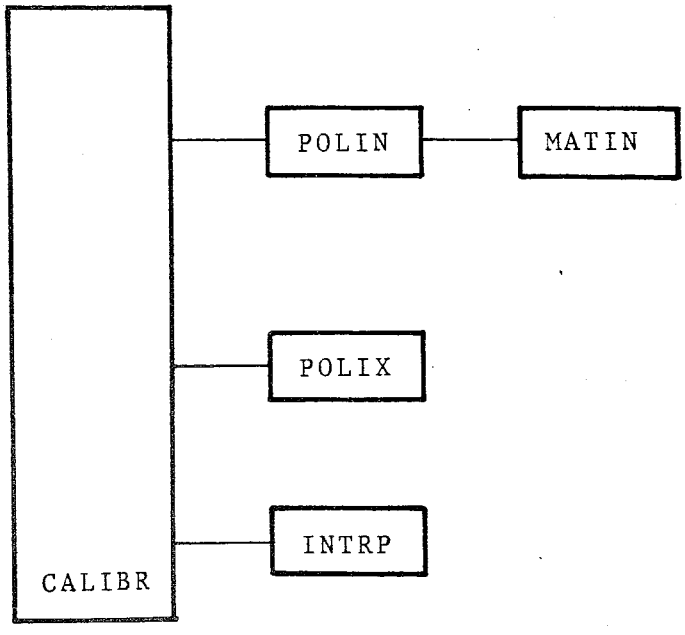

Fig.4.4. Subroutine structure of BUCAASA/CALIBR.

Main program CALIBR controls input, output and communication with the subroutines. Energy and efficiency calibration data are supplied by the user.

Subroutine POLIN performs the linear least squares fitting (see subsection 3.3).

Subroutine MATIN forms the inverse of the matrix whose elements are the coefficients of the resultant simultaneous linear equations from linear least squares fitting.

Subroutine POLIX calculates the energy and efficiency values using the coefficients determined by subroutine POLIN.

Subroutine INTRP interpolates logarithmically between efficiency calibration data points. This routine must be used only in cases where eq.(3.35) is not valid, i.e., for gamma energies less than 100 keV because an accurate error estimation is nearly impossible.

4.5. Program BUCAASA/NUCLIDE

BUCAASA/NUCLIDE Consists of one main program and eight subroutines

The main program controls only input output.

Subroutine IDENT performs the nuclide identification using the method of associated lines (subsection 3.4.2).

Subroutine LIBRAR reads in the gamma reference library for the identification routines.

Subroutine MATRIX forms the working matrix in the nuclide identification.

Subroutine CORR makes the time corrections to the activities calculated at the identification stage.

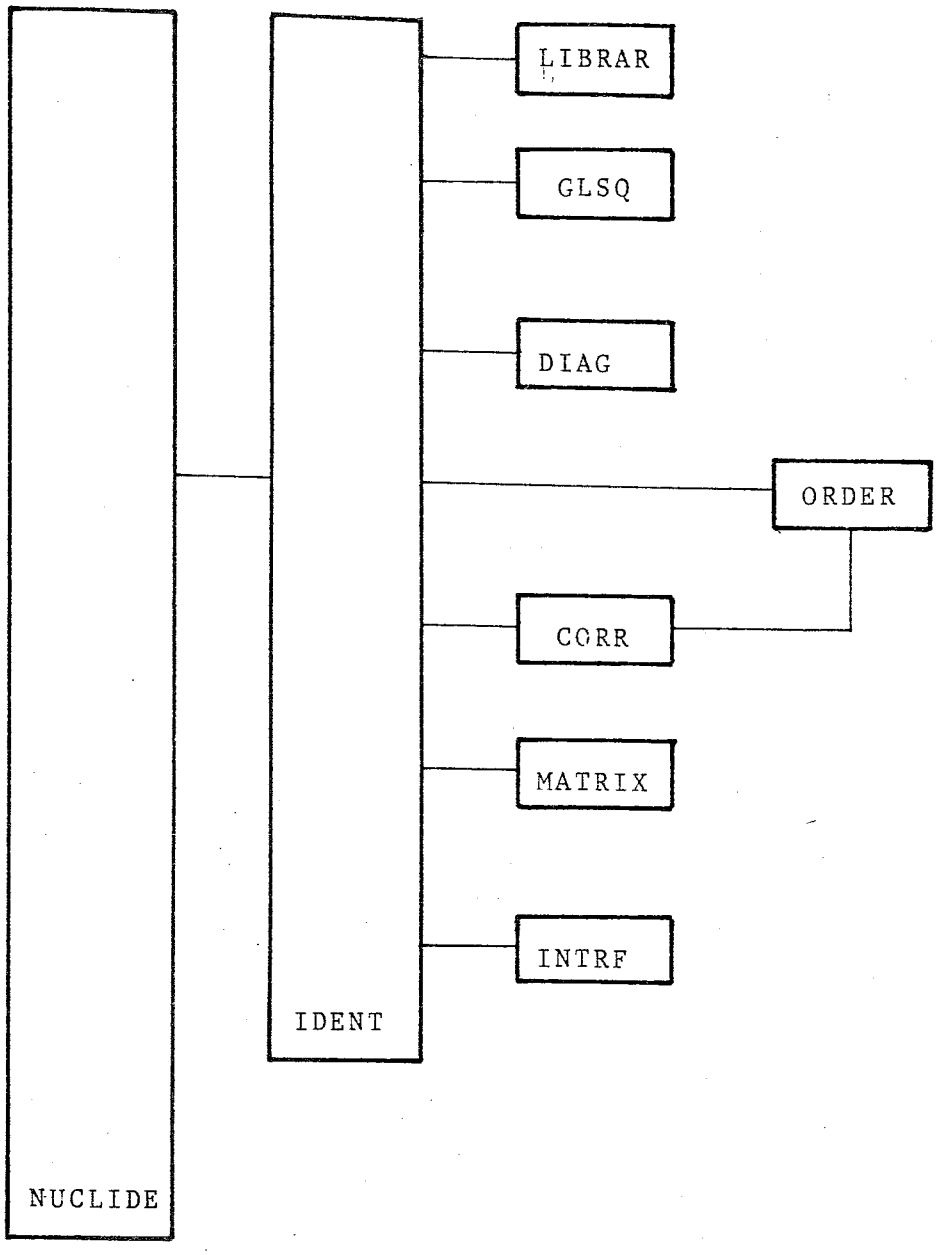

Fig.4.5. Subroutine structure of BUCAASA/NUCLIDE

Subroutine INTRF determines the interfering nuclides in the nuclide identification.

Subroutine GLSQ is a general linear least squares routine.

Subroutine DIAG determines the diagonal values of the inverse of the matrix produced from the GLSQ.

Subroutine ORDER arranges a number of arrays into the order of the ascending values of the first array.

5. DISCUSSIONS AND CONCLUSIONS

In 1976 as part of its Analytical Quality Control Services programme, the IAEA offered an intercomparison dealing with the evaluation of methods for processing Ge(Li) gamma-ray spectra[24]. 212 sets of results from 163 laboratories in 34 Member States of the IAEA were received. The environment of 207 results was computers. From the evaluation of these results it was evident that no one particular method was overwhelmingly superior.

The information on methodology provided by the participants showed that many of them were using the same computer programs for spectrum evaluation. Four programs in particular were found to be relatively frequently used, namely SAMPO by 23 participants, GASPAN by 9 participants, GAMANAL by 8 participants and PHILIPOT by 7 participants. An intercomparison of these four programs showed that SAMPO was the best. However it was also found that success in evaluating these spectra was not so much dependent on the principle of the method used but more on the details of how this method was applied.

The major aim in preparing this package was supplying the analyst with a powerful and easy to use tool whose performance was not dependent on the user.

However, even from the beginning, i.e, peak identification routines, it was found that since gamma ray detection is a statistical phenomenon the criteria for identification were somewhat subjective; no method rejected the spurious peaks while identifying all the real ones however weak they may be and whatever their interferences are. Experienced human eye is still the best tool in identifying especially the multiple peaks.

In the fitting process, all the functions which are supposed to represent the experimental data as accurately as possible were determined to be valid only under special

experimental conditions. The functions are not easy to handle since nearly in every care they are non linear, and the non linear least squares methods used in fitting are always iterative processes, requiring initial guesses for the parameters in the fit. Optimization of those parameters requires tedious calibration runs until they are reasonably well determined for the system being used.

Taking into consideration the above stated facts the programs were designed to allow the user to chose the degree of thoroughness of the analysis he desires. A compromise was found to be necessary between accuracy and automatic use of the program.

In BUCAASA/FIND four parameters direct the peak identification process, by lowering their values, it is possible to detect even very weak peaks, however, the user can choose among them the peaks he considers to be significant for his work by adjusting the parameter PTEST. These are all subjective criteria but it must be stated that they are necessary inorder to analyze exceptional spectra, e.g, peaks on high background, multiplets with a channel difference of less than three channels, etc. For routine work and well behaved spectra, i.e, spectra with low background and well defined peaks, they can be taken as constants.

In BUCAASA/FIT it is possible to use three different fitting functions all of them being a subset of eq. (3.28). Using eq. (3.28) in fitting is a tedious process since the initial guesses for the parameters have not been optimized due to lack of sufficient data. The main reason being our not being able to interface the counting system with the computer.

However, eq. (3.28) is one of the best functions representing the spectral data, for it was derived taking into consideration the physical events occuring through out the detection process. This is evident from the χ^2 parameters (see Table 5.3).

TABLE 5.1. Comparison of the results of the peak identification routines of BUCAASA and SAMPO80[30] for the IAEA G-2 program source no 2 spectrum[30]

Peak	Peak Channel		Notes
	BUCAASA/FIND	SAMPO80	
1	97.895	97.563	Multiplet
2	106.343	106.670	Multiplet
3	161.571	161.627	
4	246.203	246.201	
5	370.628	370.661	
6	414.249	414.223	
7	837.380	837.376	
8	917.378	917.376	
9	1078.138	1078.129	
10	1162.312	1162.301	
11	1546.830	1546.852	Annihilation peak
12	2526.858	2526.842	
13	3376.816	3376.819	

TABLE 5.2. Comparison of the results of peak fitting routines of BUCAASA* and SAMPO[30] for the IAEA G-2 program source no 2 spectrum[30]

Peak	Peak Channel		Peak Area		χ^2	
	BUCAASA	SAMPO80	BUCAASA	SAMPO80	BUCAASA	SAMPO80
5	370.658	370.661	5705474	5381146	425.9	1080.0
7	837.415	837.376	343154	352240	13.5	94.2
8	917.398	917.376	812008	813671	24.5	305.0
9	1078.145	1078.129	2306940	2317276	194.4	1760.0
10	1162.313	1162.301	308453	309687	12.7	194
11	1546.815	1546.852	156717	103390	21.1	6550
12	2526.870	2526.842	1637565	1633791	34.7	747
13	3376.860	3376.819	1225050	1222545	52.8	885

* Gaussian + Tailing + Parabolic compton continuum approximation is used.

In a non linear least squares fit the number of parameters that are used in the minimization must be kept few because sometimes the other parameters tend to compensate for the divergence of a parameter from its optimum value giving a low χ^2 value but leading to erroneous results. Considering this fact it is better not to use the step function since it does not give a considerable improvement (see Table 5.3). The parameter C which defines the shifting of the centroid of the tailing function is now thought to be constant for a given spectrometer system, however, again due to lack of sufficient data from different spectrometer systems and under different experimental conditions this points is still controversial and needs elaboration. Nevertheless it is evident from previous experience[3] that its values is around 1.

For comparison purposes the gamma spectrum of the IAEA G-2 program source no 2 obtained by Koskelo et al[30] was analyzed. This data has several interesting features, i.e, it contains multiplets with intensity ratios less than 1:2 and all the peaks are situated on a very high background, thus providing an excellent challenge for any spectrum analysis program. The results of SAMP080[30] are compared with BUCASA in Table 5.1.

It is evident that both programs are equally powerful. A later visual inspection showed that peaks no 3 and 4 were actually doublets which was confirmed by fitting. Both programs, BUCAASA and SAMP080[30] overlooked this fact.

As a fitting function SAMP080 uses a Gaussian with exponential tails on a quadratic background and the minimizaton of χ^2 is done by a linear least squares routine keeping the peak channels constant[30] whereas in BUCAASA. The variable metric minimization is used and the peak channels are allowed to vary. Although this sometimes causes the iteration to diverge, e.g, in the case of too far off initial parameters using e.q. (3.28) nearly in every case, better fits are obtained even with a pure Gaussian on a parabolic background (see Tables 5.2 and 5.3).

TABLE 5.3. Comparison of different fitting functions used by BUCAASA. Data obtained from the analysis of IAEA G2 program source no 2 spectrum[30]

Peak	Peak Channel			Peak Area			χ^2			χ^2_ν		
	F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3
5	370.580	370.658	370.658	5253962	5705474	5950737	40975	425.9	5761.3	128.0	15.8	221.5
7	837.372	837.415	837.405	339917	343154	348099	42.9	13.5	18.9	3.0	1.9	3.2
8	917.364	917.398	917.376	792374	812008	806443	220.0	24.5	23.4	12.2	2.7	2.9
9	1078.109	1078.145	1078.125	2299111	2306940	2369293	2510.9	194.4	496.0	81.0	10.2	27.6
10	1162.292	1162.313	1162.315	302281	308453	311934	52.1	12.7	22.0	2.6	1.6	3.1
11	1546.809	1546.815	1546.823	149646	156717	165783	37.4	21.1	41.6	1.7	1.5	3.2
12	2526.837	2526.870	2526.844	1620984	1637565	1620374	1399.5	34.7	678.0	40.0	2.0	42.3
13	3376.782	3376.860	3376.857	1212251	1225050	1226269	1821.4	52.8	54.9	49.0	2.4	2.6

F1: Gaussian + Parabolic compton continuum

F2: Gaussian + Tailing + Parabolic compton continuum

F3: Gaussian + Tailing + Step + Parabolic compton continuum

For energy calibration in BUCAASA/CALIBR a linear least squares fit of a polynomial to the calibration data is used. Polynomials of order one and two give satisfactory results[3] depending on the linearity of the spectrometer system. Again this provides estimates of the errors of the energy calibration. In SAMP080 the errors are calculated by interpolating between the errors of each calibration point, however this can only give an estimate of the systematic error but not of the errors corresponding to the calibration process itself which is in SAMP080 again an interpolation between calibration points. Nevertheless, considering that such errors are often less than the systematic error they can be neglected in routine work.

In nuclide identification an adapted version of SAMPOID is used which still seems to be the best code available.

The program package as it stands now is easy to implement on other main frame computers which have FORTRAN 77 compilers, however implementation on microcomputers requires overlaying. A user after a few runs will find it easy to use the package, but a thorough analysis using eq. (3.28) as the fitting function will present difficulties such as divergences but if time allows these problems can be overcome. Since after a few runs correct initial values for the parameters can be determined by analysing the gradients. It must be stated that most of the divergences occur as a result of hidden peaks which have been not identified by the peak search routines. In such a case careful visual inspection of the spectrum will yield the cause of the divergence. It must be stated that a blind faith in any computer program for gamma spectrum analysis will nearly in every case lead to erroneous results, because every criteria used in the analysis have a limited validity determined by the actual conditions of the analysis. Therefore computational programs are not sufficient, for a really accurate, precise and sensitive analysis and artificial intelligence techniques with pattern recognition capabilities simulating an experienced worker must be introduced into the field.

APPENDIX-A. THE VARIABLE METRIC MINIMIZATION METHOD

In this analysis, it is assumed that part of a spectrum can be represented by a function $F(x_i, \underline{p})$. x_i are the channel numbers and $\underline{p}$ is the parameter vector. The fitting is based on the determination of parameters p_k through the least squares method such that sum of the squares of the deviation of $F(x_i, p_k)$ function from the data is a minimum.

$$R^2 = \Sigma w_i \left[y_i - F(x_i, \underline{p})\right]^2 \tag{A.1}$$

gives the weighted sum of the squares of the deviation thus

$$\frac{\partial R^2}{\partial p_k} = 0 \tag{A.2}$$

must be satisfied.

R^2 is a measure of the goodness of fit. If in eq. (A.1) $w_i = 1/\sigma_i^2$ is considered where σ_i^2 is the variance of the datum at channel x_i then the statistical variation of R^2 is described by the Chi-Square distribution. For this reason eq. (A.1) can be rewritten as

$$\chi^2(x, \underline{p}) = \sum_i \frac{1}{\sigma_i^2} \left[y_i - F(x_i, \underline{p})\right]^2 \tag{A.3}$$

where χ^2 is the independent variable of the Chi-Square distribution. And thus eq. (A.2) is reduced to

$$\frac{\partial \chi^2}{\partial p_k} = -2 \Sigma_i \frac{1}{\sigma_i^2} \left[\bar{y}_i, F(x_i, \underline{p})\right] \frac{\partial F(x_i, \underline{p})}{\partial p_k} = 0 \tag{A.4}$$

and from eq. (A.2)

$$\sum_i \frac{1}{\sigma_i^2} y_i \frac{\partial F(x_i, \underline{p})}{\partial p_k} = \sum_i \frac{1}{\sigma_i^2} F(x_i, \underline{p}) \frac{\partial F(x_i, \underline{p})}{\partial p_k} \tag{A.5}$$

Considering the Taylor expansion of a general function $f(x,\underline{p})$ neglecting the third and the higher order terms

$$f(x,\underline{p}+\Delta\underline{p}) = f(x,\underline{p}) + \sum_j \frac{\partial f(x,\underline{p})}{\partial p_j}\Delta p_j + \frac{1}{2}\sum_j\sum_k \frac{\sigma^2 f(x,\underline{p})}{\sigma p_j \sigma p_k}\Delta p_j \Delta_k \qquad (A.6)$$

or in short

$$\Delta f(x,\underline{p}) = f(x,\underline{p}+\Delta\underline{p}) - f(x,\underline{p}) = \sum_j a_j \Delta p_j + \frac{1}{2}\sum_{jk} G_{jk}\Delta p_j \Delta p_k \qquad (A.7)$$

can be written. Here $\underline{p}$ is the parameter vector. Taking the derivatives with respect to Δp_j

$$\underline{g} = \underline{a} + G\ \Delta\underline{p} \qquad (A.8)$$

is obtained. Here $\underline{g}$, is the gradient vector at the point defined by the vector $\underline{p}$ in the parameter space, $\underline{a}$ is the derivative vector at the point of Taylor expansion and $\underline{\underline{G}}$ is the Hessian matrix whose elements are defined by

$$G_{jk} = \frac{\partial^2 f(x,\underline{p})}{\partial p_j \partial p_k} \qquad (A.9)$$

Multiplying both sides of eq. (A.7) with $-\underline{\underline{G}}^{-1}$

$$-\underline{\underline{G}}^{-1}\underline{g} = -\underline{\underline{G}}^{-1}\underline{a} - \Delta\underline{p} \qquad (A.10)$$

can be written.

Assuming that $\underline{\underline{G}}$ is constant around the minimum and considering $\underline{g} = 0$ at the minimum

$$\underline{a} + \underline{\underline{G}}\ \Delta\underline{p}_o = 0 \qquad (A.11)$$

is obtained from eq. (A.8).

Here $\Delta\underline{p}_o$ is the increment in$\Delta\underline{p}$ necessary to reach the

minimum. Multiplying both sides of eq. (A.10) with $\underline{\underline{G}}^{-1}$ and substituting the values obtained, namely $\underline{\underline{G}}^{-1}\underline{a}$ in eq. (A.9)

$$\Delta\underline{p}_o - \Delta\underline{p} = -\underline{\underline{G}}^{-1}\underline{g} \tag{A.12}$$

is found.

Since the Hessian matrix is in general not constant $\Delta\underline{p}_o$ cannot be readily calculated from eq. (A.11). For this reason an arbitrary positive definite and symmetrical matrix $\underline{\underline{H}}$ is used. This matrix is used in eq. (A.11) instead of $\underline{\underline{G}}^{-1}$ and $\underline{\underline{G}}^{-1}$ is formed with iteration inorder to satisfy this equation.

The step vector $\Delta\underline{s}_i$ belonging to i^{th} iteration from (eq. A.11) is

$$\Delta\underline{s}_i = -\underline{\underline{H}}_i\underline{g}_i \tag{A.13}$$

At the end of each iteration $\underline{\underline{H}}_i$ is changed suitably while maintaining its positive-definiteness. Since $\underline{g}_i$ is in the direction of steepest ascent and H_i is positive definite and

$$\Delta\underline{s}_i^T\underline{g}_i = \underline{g}_i^T\underline{\underline{H}}_i\underline{g}_i \tag{A.14}$$

is always positive definite, $\Delta\underline{s}_i$ will be in the direction of the minimum. Taking $\underline{\underline{H}}$ as the identity matrix in the initial step forces the initial step to be in the direction of the steepest descent.

Iteration process can be described as follows:

Let us assume that at the i^{th} step the point in the parameter space is represented by $\underline{p}_i$ the gradient vector at this point is g_i and $\underline{\underline{H}}$ is $\underline{\underline{H}}_i$.

(a) As the function approaches the minimum on the path $\underline{p}_i + \lambda_i\Delta s_i$ where $\Delta\underline{s}_i = -\underline{\underline{H}}_i\underline{g}_i$ a value α_i is calculated such that $f(x,\underline{p}_i + \lambda_i\Delta\underline{s}_i)$ with respect to λ is a minimum. For

$\lambda_i = \alpha_i f(x, \underline{p}_i + \alpha_i \Delta \underline{s}_i)$ is a minimum. Since the direction of $\Delta \underline{s}_i$ is towards the minimum α_i is always positive.

(b) Defining

$$\underline{\sigma}_i = \alpha_i \Delta \underline{s}_i \tag{A.15}$$

the new values of the parameters

$$\underline{p}_{i+1} = \underline{p}_i + \underline{\sigma}_i \tag{A.16}$$

are calculated.

(c) Using $\underline{p}_{i+1}$, $f(x, \underline{p}_{i+1})$ and $\underline{g}_{i+1}$ are calculated

(d) Taking

$$\underline{Y}_i = \underline{g}_{i+1} - \underline{g}_i \tag{A.17}$$

$$\underline{A}_i = \frac{\underline{\sigma}_i \underline{\sigma}_i^T}{\underline{\sigma}_i^T \underline{Y}_i} \tag{A.18}$$

$$\underline{B}_i = \frac{\underline{H}_i \underline{Y}_i \underline{Y}_i^T \underline{H}_i}{\underline{Y}_i^T \underline{H} \underline{Y}_i} \tag{A.19}$$

The H matrix is changed as

$$\underline{H}_{i+1} = H_i + \underline{A}_i + \underline{B}_i \tag{A.20}$$

where $\underline{H}_{i+1}$ is still positive definite[3].

(e) Making i = i+1, the steps a through d are repeated until the minimum is reached.

The minimization is terminated if all the components in the next step change by less than 10^{-11}, or if five succeeding values of χ^2 are the same, or if 100 iterations have been completed.

The variable metric minimization is carried in program BUCAASA/FIT by subroutine VARMET.

The value of $\lambda_{min} = \alpha_i$ in step (a) is calculated through cubic interpolation:

Assuming that the variation of $f_i(x,\lambda)$ is a cubic function

$$f_i(x,\lambda) = a\lambda^3 + b\lambda^2 + c\lambda + d \tag{A.21}$$

can be written here a, b, c, and d are constants.

$$\frac{\partial f_i(x,\lambda)}{\partial \lambda} = 3a\lambda^2 + 2b\lambda + c \tag{A.22}$$

taking $\partial f_i/\partial\lambda = 0$

$$\lambda_{min} = \alpha_i = \frac{-b \pm \sqrt{b^2 - 3ac}}{3a} \tag{A.23}$$

and

$$\left.\frac{f_i(x,\lambda)}{\partial \lambda}\right|_{\lambda = 0} = c = g_o \tag{A.24}$$

$$\left.\frac{\partial f_i(x,\lambda)}{\partial \lambda}\right|_{\lambda = \lambda_k} = 3a\lambda_k^2 + 2b\lambda_k + c = g_1 \tag{A.25}$$

is obtained.

Here g_o and g_1 are the derivatives at M and N in fig A.1. If h_o and h_1 are the values of $f(x_i,\lambda)$ at points M and N respectively.

$$f_i(x,0) = d = h_o \tag{A.26}$$

$$f_i(x,\lambda_k) = a\lambda_k^3 + b\lambda_k^2 + c\lambda_k + d = h_1 \tag{A.27}$$

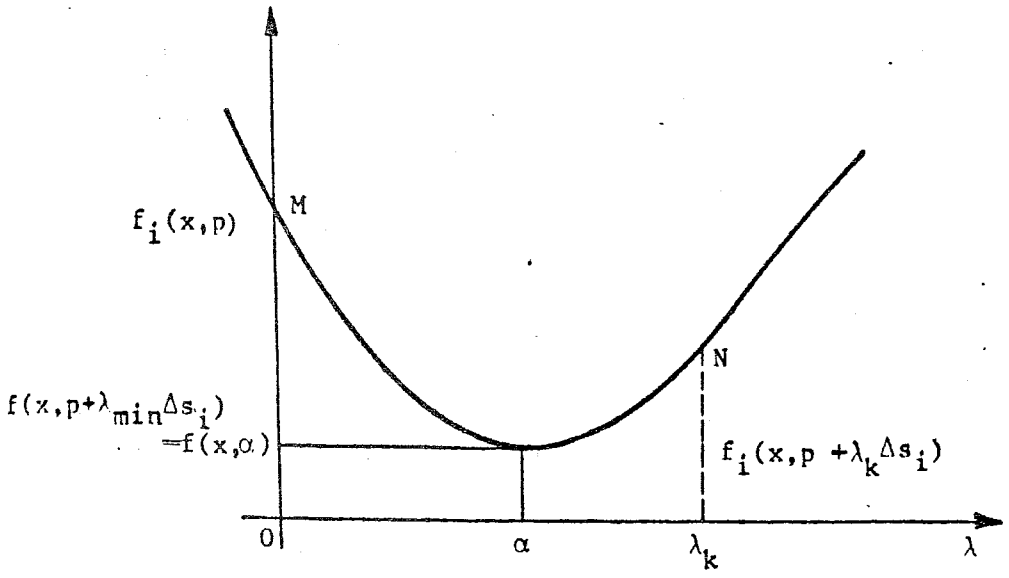

Fig. A.1. The schematic representation of $f_i(x,p + \lambda\Delta s_i)$ for the cubic interpolation.

can be written.

From eqs. (A.23), (A.24), (A.25) and (A.26)

$$a = \frac{1}{\lambda_k^2}\left[\frac{2(h_o-h_1)}{k} + g_o + g_1\right] \tag{A.28}$$

$$b = \frac{3(h_1-h_o)}{\lambda_k^2} - \frac{2g_o + g_1}{\lambda_k} \tag{A.29}$$

can be found.

Taking

$$Z = 3\,\frac{h_o-h_1}{\lambda_k} + g_o + g_1 \tag{A.30}$$

and

$$W = (Z^2 - g_o g_1)^{1/2} \tag{A.31}$$

$$\alpha = \alpha_k = \lambda_k \left(1 - \frac{Z + g_1 - w}{g_o + g_1 + 2Z}\right) \tag{A.32}$$

is obtained.

The cubic interpolation is implemented in BUCAASA/FIT under the subroutine name CUBINT.

Defining

$$F_D(x,\underline{p}) = A_t \exp\left[B(x-x_o)\right] \{1-\exp\left[-\frac{1}{2}\left(\frac{x-Cx_o}{t}\right)^2\right]\} \delta \tag{A.33}$$

from

$$-\frac{\partial \chi^2}{\partial p_k} = 2\sum_i \{\frac{1}{\sigma_i^2}\left[(y_i - F(x_i,\underline{p})\right] \frac{\partial F(x_i,\underline{p})}{\partial p_k}\} \tag{A.34}$$

$F(x_i,\underline{p})$ being given by eq. (3.28), the derivatives are readily calculated as

$$\frac{\partial F(x,\underline{p})}{\partial a_k} = x^{k-1} \tag{A.35}$$

$$\frac{\partial F(x,\underline{p})}{\partial h} = \text{erfc}\left(\frac{x-x_o}{\sigma_g}\right) \tag{A.36}$$

$$\frac{\partial F(x,\underline{p})}{\partial A_t} = \frac{F_D(x,\underline{p})}{A_t} \tag{A.37}$$

$$\frac{\partial F(x,\underline{p})}{\partial A_g} = \exp\left[-\frac{1}{2}\left(\frac{x-x_o}{\sigma_g}\right)^2\right] \tag{A.38}$$

$$\frac{\partial F(x,\underline{p})}{\partial x_o} = \left[\frac{h}{\sigma_g\sqrt{2\pi}} + A_g \frac{x-x_o}{\sigma_g^2}\right] \exp\left[-\frac{1}{2}\left(\frac{x-x_o}{\sigma_g}\right)^2\right] -$$

$$- \{BF_D(x,\underline{p}) - \frac{C}{\sigma_t^2}(x-Cx_o)\left[F_D(x,\underline{p}) - A_t \exp(B(x-x_o))\right]\} \delta \quad (A.39)$$

$$\frac{\partial F(x,\underline{p})}{\partial B} = F_D(x,\underline{p})\ (x-x_o) \quad (A.40)$$

$$\frac{\partial F(x,\underline{p})}{\partial C} = \frac{x_o}{\sigma_t}\ (x-Cx_o)\{F_D(x,\underline{p}) - A_t \exp\left[B(x-x_o)\right]\}\ \delta \quad (A.41)$$

$$\frac{\partial F(x,\underline{p})}{\partial \bar{\sigma}_t} = \frac{(x-Cx_o)}{\sigma_t^3}\ \{F_D(x,\underline{p}) - A_t\left[\exp\ B(x-x_o)\right]\}\ \delta \quad (A.42)$$

$$\frac{\partial F(x,\underline{p})}{\partial \sigma_g} = \left[\frac{h}{\sigma_g^2\sqrt{2\pi}}\ (x-x_o) + A_g\ \frac{(x-x_o)}{\sigma_g^3}\right] \exp\left[-\frac{1}{2}\left(\frac{x-x_o}{\sigma_g}\right)^2\right] \quad (A.43)$$

Where the index j has been omitted for clarity.

APPENDIX-B. ERROR ESTIMATION IN FITTING

B.1. Error Estimates of the Parameters in Fitting

A function $F(x_i,\underline{p})$ which is linear with respect to the parameters given by the vector $\underline{p}$ can be written as

$$F(x_i,\underline{p}) = \sum_j p_j P_j(x_i) \tag{B.1}$$

where $P_j(x_i)$ are any functions of the independent variables x_i.

The variances in determining the parameters p_j are given by the standard formulation as

$$\sigma^2_{pj} = \sum_i \sum_{i'} \sigma_{ii'} \frac{\partial p_j}{\partial y_i} \frac{\partial p_j}{\partial y_{i'}}, \tag{B.2}$$

here, $\sigma_{ii'}$, in general represents the variances and covariances of the data points and for $i = i'$ variances are given by

$$\sigma^2_i = \sigma_{ii'} \tag{B.3}$$

In eq. (B.2) y_i and $y_{i'}$ represent the data points. Since in optimizing the parameters of the photopeak function $F(x,\underline{p})$, the data points are used, the indeterminancies of the parameters will be a function of the standard deviations of every data point.

Since in this study it is assumed that a high resolution spectrometer system (Ge(Li) based) is used, consequently it is assumed that there is no correlation among the data points. For this reason from eq. (B.2) and (B.3)

$$\sigma^2_{pj} = \sum_i \left[\sigma^2_i \left(\frac{\partial p_j}{\partial y_i}\right)^2\right] \tag{B.4}$$

can be written.

From eq. (B.1) the derivatives of $F(x_i, \underline{p})$ with respect to p_j can be readily found as

$$\frac{\partial F(x_i, \underline{p})}{\partial p_j} = P_j(x_i) \tag{B.5}$$

By substituting eq. (B.5) in eq. (A.5)

$$\sum_i \frac{1}{\sigma_i^2} y_i P_k(x_i) = \sum_i \frac{1}{\sigma_i^2} P_k(x_i) \sum_j p_j p_j(x_l) \tag{B.6}$$

is obtained. Defining

$$\beta_k = \sum_i \frac{1}{\sigma_i^2} y_i P_k(x_i) \tag{B.7}$$

and

$$\alpha_{kj} = \sum_i \frac{1}{\sigma_i^2} P_k(x_i) P_j(x_i) = \frac{1}{2} \frac{\partial^2 \chi^2(x, \underline{p})}{\partial p_k \partial p_j} \tag{B.8}$$

eq. (B.6) can be written as

$$\underline{\beta} = \underline{\alpha}\ \underline{p} \tag{B.9}$$

or

$$\underline{p} = \underline{\alpha}^{-1}\ \underline{\beta} \tag{B.10}$$

and by defining

$$\varepsilon = \alpha^{-1} \tag{B.11}$$

$$\underline{p} = \underline{\varepsilon}\ \underline{\beta} \tag{B.12}$$

or

$$p_j = \sum_k \varepsilon_{jk}\beta_k = \sum_k \varepsilon_{jk} \sum_i \frac{1}{\sigma_i^2} y_i P_k(x_i) \qquad (B.13)$$

Thus for the derivative in eq. (B.4)

$$\frac{\partial p_j}{\partial y_i} = \sum_k \varepsilon_{jk} \frac{1}{\sigma_i^2} p_k(x_i) \qquad (B.14)$$

can be written.

And eq. (B.4) is transformed to

$$\sigma_{pj}^2 = \sum_i \sigma_i^2 \left(\frac{\partial p_j}{\partial y_i}\right)^2 = \sum_k \sum_\ell \{\varepsilon_{jk}\varepsilon_{j\ell} \sum_i \left[\frac{1}{\sigma_i^2} p_k(x_i) p(x_i)\right]\}$$

$$= \sum_k \sum_\ell \varepsilon_{jk} \varepsilon_{j\ell} \alpha_{k\ell} = \varepsilon_{jj}$$

Thus ε can be defined as the Error Matrix.

Similarly for the covariances

$$\sigma_{pj}\sigma_{pk} = \varepsilon_{jk} \qquad (B.16)$$

is obtained.

Using eq. (B.15) the indeterminancies in the parameters calculated with the linear least squares method can be estimated. If we consider the parabolic expansion of $\chi^2(x,\underline{p})$, it can be shown that $\Delta p_k = (\varepsilon_{kk})^{1/2} = \sigma p_k$ in one of the parameters causes to increase $\chi^2(x,\underline{p})$ by 1 or inother words,

$$\chi^2_{\nu+1}(x,p_1,p_2,\ldots,p_k+(\varepsilon_{kk})^{1/2},\ldots,p_n) = \chi^2(x,p_1,p_2,\ldots,p_k,\ldots,p_n) + 1 \qquad (B.17)$$

Here ν represents the degree of freedom. If we consider

$\nu = n-m$ here n is the number of data points and m is the number of parameters found by the optimization process. The mean value of χ^2 the independent variable of the Chi-Square distribution is equal to ν. And an increase of 1 in the degrees of freedom forces the mean value of χ^2 to increase by 1. Since eq. (B.17) satisfies this condition the definitions in eq. (B.15) and (B.16) can be used for a non linear least squares optimization process.

In the iteration process in the variable metric minimisation process the matrix $\underline{H}_i$ approaches to $\underline{H}_m$ the inverse Hessian Matrix which is by definition the second derivative matrix.

$$\underline{H}_m = \left|\left| \frac{\partial^2 F(x,\underline{p})}{\partial p_j \partial p_k} \right|\right|^{-1} = \underline{V} \qquad \text{(B.18)}$$

Using eqs. (B.8), (B.11) and (B.18)

$$\underline{\varepsilon} = \alpha^{-1} = 2\,\underline{V} \qquad \text{(B.19)}$$

can be written. An exact analysis of the statistical errors in a non linear function is nearly impossible since function used represents the data only approximately. Therefore eq. (B.19) has a limited validity since it does not take into consideration the deviations from the best function to represent the data.

In the situations where the function is approximate χ^2 is generally greater than one and the deviation is greater than that given by eq. (B.15) and is a function of χ^2. Considering this fact instead of eq. (B.19)

$$\varepsilon \cong \chi^2_\nu\, \underline{V} = \underline{V} \qquad \text{(B.20)}$$

can be used as an approximation. Here $\underline{V}$ is generally called the variance-covariance matrix.

The standard deviation of every parameter σ_{pj} is given by

$$\sigma_{pj} = (v_{jj})^{1/2} = (\chi^2_\nu V_{jj})^{1/2} \qquad (B.21)$$

and the overall standard deviation of function $f(\underline{p})$ is given by

$$\sigma_f = (\sum_j \sum_k \frac{\partial f(\underline{p})}{\partial p_j} \frac{\partial f(\underline{p})}{\partial p_k} V_{jk} \chi^2_\nu)^{1/2} \qquad (B.22)$$

B.2. Error Estimates for Areas

The standard deviations of the areas given by eq. (3.30) and (3.31) are calculated as follows:

In general the standard deviation of a function

$$y = f(u,v,\ldots) \qquad (B.23)$$

is given by

$$\sigma_y^2 = \sigma_u^2 (\frac{\partial y}{\partial u})^2 + \sigma_v^2 (\frac{\partial y}{\partial v})^2 + 2\, \sigma_{uv} (\frac{\partial y}{\partial u})(\frac{\partial y}{\partial v}) + \ldots \qquad (B.24)$$

Where σ_u^2, σ_v^2,..., are the variances and σ_{uv} are the covariances.

The standard deviation of the area of the Gaussian function given by eq. (3.30) is calculated by considering eq. (B.24) as

$$\sigma^2_{A_G} = (\frac{\partial A_G}{\partial A_g})^2 \sigma^2_{A_g} + (\frac{\partial A_G}{\partial \sigma_g})^2 \sigma^2_g + 2(\frac{\partial A_G}{\partial A_g})(\frac{\partial A_G}{\partial \sigma_g}) \sigma A_G \sigma_g \qquad (B.25)$$

And in similar manner the standard deviation of the area of the tailing function can be computed from

$$\sigma^2_{A_T} = \left(\frac{\partial A_T}{\partial A_t}\right)^2 \sigma_{A_t} + \left(\frac{\partial A_T}{\partial B}\right)^2 \sigma^2_B + \left(\frac{\partial A_T}{\partial C}\right)^2 \sigma^2_c + \left(\frac{\partial A_T}{\partial x_o}\right)^2 \sigma^2_{x_o} +$$

$$+ \left(\frac{\partial A_T}{\partial \sigma_t}\right)^2 \sigma^2_{\sigma_t} \left(\frac{\partial A_T}{\partial A_t}\right)\left(\frac{\partial A_T}{\partial B}\right) \sigma_{A_t B} + 2\left(\frac{\partial A_T}{\partial A_t}\right)\left(\frac{\partial A_T}{\partial C}\right) \sigma_{A_t C} +$$

$$+ 2\left(\frac{\partial A_T}{\partial A_t}\right)\left(\frac{\partial A_T}{\partial x_o}\right) \sigma_{A_t x_o} + 2\left(\frac{\partial A_T}{\partial A_t}\right)\left(\frac{\partial A_T}{\partial \sigma_t}\right) \sigma_{A_t \sigma_t} +$$

$$+ 2\left(\frac{\partial A_T}{\partial B}\right)\left(\frac{\partial A_T}{\partial C}\right) \sigma_{BC} + 2\left(\frac{\partial A_T}{\partial B}\right)\left(\frac{\partial A_T}{\partial x_o}\right) \sigma_{Bx_o} +$$

$$+ 2\left(\frac{\partial A_T}{\partial B}\right)\left(\frac{\partial A_T}{\partial \sigma_t}\right) \sigma_{B\sigma_t} + 2\left(\frac{\partial A_T}{\partial C}\right)\left(\frac{\partial A_T}{\partial x_o}\right) \sigma_{Cx_o} +$$

$$+ 2\left(\frac{\partial A_T}{\partial C}\right)\left(\frac{\partial A_T}{\partial \sigma_t}\right) \sigma_{C\sigma_t} + 2\left(\frac{\partial A_T}{\partial x_o}\right)\left(\frac{\partial A_T}{\partial \sigma t}\right) \sigma_{x_o \sigma_t} \qquad \text{(B.26)}$$

All the variances and covariances in eqs. (B.25) and (B.26) are given by the matrix $\underline{v}$ in eq. (B.20).

The derivatives in eqs. (B.25) and (B.26) are as follows

$$\frac{\partial A_G}{\partial A_g} = (2\pi)^{1/2} \sigma_g \qquad \text{(B.27)}$$

$$\frac{\partial A_G}{\partial \sigma_g} = (2\pi)^{1/2} A_g \qquad \text{(B.28)}$$

$$\frac{\partial A_T}{\partial A_t} = \frac{1}{B} \left(\exp(-Bx_o) \left[\exp(BCx_o) - 1\right] + (2\pi)^{1/2} \sigma_t \exp\left[\frac{1}{2}B^2 \sigma_t^2 + Bx_o(C-1)\right]\right.$$

$$\left. X \left[\operatorname{erf}(B\sigma_t) - \operatorname{erf}\left(\frac{Cx_o}{\sigma_t} + B\sigma_t\right)\right]\right. \qquad \text{(B.29)}$$

$$\frac{\partial A_T}{\partial B} = -\frac{A_t}{B^2}\exp(-Bx_o)\left[\exp(B(x_o)-1\right]+A_T\left[B\sigma_t^2 + x_o(C-1)\right]$$

$$+ A_t\sigma_t^2 \exp(-Bx_o)\left[1-\exp(-\frac{1}{2}\frac{C^2x_o^2}{\sigma_t^2})\right] + A_t x_o \frac{C}{B}\exp(-Bx_o) \qquad (B.30)$$

$$\frac{\partial A_T}{\partial C} = A_t x_o \exp(-Bx_o)+A_T Bx_o - A_t x_o \exp(-Bx_o - \frac{1}{2}\frac{C^2x_o^2}{\sigma_t^2}) \qquad (B.31)$$

$$\frac{\partial A_T}{\partial \sigma_t} = (1+B^2\sigma_t^2)\frac{1}{\sigma_t}\left[A_T + \frac{A_t}{B}\exp(-Bx_o)\right] - \frac{A_t}{B\sigma_t}\exp\left[Bx_o(C-1)\right]$$

$$- A_t\sigma_t (B - \frac{Cx_o}{\sigma_t^2})\exp(-\frac{1}{2}\frac{C^2x_o^2}{\sigma_t^2} - Bx_o) \qquad (B.32)$$

$$\frac{\partial A_T}{\partial x_o} = B(C-1)A_T + CA_t \exp(-Bx_o)\left[1-\exp(-\frac{1}{2}\frac{C^2x_o^2}{\sigma_t^2})\right] \qquad (B.33)$$

In eq. (B.25-33) the index j has been omitted for clarity.

APPENDIX-C. USERS' GUIDE FOR BUCAASA

Inorder to make the programs more user friendly the number of subjective parameters supplied by the user have been kept to a minimum, however some parameters are still necessary to provide a flexibility in controlling the flow of the programs.

All the data are inputted interactively, excepting those that are read from files and there are no format restrictions for input, i.e. free format is used.

C.1. Input Data for BUCAASA/FIND

Record 1. JOB (Job heading less than 50 alphanumeric characters)

Record 2. ATGE, ATLT, FAC, ADEL, ATEST, PTEST

(The first four are peak search parameters, all of them can be taken as 1. PTEST is the significance test parameter for the peak found by the routine, it can take any value in the range 0-100).

Record 3. INIT, IFIN, ISMT, DISK, ISTOR, ISWO

(INIT: Initial channel of the spectrum to be considered in the analysis

IFIN: Last channel of the spectrum to be considered in the analysis

ISMT: Smoothing parameter

2*ISMT+3 point smoothing is performed

ISMT: 5 no smoothing

IDISK: 1 Spectrum will be input manually

0 Spectrum will be read in from a random access file

ISTOR: 1 The results will be stored in a random access file for use by BUCAASA/FIT

0 The results will not be stored

ISWO: 1 Results of smoothing will be printed out

0 Results of smootihgn not be printed

If IDISK=0

Record |4-(IFIN-INIT+1)| Spectrum data

If IDISK=1

Record 4. NAME (Less Than 8 alphanumeric characters)
The name of the random access file which contains the spectrum data. RECL must be equal to 2.

If ISTOR=1

Record 5 or (IFIN-INIT). NAME2 (less Than 8 alphanumeric characters)

The name of the random access file which will contain the results of peak identification process. It will later be used by BUCAASA/FIT.

C.2. Input Data for BUCAASA/FIT

Record 1. JOB (Job heading less than 50 alphanumeric characters)
Record 2. INUM IGUES IDISK ISAVE II IVAR IRPT ISET

INUM: The number of the peak

IGUES: 1 Initial pramaters will be read in from R.A.F. prepared by BUCAASA/FIND

0 Input manual

IDISK: 0 Spectrum data will be input manually

1 Spectrum data will be read in from a R.A.F.

ISAVE: 0 The results of the analysis will not be stored

1 The results of the analysis will be stored in a R.A.F. for use by BUCAASA/CALIBR

II Degree of the compton polynomial -1 Must be between 1 and 3.

IVAR: 1 Amplitude of the Gaussian is variable (Gaussian + Compton Fitting)

2 Amplitude + centroid are variable (Gaussian + Compton Fitting)

3 Amplitude + centroid + FWHM are variable (Gaussian + Compton Fitting)

4 Gaussian + Tailing + Compton Fitting

5 Gaussian + Tailing + Step + Compton Fitting

IRPT: 0 Do not repeat analysis

1 Repeat fitting by taking IVAR=IVAR+1 (IVAR must be less than 5)

ISET: 0 Stop execution after fitting

1 Read data for another fitting

-1 Used for sequential fitting for a whole peak set valid only if IGUES=1. In this case INUM must be equal to 2.

IF IDISK=1

Record 3. NAM2 (less than 8 alphanumeric characters) The name of the R.A.F. which contains the spectrum data. RECL must be equal to 2.

IF ISAVE=1

Record 4. NAM3 (less than 8 alphanumeric characters) The name of the R.A.F. which will contain the results of the fitting process. It will later be used by BUCAASA/CALIBR

IF IGUES=1

Record 5. NAM (less than 8 alphanumeric characters) The R.A.F. created by BUCAASA/FIND

IF IGUESS=0

Record 5. INIT, IFIN, MUL

INIT Left channel of the peak region

IFIN Right channel of the peak region

MUL: The number of peaks in the peak region (Multiplicity parameter)

Record 6. SIG, (XJ(J),J=1,MUL)

SIG: FWHM/2.335:Width parameter

XJ(J): Approximate peak channel for each peak.

Record 7. (AI(I), I=1,IP)

IP: Number of parameters in fit

AI(I): I=1,II Coefficients of the Compton polynomial

(If IVAR=3 AI(I), I=II+1, II+MUL coefficients of the step function)

AI(I), I=II+MUL+1,IP Heights of the Gaussians

For IVAR ≤ 4

AI(I), I=II+1,IP Heights of the Gaussians

IF IVAR ≥ 4

Record 8. AT, BT, CT, SIGT

Tailing function parameters

IF IDISK=0

Record |9-(IFIN-INIT+1)| Spectrum data

C.3. Input Data for BUCAASA/CALIBR

Record 1. JOB (less than 50 alphanumeric characters)

Job heading.

Record 2. NP, MP, SYSER

NP: Number of energy calibration points

MP: Order of the polynomial used for calibration

SYSER: Systematic error for energy calibration

Record |3-(3+NP)| Energy calibration data

(Peak channels and corresponding energies in keV)

Record(NP+4). LP, SYSEF, EFIN

LP: Number of efficiency calibration points

SYSEF: Systematic error for efficiency calibration

EFIN: 1 Interpolate between calibration data points

0 Fit a line

IF EFIN=1

Record(NP+5)-(NP+LP+5). Efficiency calibration data

Energies in keV, efficiencies and corresponding errors in percent

IF EFIN=0

Record(NP+5)-(NP+LP+5) Efficiency calibration data

Energies in keV and efficiencies.

Record(NP+LP+6). NAM (less than 8 alphanumeric characters) R.A.F. which contains the results of BUCAASA/FIT.

Record(NP+LP+7)NAM2 (less than 8 alphanumeric characters) The name of the sequential, formatted file which will contain the results of energy and efficiency calibration to be used by BUCAASA/NUCLIDE.

C.4. Input Data for BUCAASA/NUCLIDE .

Record 1. JOB (less than 50 alphanumeric characters) Job heading

Record 2. TCO, TEX, TWA, TLIVE, ETOL

TCO: Counting time in minutes

TEX: Irradiation time in minutes

TWA: Cooling time in minutes

TLIVE: Live time of detection in seconds

ETOL: Energy tolerance used in gamma line identification in keV

Record 3. INF (less than 8 alphanumeric characters) The name of the sequential, formatted file created by BUCAASA/CALIBR

Record 4. LIBF (less than 8 alphanumeric characters) The sequential formatted file which contains the gamma library, e.g. half lives, energies, etc.

The format of the gamma library LIBF is as follows

Record 1. String 'UNK' read into UNK

Record 2. BLNK. Any string of 8 characters which marks the begining of the library. This same string

must also mark the end of the library.

Record 3. LIB, BNR

Library name and number FORMAT(A8,2X,E10.0)

Record 4. ISO, THA, IHA, NGA, APR, ANE, CRO, THR

(A8, 2X, E 10.0, A1, 7X, I2, 4E10.0, 62X)

ISO: Nuclide's name (e.g. ZN-65)

THA: Its half life

IHA: Unit of half life |M(inute), H(our), D(ay), Y(ear)|

NGA: Number of gammalines

APR: Number of protons

ANE: Number of neutrons

CRO: Cross section for thermal neutron capture

THR: Threshold cross section

The following lines of FORMAT(8E10.0) contain pairs of peak energies (keV) and branching ratio (per cent).

APPENDIX-D. IMPLEMENTATION GUIDE FOR BUCAASA

BUCAASA as a whole is coded in ANSI FORTRAN 77 programming language. There are two exceptions to the standard.

a) The PROGRAM statement which is compulsory for the FTN5 compiler of CDC Cyber 170/815 and can be omitted in other systems, contains several file informations.

b) Non standard functions TIME and DATE can be omitted
TIME supplies the current time
DATE supplies the current date
Both of them are character functions and return a string 10 characters long.

D.1. The File Structure of BUCAASA

Since the package consists of four main programs it was found necessary to use files for data communication between consecutive programs. A description of the files is given in Tables D.1-D.5.

TABLE D.1. Files used by BUCAASA/FIND

File #	Name*	Type	Notes
5	-	Sequential	Terminal input file
6	-	Sequential	Terminal output file
11	-	Sequential	Printer file
15	NAME2	Random Access (RECL=50)	Transfer file; contains results of peak identification
16	NAME	Random Access (RECL=2)	Spectrum file

* Only user supplied names are given

TABLE D.2. Files used by BUCAASA/FIT

File ≠	Name	Type	Notes
5	-	Sequential	Terminal input file
6	-	Sequential	Terminal output file
11	-	Sequential	Printer file
15	NAM	Random Access (RECL=50)	Transfer file, contains the rsults of peak identification
16	NAM2	Random Access (RECL=2)	Spectrum file
17	NAM3	Random Access (RECL=30)	Transfer file, contains results of fitting

TABLE D.3. Files used by BUCAASA/CALIBR

File ≠	Name	Type	Notes
5	-	Sequential	Terminal input file
6	-	Sequential	Terminal output file
11	-	Sequential	Printer file
15	NAM2	Sequential Formatted	Transfer file, contains the results the results of calibration
17	NAM	Random Access (RECL=30)	Transfer file, contains results of fitting

TABLE D.4. Files used by BUCAASA/NUCLIDE

File ≠	Name	Type	Notes
5	-	Sequential	Terminal input file
6	-	Sequential	Terminal output file
11	-	Sequential	Printout file
15	INF	Sequential FMT	Transfer file, contains the results the results of calibration
17	LİBF	Sequential FMT	Gamma library file

D.2. Description of the Important Variables of the Program units of BUCAASA

D.2.1. Program FIND

NPEAK: number of peaks in the spectrum

NCHAN: Number of channels in the spectrum

A(NPEAK,10): Contains several data about each peak

A(NPEAK,I) : I=1,2,3 Coefficients of the Compton polynomial

I>4 Centroids of peaks for a peak group

ADEL Parameter for peak serach

ATEST

ATGE Parameter for peak search

ATLT

BDATA(NCHAN) Background counts for each channel

DATA(NCHAN) Spectrum

DDATA(NCHAN) First derivatice spectrum

FAC Parameter for peak search

IAREA(NPEAK) Area under peaks

IBGD(NPEAK) Area under the compton polynomial

IFIN Appendix C

INIT

ISET

ISMT

ISTOR

ISWO

JFIT(NPEAK) Multiplicity parameter for each peak

K Number of detected photopeak groups

LN Number of significant photopeak groups

MAX(NPEAK) Right channel for each peak

MIN(NPEAK) Left channel for each peak

NAME See appendix C

NAME2

PCERR(NPEAK) Significance parameter for each peak

PEAK(NPEAK) Centroid channel for each peak

SDATA(NCHAN)	Smoothed spectrum
SIG(NPEAK)	Width parameter
SM(NCHAN)	Standard deviation of counts in each channel
TDATA(NPEAK)	Background subtracted counts in each channel.

D.2.2. Subroutine SMOOTH

DATA(NCHAN)	Spectrum
DDATA(NCHAN)	First derivative spectrum
IFIN	See appendix C
INIT	
IO	2*IO+1 point smoothing is performed
ISMT	See appendix C
MCOE(5)	Normalization constants for smoothing
NDEN(5)	Normalization constants for firs derivatives
NMCOE(6,5)	Convolution coefficients for smoothing and first derivative
SDATA(NCHAN)	Smoothed spectrum
SM(NCHAN)	Standard deviation spectrum.

D.2.3. Subroutine PEAKID

ADEL	
ADL	
ATEN	
ATES	parameters for peak search
ATEST	
ATGE	
ATLT	
DATA(NCHAN)	Spectrum
DDATA(NCHAN)	First derivatice spectrum
IPKN(NPEAK	Multiplicity parameter for each peak
JFIT(NPEAK)	
JMUL(NPEAK)	
MAX(NPEAK)	Right channel of the peak
MIN(NPEAK)	Left channel of the peak

PEAK(NPEAK)	Centroid of the peak
PR(20)	Centroid of the peak (temporary storage)
SDATA(NCHAN)	Smoothed data
SM(NCHAN)	Standard deviation spectrum.

D.2.4. Program FIT

AF(15)	Parameters in fitting
AI(15)	
AG(5)	Area of Gaussians
AIC(5)	Tailing height parameter
AIX	Parameter used in compensating for step
HX	function
AT	Ag/At
BT	
CT	Parameters of the tailing function
SIGT	
EAF	Error matrix
ERRC	Error correlation parameter
F(50)	Values of the fitted function
IC	Number of parameters in fit
IDISK	
IFIN	
IGUES	
IT	
INIT	
INUM	See Appendix C
IRPT	
ISAVE	
ISET	
ISTOP	
IVAR	
NAM	
NAM2	See Appendix C
NAM3	
NP	Number of channels in fit

NPK	Number of peaks to be fitted
RD(5)	$\partial AT_j/\partial P_k$ (see Appendix B)
SIG	Peak width parameter
VARI(5)	Variance of peak area
W(50)	Weighting constants
X(50)	Channel numbers
XJ(5)	Gaussian centroid channels
Y(50)	Spectrum data

D.2.5. Subroutine VARMET

AF(15)	Parameters in fit
AI(15)	
AIC(5)	Tailing height parameter
AT	Ag/At
BT	
CT	Parameters of the tailing function
SIGT	
CHIF	χ^2_ν
CSQ1	χ^2
CSQ2	
EAF(15,15)	Error matrix
EPs	Convergence test parameter
ETAA	Step length
F(50)	Values of the fitted function
GA(15)	
GB(15)	Gradient vectors
GC(15)	
H(15,15)	Hessian matrix
IC	Number of parameters in fit
IKNT	Number of iterations
IRPT	See Appendix C
IVAR	
MUL	Multiplicity parameter for peak groups
SIG	Peak width parameter

XFA

XFB χ^2

XFD

Y(50) Spectrum

D.2.6. Subroutine DERIVS

EXPF(50) Values of the gaussian function

FMODG(50) F_D (see Appendix A)

NP Number of channels

G Gradients

D.2.7. Subroutine CUBINT

A

GAMMA

G1

G2 See Appendix A

W

X1

X2

D.2.8. Subroutine RINT

RD(5,15) $\partial AT_j/\partial P_k$ (See Appendix B)

RIN(5) Tailing area

D.2.9. Program CALIBR

ACE(50) Energy calibration data (energy values in KeV)

RCEP(50) Energy calibration data (Channel values)

AREA(150) Peak areas

EAREA(150) Errors in peak area determination

EEF(150) Efficiency errors %

EENGY(150) Calculated eneryg erros

EFIN See Appendix C

EFL2(150) Calibrated Efficiency values

ENGY(150) Calibrated energy value

EPK(150) Errors in peak channel determination

LP	See Appendix C
MP	
NP	
MUL	Multiplicity parameter
mX	Number of peaks
SYSEF	See Appendix C
SYSER	

D.2.10. Subroutine POLIN

N : Order of the fitting polynomial + 1
M L Number of calibration points
YC: The fitted values
S : Variance

D.2.11. Subroutine MATIN

A : The matrix to be inverted
B : Inverse of A

D.2.12. Subroutine POLIX

C(10) Co-efficients of the polynomial
YD(150) Value of the fitted function
YE(L50) Corresponding errors

D.2.13. Subroutine INTRP

XX(N)	Calibration data
YY(N)	
Y(M)	interpolated values
YE((M)	interpolated erros

D.2.14. Program NUCLIDE

ETOL	
INF	
LIBF	
TCO	See Appendix C
TEX	
TLIVE	
TWA	

D.2.15. Subroutine CORR

A(80,30) The associated lines matrix
ACOL2(30) Ordering numbers
ACOL5(30) Saturation activities
ASAT
COL(30) Names of the isotopes
CONF(30) Confidence valves
EFER(30) Efficiency error
EFFSUM(30) Intensity of peaks
ENER(30) Energy of peaks
ETOL Energy tolerance
FITCH(30) Peak centroid
HALF(30) Halflives of isotopes
ITUN(30) Names of the isotopes
NUMF Number of peaks
TCO
TEX
TLIVE
TWA

See Appendix C

D.2.16. Subroutine DIAG

N: Number of diagonal elements of the matrix
A(80,30) The matrix whose diagonal values of its inverse are to be determined.

D.2.17. Subroutine GLSQ

A(80,30): Coefficient matrix of simultaneous linear equations
X(30) : Interference coefficients

D.2.18. Subroutine IDENT

A(80,30) The associated lines matrix
ACOL2(30) Ordering numbers
ACOL5(30) Saturation activities
ANEU: Number of neutrons of the isotope

APRO: Number of protons of the isotope

B(80,30): Interference matrix

CONF(30) Confidence values

EREF(30) Gamma energies of the isotopes

EFER(100) Efficiency errors

EFFSUM(100) Intensity of the peaks

ENERG(100) Energy of the peaks

ETOL: Energy tolerence

FITCH(100) Centroid of the peaks

GREF(30) Branching intensities of the isotopes

HALF(30) Half lives of the isotopes

ITUNN(30) symbols of the isotopes

TCO

TEX See Appendix C

TLIVE

TWA

THAR: Half life of the isotope

X(30): Interference coefficients.

D.2.19. Subroutine INTRF

A(80,30): Associated lines matrix

NUMF: Number of peaks

D.2.20. Subroutine LIBRAR

ANE: Number of neutrons of the isotope

APR: Number of protons of the isotope

ENE(30): Energies of the gamma lines

IHA: Symbol of the unit of half life

LINES: Total number of gamma lines of the isotopes in the library

NGA: Number of gamma lines of an isotope in the library

NDISO: Total number of isotopes in the library

PCE(30): Branching intensities of the isotopes

THA Half life of the isotope

THH

D.2.21. Subroutine MATRIX

A(80,30) Associated lines matrix

ACOL2(30) Ordering numbers

ACOL5(30) Saturation activities

COL(30) Names of the isotopes

CONF(30) Confidence values

EFER(100) Efficiency errors

EFFSUM(100) Intensities of the peaks

ENERG(100) Energies of the peaks

ETOL: Energy tolerance parameter

FITCH(100) Centroid of the peaks

GREF(30) Branching ratios of the gamma lines of an isotope

HALF(30) Half lives of the isotopes

ITUNN(30) Names of the isotopes

NUMF: Number of peaks

D.2.22. Subroutine ORDER

S(30)
T(30)
W(30)
Z(30) Vectors to be ordered
Y(30)
X(30)

SNEW
TNEW
WNEW New values to be entered into the vectors
ZNEW
XNEW

APPENDIX-E. LISTING OF BUCAASA

```
FTN5,I=NEFIND,L=L,B=B.

      PROGRAM FIND(INPUT,OUTPUT,OUF,TAPE5=INPUT,TAPE6=OUTPUT,TAPE11=OUF)
*****************************************************************************
*     PROGRAM BUCAASA/FIND                                                  *
*     AUTHOR: DR. OZER CIFTCIOGLU, ITU (1979)                               *
*     REVISED AND DEVELOPED BY : LEVENT AKIN, BU (1984)                     *
*****************************************************************************
      PARAMETER(NPEAK=100,NCHAN=4096)
      DIMENSION  AREAP(NPEAK),BGDP(NPEAK)
      DIMENSION DATA(NCHAN),SDATA(NCHAN),DDATA(NCHAN),SM(NCHAN)
      DIMENSION PCERR(NPEAK),TDATA(NCHAN),BDATA(NCHAN),A(NPEAK,10)
      DIMENSION H1(NPEAK),H2(NPEAK),MIN(NPEAK),MAX(NPEAK),JFIT(NPEAK)
      DIMENSION  PEAK(NPEAK,10),IPKN(NPEAK),SIG(NPEAK)
      COMMON ATGE,ATLT,FAC,ADEL,ATEST
      CHARACTER*6 NAME,NAME2
      CHARACTER*10 TIME,DATE,GUN,SAAT
      CHARACTER*50 JOB
      DATA ON/2.302585141/
      GUN=DATE()
      SAAT=TIME()
      WRITE(6,*)'JOB ?'
      READ(5,4) JOB
      WRITE(6,*)'ATGE ATLT FAC ADEL ATEST PTEST'
      READ(5,*) ATGE,ATLT,FAC,ADEL,ATEST,PTEST
      WRITE(6,*)' INIT IFIN ISMT IDISK ISTOR ISWO'
 100  READ(5,*) INIT,IFIN,ISMT,IDISK,ISTOR,ISWO
      WRITE(11,3) GUN,SAAT,JOB,INIT,IFIN,ATGE,ATLT,FAC,ADEL,ATEST,PTEST
      DO 101 I=1,NCHAN
      DATA(I)=0.0
      DDATA(I)=0.0
      SDATA(I)=0.0
 101  SM(I)=0.0
      IF(IDISK.EQ.0) THEN
        WRITE(11,*)' SPECTRUM INPUT MANUALLY'
        WRITE(6,*)'PLEASE ENTER SPECTRUM DATA'
        DO 200 I=INIT,IFIN
 200    READ(5,*)DATA(I)
      ELSE
        WRITE(6,*)'NAME OF THE SPECTRUM FILE?'
        READ(5,2)NAME
        OPEN(UNIT=16,FILE=NAME,ACCESS='DIRECT',FORM='UNFORMATTED',
     *  RECL=2)
        DO 210 I=INIT,IFIN
 210    READ(16,REC=I)DATA(I)
        WRITE(11,*)' SPECTRUM INPUT FROM FILE ',NAME
      ENDIF
      CALL SMOOTH(INIT,IFIN,ISMT,DATA,SDATA,DDATA,SM,NCHAN)
      IF(ISWO.EQ.1) THEN
        WRITE(11,15)
        DO 122  I=INIT,IFIN
 122    WRITE(11,14) I,DATA(I),SDATA(I),DDATA(I),SM(I)
      ENDIF
      CALL PEAKID(INIT,IFIN,K,ISMT,DATA,SDATA,DDATA,SM,NCHAN,NPEAK,H1,
     &H2,MIN,MAX,JFIT,IPKN,PEAK)
      IF(K.LE.0) THEN
        WRITE(11,11)
```

```
56                    STOP
57                  ENDIF
58                  DO 320 N=1,K
59                  NMIN=MIN(N)+1
60                  NMAX=MAX(N)-1
61                  AREA=0.0
62                  BGD=0.0
63                  AN=MAX(N)-MIN(N)
64                  DO 310 L=NMIN,NMAX
65                  BDATA(L)=EXP(((H2(N)-H1(N))/AN*(L-MIN(N))+H1(N))*ON)
66                  TDATA(L)=DATA(L)-BDATA(L)
67                  IF(TDATA(L).LT.0.0) TDATA(L)=0.0
68                  AREA=AREA+TDATA(L)
69             310  BGD=BGD+BDATA(L)
70                  AREAP(N)=AREA
71                  BGDP(N)=BGD
72                  PCERR(N)=SQRT(AREA+BGD+BGD)*100./AREA
73                  H1(N)=EXP(H1(N)*ON)
74                  H2(N)=EXP(H2(N)*ON)
75                  A(N,1)=H1(N)
76                  A(N,2)=(H1(N)-H2(N))/(MIN(N)-MAX(N))
77                  A(N,3)=-A(N,2)*0.01
78                  DO 315 L=1,JFIT(N)
79                  LL=L+3
80                  KK=INT(PEAK(N,L)+0.5)-2
81                  A(N,LL)=0.0
82                  DO 314 I=KK+1, KK+3
83                  CC=1.0
84                  DO 312 J=KK+1,KK+3
85                  IF(I.EQ.J) GO TO 312
86                  CC=CC*(PEAK(N,L)-J)/(I-J)
87             312  CONTINUE
88             314  A(N,LL)=A(N,LL)+TDATA(I)*CC
89             315  CONTINUE
90             320  SIG(N)=AREAP(N)/(A(N,4)*2.5066283)
91                  NM=0
92                  YY=0.0
93                  EY=0.0
94                  E=0.0
95                  E2=0.0
96                  DO 325 N=1,K
97                  IF(PCERR(N).LE.PTEST.AND.JFIT(N).EQ.1)THEN
98                    NM=NM+1
99                  YY=YY+SIG(N)
100                 EY=EY+SIG(N)*PEAK(N,1)
101                 E=E+PEAK(N,1)
102                 E2=E2+PEAK(N,1)**2
103                 ENDIF
104            325  CONTINUE
105                 IF(NM.GT.1) THEN
106                   P=(YY*E2-EY*E)/(NM*E2-E*E)
107                   Q=(NM*EY-E*YY)/(NM*E2-E*E)
108                 DO 326 N=1,K
109                   IF(PCERR(N).LE.PTEST)SIG(N)=P+Q*PEAK(N,1)
110            326  CONTINUE
111                 ENDIF
112                 IF(ISTOR.GE.1) THEN
```

```
                WRITE(6,*)'NAME OF THE INTERMEDIATE FILE'
               READ(5,2)NAME2
                OPEN(UNIT=15,FILE=NAME2,ACCESS='DIRECT',RECL=50)
                LN=1
                DO 340 N=1,K
                IF(PCERR(N).LE.PTEST) THEN
                  LN=LN+1
                  WRITE(15,REC=LN)LN,MIN(N),MAX(N),JFIT(N),(PEAK(N,L),L=1,
              *     JFIT(N)),(A(N,L),L=1,JFIT(N)+3),SIG(N)
                ENDIF
            340   CONTINUE
                  WRITE(15,REC=1)LN
                ENDIF
                WRITE(11,9)
                LN=0
                DO 350 N=1,K
                IF(PCERR(N).LE.PTEST) THEN
                  LN=LN+1
                  WRITE(11,10)LN,MIN(N),MAX(N),AREAP(N),BGDP(N),PCERR(N),SIG(N),
              &   (PEAK(N,L),L=1,JFIT(N))
                ELSE
                  WRITE(11,17)MIN(N),MAX(N),AREAP(N),BGDP(N),PCERR(N),SIG(N),
              &               (PEAK(N,L),L=1,JFIT(N))
                ENDIF
            350   CONTINUE
                  WRITE(11,12) LN
                  STOP
      ************************************************************************
                2  FORMAT(A6)
                4  FORMAT(A50)
                3  FORMAT(1H1,/10('*'),' BUCAASA/FIND REV. 2.1 ',2A10,10('*')//
                  &' JOB: ',A50/' PEAK SEARCH BETWEEN',I5,' AND',I5,' CHANNELS'/
                  &' ATGE=',F5.2,' ATLT=',F5.2,' FAC=',F5.2,' ADEL=',F5.2,
                  &' ATEST=',F5.2,' PTEST=',F5.2/)
               15  FORMAT(//2X,' SMOOTHED DATA AND FIRST DERIVATIVES'//2X,
                  *'CH.NO',8X,'DATA',11X,'SDATA',11X,'DDATA',11X,'STD.DEV')
               14  FORMAT(I6,2X,E14.6,2X,E14.6,4X,E14.6,4X,E14.6)
               11  FORMAT(//10X,'*** NO PEAKS WERE DETECTED ***')
                9  FORMAT(T20,'*** PEAKS FOUND ***'//,
                  &7X,'MIN.CHN',1X,'MAX.CHN',2X,'PEAK AREA',2X,
                  &'COMP.CONT',3X,'STD.DEV',2X,'SIGMA',4X,'PEAK CHANNELS'/)
               10  FORMAT(1X,I3,2(4X,I4),2(4X,F8.0),4X,F5.1,4X,F5.3,5(4X,F8.3))
               12  FORMAT(//10X,'TOTAL NUMBER OF DETECTED PHOTOPEAK GROUPS',I4)
               17  FORMAT(4X,2(4X,I4),2(4X,F8.0),4X,F5.1,4X,F5.3,5(4X,F8.3))
      ************************************************************************
                END
```

--VARIABLE MAP--(LO=A)

-NAME-	--ADDRESS--	BLOCK-----	PROPERTIES-------	TYPE---------	SIZE
A	62222B			REAL	1000
ADEL	3B	//		REAL	
AN	67473B			REAL	
AREA	67471B			REAL	
AREAP	1546B			REAL	100
ATEST	4B	//		REAL	

-NAME-	--ADDRESS--	BLOCK-----	PROPERTIES-------	TYPE---------	SIZE
ATGE	0B	//		REAL	
ATLT	1B	//		REAL	
BDATA	522228			REAL	4096
BGD	67472B			REAL	
BGDP	1712B			REAL	100
CC	67502B			REAL	

```
FIN5,I=NEFIND,L=L,B=8.
```

```
      1                SUBROUTINE SMOOTH(INIT,IFIN,ISMT,DATA,SDATA,DDATA,SM,NCHAN)
      2                DIMENSION NMCOE(6,5),MCOE(5),NDEN(5)
      3                DIMENSION DATA(NCHAN),SDATA(NCHAN),DDATA(NCHAN),SM(NCHAN)
      4                DATA NMCOE,MCOE,NDEN/2,1,4*0,17,12,-3,3*0,7,6,3,-2,2*0,179,135,30,
      5               *-55,15,0,143,120,60,-10,-45,18,4,35,21,2*429,2,10,28,60,110/
      6                IF(ISMT.GT.4) GO TO 400
      7                IO=ISMT+1
      8                INIT=INIT+IO
      9                IFIN=IFIN-IO
     10                DO 200 I=INIT,IFIN
     11                SDATA(I)=DATA(I)*NMCOE(1,IO)
     12                DO 210 K=1,IO
     13                L=I-K
     14                M=I+K
     15            210 SDATA(I)=SDATA(I)+(DATA(L)+DATA(M))*NMCOE(K+1,IO)
     16                SDATA(I)=SDATA(I)/MCOE(IO)
     17                IF(DATA(I).LT.4.) GO TO 212
     18                IF(DATA(I)-SDATA(I).GT.DATA(I)) SDATA(I)=DATA(I)
     19            212 IF(SDATA(I).LT.1.) SDATA(I)=1.
     20            200 SM(I)=SQRT(SDATA(I))
     21                INIT=INIT+IO
     22                IFIN=IFIN-IO
     23                DO 300 I=INIT,IFIN
     24                DDATA(I)=0.0
     25                DO 310 K=1,IO
     26                L=I-K
     27                M=I+K
     28            310 DDATA(I)=DDATA(I)+K*(SDATA(M)-SDATA(L))
     29            300 DDATA(I)=DDATA(I)/NDEN(IO)
     30                RETURN
     31            400 DO 450 I=INIT,IFIN
     32                SDATA(I)=DATA(I)
     33                IF(SDATA(I).LT.1.) SDATA(I)=1.
     34            450 SM(I)=SQRT(SDATA(I))
     35                INIT=INIT+2
     36                IFIN=IFIN-2
     37                DO 460 I=INIT,IFIN
     38                DDATA(I)=0.0
     39                DO 430 K=1,2
     40                L=I-K
     41                M=I+K
     42            430 DDATA(I)=DDATA(I)+K*(SDATA(M)-SDATA(L))
     43            460 DDATA(I)=DDATA(I)/NDEN(2)
     44                RETURN
     45                END
```

--VARIABLE MAP--(LO=A)

-NAME-	--ADDRESS-	-BLOCK-	----PROPERTIES----	----TYPE----	----SIZE----
DATA	4	DUMMY-ARG		REAL	ADJ-ARY
DDATA	6	DUMMY-ARG		REAL	ADJ-ARY
I	4008			INTEGER	
IFIN	2	DUMMY-ARG		INTEGER	
INIT	1	DUMMY-ARG		INTEGER	
IO	3778			INTEGER	
ISMT	3	DUMMY-ARG		INTEGER	
K	402B			INTEGER	
L	404B			INTEGER	
M	405B			INTEGER	

```
      SUBROUTINE PEAKID(II,IFF,JPK,ISMT,DATA,SDATA,DDATA,SM,NCHAN,NPEAK
     *,H1,H2,MIN,MAX,JFIT,IPKN,PEAK)
      DIMENSION INT1(20),INT2(20),FOOT(20),PK(20),JMUL(10)
      DIMENSION  DATA(NCHAN),SDATA(NCHAN),DDATA(NCHAN),SM(NCHAN)
      DIMENSION H1(NPEAK),H2(NPEAK),MIN(NPEAK),MAX(NPEAK),JFIT(NPEAK),
     *IPKN(NPEAK),PEAK(NPEAK,10)
      COMMON ATGE,ATLT,FAC,ADEL,ATEST
      DATA J,IFST,IFNL,DN/1,0,0,2.302585141/
      IST=II
  100 IFT=1
      IF(DATA(IST).GE.1000.) THEN
        IF(DATA(IST).GE.3000.) THEN
          ATEN=FAC*SQRT(DATA(IST)*2.E-4)
        ELSE
          ATEN=ATGE
        ENDIF
      ELSE
        ATEN=ATLT
      ENDIF
      IF(ISMT.LE.1.OR.ISMT.GT.4) THEN
        ADL=1.
      ELSE
        SMP=0.25*(ISMT-1)
        ADL=1.+SMP
        ATEN=ATEN*(1.-SMP)
      ENDIF
      ADL=ADL*ADEL
      ND1=ADL
      ND2=ADL+ADL
      ND4=ADL+ADL+ADL+ADL
      ND5=ND4+ADL
      ATES=ATEN*0.4*ATEST
  300 ICAUT=0
      IRT=0
      JMUL(IFT)=1
      DO 305 N=IST,IFF
      DO=ATEN*SM(N)
      IF(DDATA(N).GT.DO) GO TO 310
  305 CONTINUE
      J=J-1
      JPK=J
      IF(J) 800,900,800
  310 IF(IFT.LE.1) THEN
        MIN(J)=N-1
        MEI=MIN(J)
      ELSE
          IF(ABS(DDATA(N+1)).LE.DO) THEN
            IFT=IFT-1
            MAX(J)=IST-1
            GO TO 600
          ENDIF
      ENDIF
      IRG=0
      DERI=0.0
      NND=N+ND2
```

```
                DO 370 M=NND,IFF
                DO=ATEN*SM(M)
                IF(DDATA(M).LE.DO) THEN
                  ICAUT=ICAUT+1
                  IF(DDATA(M).LT.0.0) GO TO 400
                  IF(IRT.EQ.0) THEN
                    IF(ICAUT.NE.ND2) THEN
                      JMUL(IFT)=JMUL(IFT)+1
                      IRT=1
                    ELSE
                      GO TO 370
                    ENDIF
                  ENDIF
                  IF(ICAUT.GE.ND4) THEN
                    IST=M
                    IF(IFT.EQ.1) GO TO 100
                    IFT=IFT-1
                    GO TO 600
                  ENDIF
                ELSE
                  DERI=DDATA(M-1)
                  IF(ABS(DDATA(M) -DERI).LE.DO) THEN
                    IRG=IRG+1
                    IF(IRG.EQ.ND4) JMUL(IFT)=JMUL(IFT)+1
                  ENDIF
                  ICAUT=0
                ENDIF
            370 CONTINUE
                GO TO 750
            400 PM=M-1
                DELM=DDATA(M-1)/(DDATA(M-1)-DDATA(M))
                PK(IFT)=PM+DELM
                MM=M+NDL
            405 IIT=0
                IRG=0
                DERI=0.0
                DO 450 L=MM,IFF
                DON=-ATEN*SM(L)
                IF(DDATA(L).GT.DON) GO TO 505
                DERI=DDATA(L-1)
                DMIN=SM(L)*ATES
                IF(ABS(DDATA(L) -DERI).GT.DMIN) THEN
                  IIT=0
                  IF(ABS(DDATA(L) -DERI).LE.-DON) THEN
                    IRG=IRG+1
                    IF(IRG.EQ.ND4) JMUL(IFT)=JMUL(IFT)+1
                  ELSE
                    IRG=0
                  ENDIF
                ELSE
                    IF(IIT.LT.ND2) THEN
                      IIT=IIT+1
                    ELSE
                      MEF=L-IIT-1
                      IF(DATA(MEI).GT.DATA(MEF).OR.DDATA(MEF).GE.0.0) GO TO 500
                      JMUL(IFT)=JMUL(IFT)+1
                      WRITE(11,5) MEF
```

```
113                 IIT=0
114               ENDIF
115             ENDIF
116      450    CONTINUE
117             IFNL=1
118             JPK=J
119             GO TO 800
120      500    L=L-IIT-1
121      505    INT1(IFT)=L
122             IIT=0
123             LL=L
124      510    LL=LL+1
125             IF(LL-IFF) 515,520,795
126      515    LDEL=LL-L
127             IF(LDEL.GE.ND5) GO TO 530
128      520    DO=ATEN*SM(LL)
129             IF(ABS(DDATA(LL)).LE.DO) GO TO 510
130             DMIN=SM(L)*ATES
131             DERI=DDATA(LL-1)
132             IF(ABS(DDATA(LL)-DERI).LT.DMIN) GO TO 510
133      530    INT2(IFT)=LL-1
134             IF(DDATA(LL-1).LT.20.) THEN
135               SUM=0.0
136               DO 540 IL=INT1(IFT),INT2(IFT)
137      540      SUM=SUM+ALOG10(DATA(IL))
138               FOOT(IFT)=SUM/LDEL
139             ELSE
140               FOOT(IFT)=ALOG10(DATA(L))
141             ENDIF
142             IF(LDEL.GT.ND2) GO TO 600
143             IF(DDATA(LL).GE.0.0) THEN
144               IFT=IFT+1
145               PK(IFT)=0.0
146               IST=LL
147               GO TO 300
148             ENDIF
149             MM=LL
150             JMUL(IFT)=JMUL(IFT)+1
151             WRITE(11,5) MM
152             GO TO 405
153      600    FIF=FOOT(IFT)
154             IF(MIN(J).GT.II)  THEN
155               LPT=1
156               LST=MIN(J)
157               IF(DATA(LST).LE.20.) THEN
158                 SUM=ALOG10(DATA(LST))
159      620        LRV=LST-LPT
160                 IF(LRV.LT.II) GO TO 635
161                 DO=ATEN*SM(LRV)
162                 IF(ABS(DDATA(LRV)).GE.DO) GO TO 635
163      630        SUM=SUM+ALOG10(DATA(LRV))
164                 LPT=LPT+1
165                 IF(LPT-ND5) 620,635,635
166      635        FTI=SUM/LPT
167               ELSE
168                 FTI=ALOG10(DATA(LST))
169               ENDIF
```

```
170               ELSE
171                 MIN(J)=II
172                 IFST=1
173                 FTI=ALOG10(DATA(II))
174               ENDIF
175               IF(IFT.LE.1) THEN
176               H1(J)=FTI
177               H2(J)=FOOT(IFT)
178               PEAK(J,1)=PK(1)
179               IPKN(J)=1
180               MAX(J)=INT1(IFT)
181               IST=MAX(J)+1
182               IPEK=PEAK(J,1)+0.5
183               IF(MAX(J)-IPEK.GT.ND2) THEN
184                 JPK=J
185                 JFIT(J)=JMUL(J)-1
186                 MULT=JFIT(J)+1
187                 JFIT(J)=IPKN(J)
188                 J=J+1
189               ENDIF
190               ELSE
191               PIN=MIN(J)
192               PFI=INT1(IFT)
193               H1(J)=FTI
194               IO=0
195               I=1
196               JFIT(J)=-1
197               SLP=(FIF-FTI)/(PFI-PIN)
198          705  PIM=INT1(I)
199               FC1=FTI+SLP*(PIM-PIN)
200               K1=I-IO
201               PEAK(J,K1)=PK(I)
202               DO=ATEN*SM(INT1(I))
203               EFCM=EXP(FOOT(I)*ON)-EXP(FC1*ON)
204               JFIT(J)=JFIT(J)+JMUL(I)
205               IF(EFCM.LE.DO) THEN
206                 MAX(J)=INT1(I)
207                 H2(J)=FOOT(I)
208                 IPKN(J)=I-IO
209                 IO=I
210                 MULT=JFIT(J)+1
211                 JFIT(J)=IPKN(J)
212                 J=J+1
213                 MIN(J)=INT2(I)
214                 H1(J)=FOOT(I)
215                 JFIT(J)=-1
216               ENDIF
217          720  I=I+1
218               IF(I-IFT) 705,725,725
219          725  MAX(J)=INT1(IFT)
220               H2(J)=FOOT(IFT)
221               IPEK=INT2(IFT-1)
222               IF(MAX(J)-IPEK.GT.ND2) THEN
223                 JPK=J
224                 PEAK(J,I-IO)=PK(I)
225                 IPKN(J)=IFT-IO
226                 JFIT(J)=JFIT(J)+JMUL(IFT)
```

```
      227                 MULT=JFIT(J)+1
      228                 JFIT(J)=IPKN(J)
      229                 J=J+1
      230               ENDIF
      231               IST=INT2(IFT)
      232               ENDIF
      233               GO TO  100
      234           750 IF(IFT-1) 900,760,751
      235           751 IFT=IFT-1
      236               MAX(J)=INT1(IFT)
      237               H2(J)=FOOT(IFT)
      238               JFIT(J)=JFIT(J)-1
      239               MULT=JFIT(J)+1
      240               GO TO 795
      241           760 J=J-1
      242               JPK=J
      243               IF(J)900,900,795
      244           795 IFNL=1
      245           800 IF(IFST.GT.0)WRITE(11,40)
      246               IF(IFNL.GT.0)WRITE(11,50)
      247               WRITE(11,20) JPK
      248           900 RETURN
      249             5 FORMAT(7X,'THERE MAY BE A PEAK NEAR CHANNEL # ',I4)
      250            20 FORMAT(//10X,' NUMBER OF DETECTED PEAKS IN THIS REGION : ',I3//)
      251            40 FORMAT(//10X,'INSPECT THE FIRST PEAK CAREFULLY...')
      252            50 FORMAT(//10X,'INSPECT THE LAST PEAK CAREFULLY...')
      253               END
```

--VARIABLE MAP--(LO=A)

-NAME-	-ADDRESS-	-BLOCK-	-PROPERTIES-	-TYPE-	-SIZE-
ADEL	3B	//		REAL	
ADL	1521B			REAL	
ATEN	1520B			REAL	
ATES	1527B			REAL	
ATEST	4B	//		REAL	
ATGE	0B	//		REAL	
ATLT	1B	//		REAL	
DATA	5	DUMMY-ARG		REAL	ADJ-ARY
DDATA	7	DUMMY-ARG		REAL	ADJ-ARY
DELM	1544B			REAL	
DERI	1537B			REAL	
DMIN	1552B			REAL	
DO	1534B			REAL	
DON	1551B			REAL	
EFCM	1600B			REAL	
FAC	2B	//		REAL	
FC1	1576B			REAL	
FIF	1561B			REAL	
FOOT	1430B			REAL	20
FTI	1565B			REAL	
H1	11	DUMMY-ARG		REAL	ADJ-ARY
H2	12	DUMMY-ARG		REAL	ADJ-ARY
I	1573B			INTEGER	
ICAUT	1530B			INTEGER	
IFF	2	DUMMY-ARG		INTEGER	
IFNL	1514B			INTEGER	
IFST	1513B			INTEGER	
IFT	1517B			INTEGER	
II	1	DUMMY-ARG		INTEGER	
IIT	1546B			INTEGER	
IL	1557B			INTEGER	
INT1	1360B			INTEGER	20
INT2	1404B			INTEGER	20
IO	1572B			INTEGER	
IPEK	1566B			INTEGER	
IPKN	16	DUMMY-ARG		INTEGER	ADJ-ARY
IRG	1536B			INTEGER	
IRT	1531B			INTEGER	
ISMT	4	DUMMY-ARG		INTEGER	
IST	1516B			INTEGER	
J	1512B			INTEGER	
JFIT	15	DUMMY-ARG		INTEGER	ADJ-ARY
JMUL	1500B			INTEGER	10
JPK	3	DUMMY-ARG		INTEGER	
K1	1577B			INTEGER	
L	1547B			INTEGER	
LDEL	1555B			INTEGER	
LL	1554B			INTEGER	
LPT	1562B			INTEGER	
LRV	1564B			INTEGER	

```
      PROGRAM FIT(INPUT,OUTPUT,YAZ,TAPE6=OUTPUT,TAPE5=INPUT,TAPE11=YAZ)
*************************************************************************
*     PROGRAM BUCAASA/FI                                                *
*     AUTHOR: DR.OZER CIFTCIOGLU, ITU (1979)                            *
*     REVISED AND DEVELOPED BY : LEVENT AKIN, BU (1984)                 *
*************************************************************************
      DIMENSION AREA(5),EII(5),EAREA(5),VARI(5),EAFSR(15),EAREAG(5),
     &BBI(9),AG(5)
      COMMON AI(15),AF(15),EAF(15,15),AIC(5),RD(5,15),RIN(5)
      COMMON X(50),Y(50),W(50),F(50),WRES(50),BGD(50),XJ(6)
      COMMON II,IP,IC,NP,MUL,SIG,IVAR,IRPT,ISTOP,IN
      COMMON AT,BT,CT,SIGT
      EXTERNAL ERF
      LOGICAL JXST
      CHARACTER*7 NAM,NAM2,NAM3
      CHARACTER*10 TIME,DATE,GUN,SAAT
      CHARACTER*50 JOB
      GUN=DATE()
      SAAT=TIME()
      WRITE(11,1200) GUN,SAAT
 1200 FORMAT(1H1,6('*'),' BUCAASA/FIT REV: 3.0 ',2A10,6('*')/)
      PRINT*,'NAME OF JOB?'
      READ(5,1100) JOB
 1100 FORMAT(A50)
      WRITE(11,1110) JOB
 1110 FORMAT(' JOB : ',A50)
  100 CONTINUE
  200 PRINT*,'INUM IGUES IDISK ISAVE II  IVAR IRPT ISET'
      READ*,INUM,IGUES,IDISK,ISAVE,II,IVAR,IRPT,ISET
      ISTOP=0
      IF(IDISK.EQ.1) THEN
        PRINT*,'NAME OF THE SPECTRUM FILE?'
  256   READ(*,2) NAM2
        INQUIRE(FILE=NAM2,EXIST=JXST)
        IF(.NOT.JXST) THEN
          PRINT*,'THERE IS NO SUCH FILE'
          GO TO 256
        ENDIF
        WRITE(11,*)'SPECTRAL DATA READ IN FROM FILE',NAM2
      ENDIF
      IF(ISAVE.EQ.1) THEN
         PRINT*,'NAME OF THE OUTPUT FILE?'
         READ(*,2)NAM3
         OPEN(17,FILE=NAM3,ACCESS='DIRECT',RECL=30)
         WRITE(11,*)'OUTPUT FILE IS ',NAM3
      ENDIF
      IF(IGUES) 215,240,215
  215 PRINT*,'NAME OF THE INPUT FILE?'
      READ(5,1000) NAM
 1000 FORMAT(A7)
      INQUIRE(FILE=NAM,EXIST=JXST)
      IF(.NOT.JXST) THEN
         PRINT*,'THERE IS NO SUCH FILE'
         GO TO 215
      ENDIF
```

```
      WRITE(11,*) 'INITIAL VALUES READ IN FROM FILE ',NAM
      OPEN(15,FILE=NAM,ACCESS='DIRECT',RECL=50)
      IF(ISET.EQ.-1)THEN
      READ(15,REC=1)NPK
      IF(ISAVE.EQ.1.AND.ISET.EQ.-1)WRITE(17,REC=1)NPK
      ENDIF
220   READ(15,REC=INUM) NNN,INIT,IFIN,MUL,(XJ(J),J=1,MUL),
     *(BBI(J),J=1,MUL+3),SIG
      ISTOP=0
      DO 211 I=1,II
211   AI(I)=BBI(I)
      IN=II
      IF(IVAR.GT.4) THEN
       IN=II+MUL
        IP=II+MUL+MUL
      AIX=.15
      HX=.85
        AI(1)=AI(1)+AI(2)*(IFIN-INIT)*(1.0-AIX)
        DO 212 I=1,MUL
        K=II+I
212     AI(K)=(AIX-1.0)*AI(2)*(IFIN-INIT)*HX
        AI(2)=AI(2)*AIX
        J=II+MUL+1
      ELSE
        J=II+1
        IP=II+MUL
      ENDIF
      IF(IVAR.GT.3) THEN
         AT=0.1
         BT=0.5
         CT=1.0
         SIGT=1.75
      ENDIF
      K=3
      DO 213 I=J,IP
      K=K+1
213   AI(I)=BBI(K)
      IF(IVAR.GT.3) THEN
        DO 50 J=1,MUL
        L=IN+J
50      AIC(J)=AT*AI(L)
      ENDIF
      GO TO 250
240   PRINT*,' INIT IFIN MUL'
      READ*,INIT,IFIN,MUL
      PRINT*,'SIG,(XJ(J),J=1,',MUL, ')'
      READ*,SIG,(XJ(J),J=1,MUL)
      IP=II+MUL
      IN=II
      IF(IVAR.GT.4)THEN
        IN=IN+MUL
        IP=IP+MUL
      ENDIF
      PRINT*,'(AI(I),I=1,',IP,')'
      READ*,(AI(I),I=1,IP)
      IF(IVAR.GT.3)THEN
        PRINT*,'ENTER AT BT CT SIGT'
```

```
113                READ*,AT,BT,CT,SIGT
114               DO 249 J=1,MUL
115               L=IN+J
116        249    AIC(J)=AT*AI(L)
117               ENDIF
118        250   WRITE(11,4)INUM,INIT,IFIN,MUL,II,IVAR,IRPT,IGUES
119           *,IDISK,ISAVE,ISET
120          4   FORMAT(' INUM INIT IFIN   MUL   II  IVAR IRPT IGUES',
121           *' IDISK ISAVE ISET '/ 9I5,2X,2I5)
122               NP=IFIN-INIT+1
123               IF(IDISK.EQ.0) THEN
124                 PRINT*,'PLEASE ENTER SPECTRUM DATA'
125                 DO 255 I=1,NP
126        255      READ*,Y(I)
127               ELSE
128               OPEN(16,FILE=NAM2,ACCESS='DIRECT',RECL=2)
129                 DO 257 I=1,NP
130                 K=I+INIT-1
131        257      READ(16,REC=K) Y(I)
132               ENDIF
133               DO 300 I=1,NP
134               X(I)=I
135        300    W(I)=1./Y(I)
136               DO 360 J=1,MUL
137        360    XJ(J)=XJ(J)-INIT+1
138               IF(IVAR.LE.0) GO TO 950
139               GO TO (410,420,440,450,450),IVAR
140        410    IC=IP
141               GO TO 550
142        420    IC=IP+MUL
143               IP1=IP+1
144               DO 430 I=IP1,IC
145               K=I-IP1+1
146        430    AI(I)=XJ(K)
147               GO TO 550
148        440    IC=IP+MUL+1
149               IP1=IP+1
150               IP2=IC-1
151               DO 500 I=IP1,IP2
152               K=I-IP1+1
153        500    AI(I)=XJ(K)
154               AI(IC)=SIG
155               GO TO 550
156        450    IC=IP+MUL*2+4
157               IP1=IP+1
158               IP2=IP+MUL
159               DO 510   I=IP1,IP2
160               K=I-IP1+1
161        510    AI(I)=XJ(K)
162               IP3=IP2+1
163               IP4=IP2+MUL
164               DO 520 I=IP3,IP4
165               K=I-IP3+1
166        520    AI(I)=AIC(K)
167               AI(IC-3)=BT
168               AI(IC-2)=CT
169               AI(IC-1)=SIGT
```

```
      AI(IC)=SIG
550   WRITE(11,1500) (AI(I),I=1,II)
1500  FORMAT(1X,'INITIAL PARAMETERS:'/20('-')/
     *' COMPTON CONTINUUM'/' A0=',F10.1,:,' A1=',F8.2,:
     *' A2=',F8.2)
      IF(IVAR.GE.4)THEN
      IF(IVAR.EQ.5)WRITE(11,1600) (J,AI(II+J),J=1,MUL)
1600  FORMAT(/' STEP FUNCTION'/5(1X,:,'H',I1,'=',F8.2))
      WRITE(11,1700) (J,AI(IN+J),J=1,MUL)
1700  FORMAT(/' PEAK HEIGHTS (GAUSS)'/5(1X,:,'AG',I1,'=',F11.1))
      WRITE(11,1800) (J,AI(IN+2*MUL+J),J=1,MUL)
1800  FORMAT(/' TAILING HEIGHTS'/5(1X,:,'AT',I1,'=',F10.1))
      WRITE(11,1900) (J,AI(IN+MUL+J),J=1,MUL)
1900  FORMAT(/' PEAK CHANNELS'/5(1X,:,'X0',I1,'=',F9.3))
      WRITE(11,2000) (AI(I),I=IC-3,IC)
2000  FORMAT(/' B=',F6.3,' C=',F6.3,' SIGMAT=',F6.3,' SIGMAG=',F6.3)
      ELSE
      WRITE(11,1700) (J,AI(II+J),J=1,MUL)
      WRITE(11,1900) (J,AI(II+MUL+J),J=1,MUL)
      WRITE(11,2800) AI(IC)
2800  FORMAT(/' SIGMAG=',F6.3)
      ENDIF
      CALL VARMET
      IF(ISTOP.LE.0) GO TO 580
      GO TO 900
580   IF(IVAR.LT.4) GO TO 590
      DO 586 I=1,IC
586   EAFSR(I)=SQRT(EAF(I,I))
      WRITE(11,2100) (AF(I),EAFSR(I),I=1,II)
2100  FORMAT(/' FINAL PARAMETERS:'/18('-')/' COMPTON CONTINUUM'/
     &' A0=',F11.1,' +/-',F7.1,:,' A1=',F8.2,' +/-',F6.1,:,' A2=',
     &F6.2,' +/-',F6.1)
      IF(IVAR.EQ.5)WRITE(11,2200) (J,AF(II+J),EAFSR(II+J),J=1,MUL)
2200  FORMAT(/' STEP FUNCTION'/5(1X,:,'H',I1,'=',F8.2,' +/-',F6.1))
      WRITE(11,2300) (J,AF(IN+J),EAFSR(IN+J),J=1,MUL)
2300  FORMAT(/' PEAK HEIGHTS (GAUSS)'/5(1X,:,'AG',I1,'=',F11.1,
     &' +/-',F7.1))
      WRITE(11,2400) (J,AF(IN+2*MUL+J),EAFSR(IN+2*MUL+J),
     &J=1,MUL)
2400  FORMAT(/' TAILING HEIGHTS'/5(1X,:,'AT',I1,'=',F10.1,' +/-',
     &F6.1))
      WRITE(11,2500) (AF(I),EAFSR(I),I=IC-3,IC),AF(IC)*2.35482
2500  FORMAT(/' B=',F6.3,' +/-',F5.3,' C=',F6.3,' +/-',F5.3,
     &' SIGMAT=',F6.3,' +/-',F5.3,' SIGMAG=',F6.3,' +/-',F5.3,
     &' FWHM=',F6.3)
      GO TO 595
590   WRITE(11,2100)(AF(I),EAF(I,I),I=1,II)
      WRITE(11,2300) (J,AF(II+J),EAF(II+J,II+J),J=1,MUL)
      WRITE(11,2900) AF(IC),EAF(IC,IC),AF(IC)*2.25482
2900  FORMAT(/' SIGMAG=',F6.3,' +/-',F5.3,' FWHM=',F6.3)
595   ERRC=0.0
      D1=(Y(NP)-F(NP))/SQRT(ABS(F(NP)))
      DO 630 I=1,NP
      D0=D1
      D1=(Y(I)-F(I))/SQRT(ABS(F(I)))
630   ERRC=ERRC+D0*D1
      ERRC=ERRC/(NP-IC)
```

```
      DO 700 J=1,MUL
      IF(IVAR.GT.3) GO TO 688
      K=J+II
      AREA(J)=AF(K)*(2.5066282*SIG)
      IF(IVAR.GT.2) GO TO 680
      EAREA(J)=EAF(K,K)/AF(K)*100.
      XJ(J)=XJ(J)+INIT-1
      GO TO 700
680   EAF(K,K)=EAF(K,K)/AF(K)
      EAF(IC,IC)=EAF(IC,IC)/AF(IC)
      EAREA(J)=SQRT(EAF(K,K)**2+EAF(IC,IC)**2)*100.
      GO TO 699
688   K=J+IN
      EAREAG(J)=SQRT(EAF(K,K)/(AF(K)**2)+EAF(IC,IC)/(AF(IC)**2)+
     &2*EAF(IC,K)/(AF(K)*AF(IC)))
      AREA(J)=AF(K)*2.5066282*SIG
      CALL RINT
      AG(J)=AREA(J)
      AREA(J)=AREA(J)+RIN(J)
      RD(J,1)=2.5066282*SIG
      RD(J,IC)=AF(K)*2.5066282
      IIJ=II+J
      IPJ=IP+J
      IICJ=IP+MUL+J
      VARI(J)=RD(J,1)*(RD(J,1)*EAF(IIJ,IIJ)+2*RD(J,2)*EAF(IIJ,IPJ)+
     &2*RD(J,3)*EAF(IIJ,IICJ)+2*RD(J,IC-3)*EAF(IIJ,IC-3)+
     &2*RD(J,IC-2)*EAF(IIJ,IC-2)+2*RD(J,IC-1)*EAF(IIJ,IC-1)+
     &2*RD(J,IC)*EAF(IIJ,IC))+RD(J,2)*(RD(J,2)*EAF(IPJ,IPJ)+
     &2*RD(J,3)*EAF(IPJ,IICJ)+2*RD(J,IC-3)*EAF(IPJ,IC-3)+
     &2*RD(J,IC-2)*EAF(IPJ,IC-2)+2*RD(J,IC-1)*EAF(IPJ,IC-1)+
     &2*RD(J,IC)*EAF(IPJ,IC))+RD(J,3)*(RD(J,3)*EAF(IICJ,IICJ)+
     &2*RD(J,IC-3)*EAF(IICJ,IC-3)+2*RD(J,IC-2)*EAF(IICJ,IC-2)+
     &2*RD(J,IC)*EAF(IICJ,IC))
      DO 693 I=IC-3,IC
      DO 693 N=IC-3,IC
693   VARI(J)=VARI(J)+RD(J,I)*RD(J,N)*EAF(I,N)
      EII(J)=(SQRT(ABS(VARI(J)))/AREA(J))*100.
      EAREA(J)=EII(J)
699   XJ(J)=XJ(J)+INIT-1
700   CONTINUE
      IF(IVAR.GE.4) THEN
      WRITE(11,2700) (J,XJ(J),EAFSR(IN+MUL+J),RIN(J),AG(J),
     &AREA(J),EAREA(J),J=1,MUL)
2700  FORMAT(//5X,'PEAK CHNL',16X,'TAILING AREA',4X,
     &'GAUSS AREA',4X,'TOTAL AREA',4X,'% ERROR'/
     &5(1X,:,I2,2X,F8.3,' +/-',4X,F5.3,4X,F9.1,4X,F11.1,4X,
     &F11.1,4X,F7.3/))
      ELSE
      WRITE(11,3000)(J,XJ(J),EAF(II+MUL+J,II+MUL+J),AREA(J),
     &EAREA(J),J=1,MUL)
3000  FORMAT(//5X,'PEAK CHNL',19X,'AREA',4X,'% ERROR'/
     &5(1X,:,I2,2X,F8.3,' +/-',F5.3,4X,F11.1,4X,F7.3/))
      ENDIF
      IF(ISAVE.EQ.1)WRITE(17,REC=INUM)MUL,(XJ(J),EAF(J,J),AREA(J
     &EAREA(J),J=1,MUL)
      WRITE(11,631) ERRC
631   FORMAT(/15X,'ERR.CORR=',1PE14.7//)
```

```
      284                IF(IVAR.EQ.5.OR.IRPT.EQ.0)GO TO 900
      285                DO 730 J=1,MUL
      286           730  XJ(J)=XJ(J)-INIT+1
      287                IVAR=IVAR+1
      288                WRITE(11,14)
      289            14  FORMAT(' THE INITIAL VALUES ARE OBTAINED FROM THE PREVIOUS FIT')
      290                WRITE(11,15) INUM,IVAR,IRPT,ISET
      291            15  FORMAT(1X,'  INUM IVAR IRPT ISET '/1X,4I5)
      292                DO 770 I=1,II
      293           770  AI(I)=AF(I)
      294                IF(IVAR.EQ.4) THEN
      295                  AT=0.11
      296                  BT=0.3
      297                  CT=1.1
      298                  SIGT=5.0
      299                  IN=II
      300                  DO 775 J=1,MUL
      301           775    AIC(J)=AT*AI(II+J)
      302                  IP=II+MUL
      303                ENDIF
      304                IF(IVAR.EQ.5)THEN
      305                  IN=II+MUL
      306                  IP=II+MUL+MUL
      307                  DO 780 I=II+1,II+MUL
      308           780    AI(I)=1.
      309                ENDIF
      310                GO TO(410,420,440,450,450),IVAR
      311           900  IF(ISET) 952,920,910
      312           910  IF(ISET-1) 920,200,100
      313           950  WRITE(11,16)
      314            16  FORMAT(' PLEASE CHECK THE INITIAL VALUES ')
      315             2  FORMAT(A7)
      316           920  STOP
      317           952  INUM=INUM+1
      318                IF(INUM.LE.NPK) GO TO 220
      319                STOP
      320                END
```

--VARIABLE MAP--(LO=A)

-NAME-	--ADDRESS--	BLOCK-----	PROPERTIES-------	TYPE---------	SIZE
AF	17B	//		REAL	15
AG	3612B			REAL	5
AI	0B	//		REAL	15
AIC	377B	//		REAL	5
AIX	3647B			REAL	
AREA	3531B			REAL	5
AT	1220B	//		REAL	
BBI	3601B			REAL	9
BGD	1116B	//		REAL	50
BT	1221B	//		REAL	
CT	1222B	//		REAL	
DO	3715B			REAL	
D1	3713B			REAL	
EAF	36B	//		REAL	225
EAFSR	3555B			REAL	15

-NAME-	--ADDRESS--	BLOCK-----	PROPERTIES-------	TYPE---------	SIZE
EAREA	3543B			REAL	5
EAREAG	3574B			REAL	5
EII	3536B			REAL	5
ERRC	3712B			REAL	
F	752B	//		REAL	50
GUN	3623B			CHAR*10	
HX	3650B			REAL	
I	3645B			INTEGER	
IC	1210B	//		INTEGER	
IDISK	3634B			INTEGER	
IFIN	3642B			INTEGER	
IGUES	3633B			INTEGER	
II	1206B	//		INTEGER	
IICJ	3721B			INTEGER	
IIJ	3717B			INTEGER	

```
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I=AD,L=L,B=ADB.

 1                SUBROUTINE VARMET
 2                DIMENSION XA(15),XB(15),GA(15),GB(15),GC(15),S(15),SMA(15),
 3               *H(15,15)
 4                COMMON AI(15),AF(15),EAF(15,15),AIC(5),RD(5,15),RIN(5)
 5                COMMON X(50),Y(50),W(50),F(50),WRES(50),BGD(50),XJ(6)
 6                COMMON II,IP,IC,NP,MUL,SIG,IVAR,IRPT,ISTOP,IN
 7                COMMON AT,BT,CT,SIGT
 8                EPS=1.E-11
 9                IXX=1
10                STEP=0.01
11           10   DO 20 I=1,IC
12           20   XA(I)=AI(I)
13           30   IKNT=0
14                IT=0
15           35   DO 40 I=1,IC
16                DO 40 J=1,IC
17           40   H(I,J)=0.0
18                DO 50 I=1,IC
19           50   H(I,I)=1.0
20                CALL DERIVS(XA,XFA,GA)
21                IF(IXX.EQ.1)WRITE(11,1) XFA
22            1   FORMAT(1X,'INITIAL CHISQR=',E14.6/1X,' GRADIENTS')
23                WRITE(11,2) (GA(I),I=1,IC)
24            2   FORMAT(1X,5E14.6)
25           60   IKNT=IKNT+1
26                CSQ1=XFA
27                GPA=0.0
28                DO 80 I=1,IC
29                S(I)=0.0
30                DO 70 J=1,IC
31           70   S(I)=S(I)-H(I,J)*GA(J)
32           80   GPA=GPA+GA(I)*S(I)
33                IF(GPA.GE.0.0) GO TO 30
34                ETAA=-(XFA+XFA)/GPA
35                IF(ETAA.GT.1.0) ETAA=1.0
36           90   DO 100 I=1,IC
37                SMA(I)=ETAA*S(I)
38          100   XB(I)=XA(I)+SMA(I)
39                GO TO (200,115,117,119,119),IVAR
40          115   K1=IP+1
41                DO 116 I=K1,IC
42                J=I-K1+1
43          116   XJ(J)=XB(I)
44                GO TO 200
45          117   K1=IP+1
46                K2=IC-1
47                DO 118 I=K1,K2
48                J=I-K1+1
49          118   XJ(J)=XB(I)
50                SIG=XB(IC)
51                GO TO 200
52          119   K1=IP+1
53                K2=IP+MUL
54                DO 121 I=K1,K2
55                J=I-K1+1
```

```
      121  XJ(J)=XB(I)
           K3=K2+1
           K4=K2+MUL
           DO 122 I=K3,K4
           K=I-K3+1
      122  AIC(K)=XB(I)
           BT=XB(IC-3)
           CT=XB(IC-2)
           SIGT=XB(IC-1)
           SIG=XB(IC)
      200  GAMMA=ETAA
           CALL DERIVS(XB,XFB,GB)
           GPB=0.0
           DO 210  I=1,IC
      210  GPB=GPB+GB(I)*S(I)
           IF(XFB.GE.XFA.OR.GPB.GE.0.0) GO TO 220
           ETAA=ETAA+ETAA
           GO TO 90
       220 IIT=0
           XFD=XFA
       230 CALL CUBINT(XFA,XFB,GAMMA,GPA,GPB,ALPHA)
           IF(ALPHA.LE.0.0) GO TO 510
           DO 240 I=1,IC
           SMA(I)=ALPHA*S(I)
       240 XB(I)=XA(I)+SMA(I)
           GO TO (300,151,153,162,162),IVAR
      151  K1=IP+1
           DO 152 I=K1,IC
      152  XJ(I-K1+1)=XB(I)
           GO TO 300
      153  K1=IP+1
           K2=IC-1
           DO 154 I=K1,K2
      154  XJ(I-K1+1)=XB(I)
           SIG=XB(IC)
           GO TO 300
      162  K1=IP+1
           K2=IP+MUL
           DO 163  I=K1,K2
      163  XJ(I-K1+1)=XB(I)
           K3=K2+1
           K4=K2+MUL
           DO 164 I=K3,K4
      164  AIC(I-K3+1)=XB(I)
           BT=XB(IC-3)
           CT=XB(IC-2)
           SIGT=XB(IC-1)
           SIG=XB(IC)
      300  CALL DERIVS(XB,XFB,GB)
           CSQ2=XFB
           GPB=0.0
           DO 310 I=1,IC
      310  GPB=GPB+GB(I)*S(I)
           IF(XFB.LE.EPS) GO TO 540
           IF(GPB.LT.0.0.OR.ABS((XFD-XFB)/XFD).LT.0.001) GO TO 320
           ITT=ITT+1
           IF(IIT.GE.5) GO TO 320
```

```
      GAMMA=ALPHA
      XFD=XFB
      GO TO 230
  320 SIGY=0.0
      DO 330 I=1,IC
      GC(I)=GB(I)-GA(I)
  330 SIGY=SIGY+GC(I)*SMA(I)
      IF(SIGY.EQ.0.0) THEN
        SIG=SIG+STEP
        AI(IC)=SIG
        WRITE(11,4) SIG
        GO TO 10
      ENDIF
      SIGHY=0.0
      DO 350 I=1,IC
      S(I)=0.0
      DO 340 J=1,IC
  340 S(I)=S(I)+H(I,J)*GC(J)
  350 SIGHY=SIGHY+GC(I)*S(I)
      DO 360 I=1,IC
      DO 360 J=1,IC
  360 H(I,J)=H(I,J)+SMA(I)*SMA(J)/SIGY-S(I)*S(J)/SIGHY
      IF(ABS((CSQ1-CSQ2)/CSQ1).GT.0.001) GO TO 370
      IT=IT+1
      IF(IT-4) 380,380,540
  370 IT=0
  380 IF(IKNT.GT.100) GO TO 520
      ISS=0
      DO 390 I=1,IC
      GA(I)=GB(I)
      IF(ABS(GB(I)).LE.EPS) ISS=ISS+1
  390 XA(I)=XB(I)
      IF(ISS.EQ.IC) GO TO 540
      XFA=XFB
      K=II+1+MUL
      DO 400 I=K,IC
      IF(XA(I).LT.0.0) GO TO 500
  400 CONTINUE
      GO TO 60
  500 WRITE(11,5)
      WRITE(11,2) (XA(I),I=1,IC)
  510 ISTOP=1
      RETURN
  520 WRITE(11,6)
      DO 530 I=1,IC
  530 AF(I)=XB(I)
      IRPT=0
      RETURN
  540 DO 550 I=1,IC
      XA(I)=XB(I)
  550 AF(I)=XB(I)
      WRITE(11,7) IKNT
      WRITE(11,2) (GB(I),I=1,IC)
      CHIF=CSQ2/(NP-IC)
      IF(IVAR.LE.3) THEN
        DO 560 I=1,IC
  560   EAF(I,I)=SQRT(CHIF*H(I,I))
```

```
      170                 ELSE
      171                   DO 570 I=1,IC
      172                   DO 570 J=1,IC
      173             570     EAF(I,J)=CHIF*H(I,J)
      174                 ENDIF
      175                 WRITE(11,8) CSQ2,CHIF
      176                 RETURN
      177               4 FORMAT(//' SIGMA MODIFIED= ',F5.3)
      178               5 FORMAT(//1X,' ONE OF THE PHOTOPEAK PARAMETERS BECAME NEGATIVE',
      179                */' FIT TERMINATED'//)
      180               6 FORMAT(1X,' NO MINIMISATION AFTER 50 ITERATIONS ')
      181               7 FORMAT(1X,' FUNCTION MINIMUM',5X,' IKNT=',I3/1X,'GRADIENTS')
      182               8 FORMAT(1X,' FINAL CHISQR=',E14.6,5X,'CHISQR/(DEG.FREE)=',E14.6/)
      183                 END
```

--VARIABLE MAP--(LO=A)

-NAME-	--ADDRESS--	BLOCK	-----PROPERTIES-------	-------TYPE---------	SIZE
AF	178	//		REAL	15
AI	0B	//		REAL	15
AIC	377B	//		REAL	5
ALPHA	2153B			REAL	
AT	1220B	//		REAL	
BGD	1116B	//		REAL	50
BT	1221B	//		REAL	
CHIF	2200B			REAL	
CSQ1	2126B			REAL	
CSQ2	2161B			REAL	
CT	1222B	//		REAL	
EAF	36B	//		REAL	225
EPS	2112B			REAL	
ETAA	2132B			REAL	
F	752B	//		REAL	50
GA	1436B			REAL	15
GAMMA	2145B			REAL	
GB	1455B			REAL	15
GC	1474B			REAL	15
GPA	2127B			REAL	
GPB	2147B			REAL	
H	1551B			REAL	225
I	2115B			INTEGER	
IC	1210B	//		INTEGER	
II	1206B	//		INTEGER	
IIT	2151B			INTEGER	
IKNT	2117B			INTEGER	
IN	1217B	//		INTEGER	
IP	1207B	//		INTEGER	
IRPT	1215B	//		INTEGER	
ISS	2173B			INTEGER	
ISTOP	1216B	//		INTEGER	

-NAME-	--ADDRESS--	BLOCK	-----PROPERTIES-------	-------TYPE---------	SIZE
IT	2120B			INTEGER	
ITT	2163B			INTEGER	
IVAR	1214B	//		INTEGER	
IXX	2113B			INTEGER	
J	2122B			INTEGER	
K	2144B			INTEGER	
K1	2134B			INTEGER	
K2	2136B			INTEGER	
K3	2141B			INTEGER	
K4	2142B			INTEGER	
MUL	1212B	//		INTEGER	
NP	1211B	//		INTEGER	
RD	404B	//		REAL	75
RIN	517B	//		REAL	5
S	1513B			REAL	15
SIG	1213B	//		REAL	
SIGHY	2166B			REAL	
SIGT	1223B	//		REAL	
SIGY	2164B			REAL	
SMA	1532B			REAL	15
STEP	2114B			REAL	
W	670B	//		REAL	50
WRES	1034B	//		REAL	50
X	524B	//		REAL	50
XA	1400B			REAL	15
XB	1417B			REAL	15
XFA	2125B			REAL	
XFB	2146B			REAL	
XFD	2152B			REAL	
XJ	1200B			REAL	5
Y	606B			REAL	50

```
      SUBROUTINE DERIVS(XX,CHISQ,G)
      DIMENSION EXPF(50,5),XX(15),G(15),FMODG(50,5),ZE(50,5)
      COMMON AI(15),AF(15),EAF(15,15),AIC(5),RD(5,15),RIN(5)
      COMMON X(50),Y(50),W(50),F(50),WRES(50),BGD(50),XJ(6)
      COMMON II,IP,IC,NP,MUL,SIG,IVAR,IRPT,ISTOP,IN
      COMMON AT,BT,CT,SIGT
      IF(IVAR.LE.3) GO TO 1001
 1003 DO 1006 J=1,MUL
      DO 1006 I=1,NP
      IF(X(I).LE.CT*XJ(J)) GO TO 1004
 1005 FMODG(I,J)=0.
      ZE(I,J)=0.0
      GO TO 1006
 1004 FMODG(I,J)=ETS(BT*(X(I)-XJ(J)))*(1.-ETS(-(X(I)-CT*XJ(J))**2
     */(2.*SIGT*SIGT)))
      ZE(I,J)=1.0
 1006 EXPF(I,J)=ETS(-(X(I)-XJ(J))**2/(2*SIG**2))
      GO TO 1008
 1001 DO 1002 J=1,MUL
      DO 1002 I=1,NP
      FMODG(I,J)=0.0
 1002 EXPF(I,J)=ETS(-(X(I)-XJ(J))**2/(2*SIG**2))
 1008 CHISQ=0.0
      DO 1020 I=1,NP
      F(I)=0.0
      DO 1010 K=1,II
 1010 F(I)=F(I)+XX(K)*(X(I)**(K-1))
      IF(IVAR.LE.4) GO TO 1011
      DO 1013 J=1,MUL
      K=II+J
      ATF=(X(I)-XJ(J))/SIG
      CALL ERF(ATF,ARF)
 1013 F(I)=F(I)+XX(K)*(0.5-ARF)
      DO 1014 J=1,MUL
      L=IN+J
 1014 F(I)=F(I)+XX(L)*EXPF(I,J)+AIC(J)*FMODG(I,J)
      GO TO 1016
 1011 DO 1015 J=1,MUL
      L=II+J
 1015 F(I)=F(I)+XX(L)*EXPF(I,J)+AIC(J)*FMODG(I,J)
 1016 WRES(I)=W(I)*(F(I)-Y(I))
 1020 CHISQ=CHISQ+W(I)*(F(I)-Y(I))**2
      DO 1025 K=1,II
      G(K)=0.0
      DO 1025 I=1,NP
 1025 G(K)=G(K)+2*WRES(I)*(X(I)**(K-1))
      IF(IVAR.LE.4) GO TO 1026
      DO 1028 J=1,MUL
      L=II+J
      G(L)=0.0
      DO 1028 I=1,NP
      ATF=(X(I)-XJ(J))/SIG
      CALL ERF(ATF,ARF)
 1028 G(L)=G(L)+WRES(I)*(1.0-2.*ARF)
      DO 1029 J=1,MUL
```

```
      L=IN+J
      G(L)=0.0
      DO 1029 I=1,NP
 1029 G(L)=G(L)+2*WRES(I)*EXPF(I,J)
      GO TO 1031
 1026 DO 1030 J=1,MUL
      L=II+J
      G(L)=0.0
      DO 1030 I=1,NP
 1030 G(L)=G(L)+2.*WRES(I)*EXPF(I,J)
 1031 IF(IVAR-3) 1035,1035,1060
 1035 IF(IVAR-2) 1050,1040,1040
 1040 DO 1045 J=1,MUL
      L=II+J
      K=IP+J
      G(K)=0.0
      DO 1045 I=1,NP
 1045 G(K)=G(K)+2*WRES(I)*XX(L)*EXPF(I,J)*((X(I)-XJ(J))/SIG**2)
      IF(IVAR.NE.3) GO TO 1050
      G(IC)=0.0
      DO 1047 I=1,NP
      DO 1047 J=1,MUL
      L=II+J
 1047 G(IC)=G(IC)+2*WRES(I)*XX(L)*EXPF(I,J)*((X(I)-XJ(J))**2/SIG**3)
 1050 GO TO 100
 1060 DO 1065 J=1,MUL
      IM=II+J
      L=IN+J
      K=IP+J
      G(K)=0.0
      DO 1065 I=1,NP
      G(K)=G(K)+2*WRES(I)*(XX(L)*EXPF(I,J)*((X(I)-XJ(J))/SIG**2)+
     &AIC(J)*ZE(I,J)*((-BT)*FMODG(I,J)+CT*((X(I)-CT*XJ(J))/(SIGT**2))*
     &(FMODG(I,J)-ETS(BT*(X(I)-XJ(J)))))
      IF(IVAR.EQ.5)G(K)=G(K)+2*WRES(I)*EXPF(I,J)*XX(IM)/SIG/2.506628274
 1065 CONTINUE
      DO 1075 J=1,MUL
      M=IP+MUL+J
      G(M)=0.0
      DO 1075 I=1,NP
 1075 G(M)=G(M)+2*WRES(I)*FMODG(I,J)
      G(IC-3)=0.0
      DO 1077 I=1,NP
      DO 1077 J=1,MUL
 1077 G(IC-3)=G(IC-3)+2*WRES(I)*AIC(J)*(X(I)-XJ(J))*FMODG(I,J)
      G(IC-2)=0.0
      DO 1082 I=1,NP
      DO 1082 J=1,MUL
 1082 G(IC-2)=G(IC-2)+2*WRES(I)*AIC(J)*(1./SIGT**2)*(X(I)-CT*XJ(J))*
     &(FMODG(I,J)-ETS(BT*(X(I)-XJ(J))))*ZE(I,J)*XJ(J)
      G(IC-1)=0.0
      DO 1085 I=1,NP
      DO 1085 J=1,MUL
 1085 G(IC-1)=G(IC-1)+2*WRES(I)*AIC(J)*((X(I)-CT*XJ(J))**2)*
     &(1./SIGT**3)*(FMODG(I,J)-ETS(BT*(X(I)-XJ(J))))*ZE(I,J)
      G(IC)=0.0
      DO 1087 I=1,NP
```

```
      113                 DO 1087 J=1,MUL
      114                 L=IN+J
      115                 IM=II+J
      116                 G(IC)=G(IC)+2*WRES(I)*(XX(L)*EXPF(I,J)*((X(I)-XJ(J))**2/SIG**3))
      117                 IF(IVAR.EQ.5)G(IC)=G(IC)+2*WRES(I)*EXPF(I,J)*XX(IM)/SIG**2/2.506628274
      118                &*(X(I)-XJ(J))
      119           1087  CONTINUE
      120             100 RETURN
      121                 END
```

--VARIABLE MAP--(LO=A)

-NAME-	--ADDRESS--	BLOCK-----	PROPERTIES-------	-TYPE---------	-SIZE
AF	17B	//		REAL	15
AI	0B	//		REAL	15
AIC	377B	//		REAL	5
ARF	3001B			REAL	
AT	1220B	//		REAL	
ATF	3000B			REAL	
BGD	1116B	//		REAL	50
BT	1221B	//		REAL	
CHISQ	2	DUMMY-ARG		REAL	
CT	1222B	//		REAL	
EAF	36B	//		REAL	225
EXPF	1410B			REAL	250
F	752B	//		REAL	50
FMODG	2002B			REAL	250
G	3	DUMMY-ARG		REAL	15
I	2770B			INTEGER	
IC	1210B	//		INTEGER	
II	1206B	//		INTEGER	
IM	3022B			INTEGER	
IN	1217B	//		INTEGER	
IP	1207B	//		INTEGER	

-NAME-	--ADDRESS--	BLOCK-----	PROPERTIES-------	-TYPE---------	-SIZE
IRPT	1215B	//		INTEGER	
ISTOP	1216B	//		INTEGER	
IVAR	1214B	//		INTEGER	
J	2766B			INTEGER	
K	2775B			INTEGER	
L	3003B			INTEGER	
M	3025B			INTEGER	
MUL	1212B	//		INTEGER	
NP	1211B	//		INTEGER	
RD	404B	//		INTEGER	
RIN	517B	//		REAL	75
SIG	1213B	//		REAL	5
SIGT	1223B	//		REAL	
W	670B	//		REAL	
WRES	1034B	//		REAL	50
X	524B	//		REAL	50
XJ	1200B	//		REAL	50
XX	1	DUMMY-ARG		REAL	6
Y	606B	//		REAL	15
ZE	2374B			REAL	50
				REAL	250

--PROCEDURES--(LO=A)

-NAME-	-TYPE--------	ARGS------	CLASS-----
ERF		2	SUBROUTINE
ETS	REAL	1	FUNCTION

--STATEMENT LABELS--(LO=A)

-LABEL-	-ADDRESS-----	PROPERTIES----	DEF
100	1361B		120
1001	111B		19
1002	INACTIVE	DO-TERM	22
1003	*NO REFS*		8
1004	40B		14
1005	*NO REFS*		11
1006	66B	DO-TERM	17
1008	151B		23
1010	INACTIVE	DO-TERM	27
1011	266B		38
1013	INACTIVE	DO-TERM	33
1014	INACTIVE	DO-TERM	36
1015	INACTIVE	DO-TERM	40
1016	314B		41
1020	INACTIVE	DO-TERM	42
1025	INACTIVE	DO-TERM	46
1026	506B		61
1028	INACTIVE	DO-TERM	54
1029	INACTIVE	DO-TERM	59
1030	INACTIVE	DO-TERM	65
1031	547B		66
1035	INACTIVE		67
1040	INACTIVE		68
1045	INACTIVE	DO-TERM	73
1047	INACTIVE	DO-TERM	79
1050	700B		80
1060	701B		81
1065	INACTIVE	DO-TERM	91
1075	INACTIVE	DO-TERM	96
1077	INACTIVE	DO-TERM	100

```
SUBROUTINE CUBINT      74/176  OPT=0,ROUND=  A/ S/ M/-D,-DS      FTN 5.1+577          84/08/10. 11.44.44      PAGE     1
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I=AD,L=L,B=ADB.

      1                 SUBROUTINE CUBINT(X1,X2,GAMMA,G1,G2,A)
      2                 Z=3.0*((X1-X2)/GAMMA)+G2+G1
      3                 W=Z*Z-G2*G1
      4                 IF(W.GE.0) THEN
      5                   W=SQRT(W)
      6                   A=GAMMA*(1.0-((G2+W-Z)/(G2-G1+W+W)))
      7                 ELSE
      8                   WRITE(11,10)
      9                   A=0.0
     10                 ENDIF
     11                 RETURN
     12              10 FORMAT(1X,' W IS COMPLEX , EXECUTION ENDS...')
     13                 END

--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE      -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  A              6    DUMMY-ARG                    REAL              W            66B                            REAL
  GAMMA          3    DUMMY-ARG                    REAL              X1            1    DUMMY-ARG               REAL
  G1             4    DUMMY-ARG                    REAL              X2            2    DUMMY-ARG               REAL
  G2             5    DUMMY-ARG                    REAL              Z            65B                            REAL

--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  SQRT       GENERIC         1       INTRINSIC

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

      10      52B      FORMAT          12

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  CUBINT       5B      6

--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE11   FMT/SEQ
```

```
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I=AD,L=L,B=ADB.

      1                SUBROUTINE RINT
      2                COMMON AI(15),AF(15),EAF(15,15),AIC(5),RD(5,15),RIN(5)
      3                COMMON X(50),Y(50),W(50),F(50),WRES(50),BGD(50),XJ(6)
      4                COMMON II,IP,IC,NP,MUL,SIG,IVAR,IRPT,ISTOP,IN
      5                COMMON AT,BT,CT,SIGT
      6                DO 10 J=1,MUL
      7                ARG1=BT*SIGT
      8                ARG2=(CT*XJ(J)/SIGT)+BT*SIGT
      9                CALL ERF(ARG1,RES1)
     10                CALL ERF(ARG2,RES2)
     11                RIN(J)=(AIC(J)/BT)*EXP(-BT*XJ(J))*(EXP(BT*CT*XJ(J))-1.)+
     12               &SQRT(2.*3.1415926)*AIC(J)*SIGT*EXP(0.5*((BT*SIGT)**2)+BT*XJ(J)*
     13               &(CT-1.))*(RES1-RES2)
     14                RD(J,3)=RIN(J)/AIC(J)
     15                RD(J,2)=(AIC(J)/BT)*(BT*(CT-1.)*EXP(BT*XJ(J)*(CT-1.))+BT*
     16               &EXP(-BT*XJ(J)))+BT*(CT-1)*(RIN(J)-(AIC(J)/BT)*EXP(-BT*
     17               &XJ(J))*(EXP(BT*CT*XJ(J))-1.))-AIC(J)*CT*EXP(-BT*XJ(J)-.5*((CT*
     18               &XJ(J)/SIGT)**2))
     19                RD(J,IC-3)=-(AIC(J)/BT**2)*EXP(-BT*XJ(J))*(EXP(BT*CT*XJ(J))-1.)+
     20               &(AIC(J)/BT)*EXP(-BT*XJ(J))*(XJ(J)*(CT-1.)*EXP(BT*XJ(J)*CT)+
     21               &XJ(J))+(BT*(SIGT**2)+XJ(J)*(CT-1.))*(RIN(J)-(AIC(J)/BT)*EXP(-BT*
     22               &XJ(J))*(EXP(BT*CT*XJ(J))-1.))+AIC(J)*(SIGT**2)*EXP(BT*XJ(J)*
     23               &(CT-1))-(AIC(J)*SIGT**2)*EXP(-0.5*((CT*XJ(J)/SIGT)**2)-BT*XJ(J))
     24                RD(J,IC-2)=RIN(J)*BT*XJ(J)+AIC(J)*XJ(J)*EXP(-BT*XJ(J))*
     25               &(1.-EXP(-0.5*(CT*XJ(J)/SIGT)**2))
     26           10   RD(J,IC-1)=(1+(BT*SIGT)**2)*(1./SIGT)*(RIN(J)-(AIC(J)/BT)*EXP
     27               &(-BT*XJ(J))*(EXP(BT*CT*XJ(J))-1.))+AIC(J)*BT*SIGT*EXP(BT*XJ(J)
     28               &*(CT-1.))-AIC(J)*SIGT*(BT-(CT*XJ(J)/SIGT**2))*EXP(-0.5*((CT*
     29               &XJ(J)/SIGT)**2)-BT*XJ(J))
     30                RETURN
     31                END
```

--VARIABLE MAP--(LO=A)

-NAME-	--ADDRESS--	BLOCK	-----PROPERTIES-------	-------TYPE---------	SIZE
AF	178	//		REAL	15
AI	0B	//		REAL	15
AIC	3778	//		REAL	5
ARG1	455B			REAL	
ARG2	456B			REAL	
AT	12208	//		REAL	
BGD	11168	//		REAL	50
BT	12218	//		REAL	
CT	12228	//		REAL	
EAF	36B	//		REAL	225
F	7528	//		REAL	50
IC	12108	//		INTEGER	
II	12068	//		INTEGER	
IN	12178	//		INTEGER	
IP	12078	//		INTEGER	
IRPT	12158	//		INTEGER	
ISTOP	12168	//		INTEGER	
IVAR	12148	//		INTEGER	
J	453B			INTEGER	
MUL	12128	//		INTEGER	
NP	12118	//		INTEGER	
RD	404B	//		REAL	75
RES1	457B			REAL	
RES2	460B			REAL	
RIN	517B	//		REAL	5
SIG	12138	//		REAL	
SIGT	12238	//		REAL	
W	670B	//		REAL	50
WRES	10348	//		REAL	50
X	524B	//		REAL	50
XJ	12008	//		REAL	6
Y	606B	//		REAL	50

DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I=AD,L=L,B=ADB.

```
1                SUBROUTINE ERF(Z,RES)
2                ZE=ABS(Z*0.70710678)
3                DE=(0.8540765+ZE*(0.30781819+ZE*(0.063832389-ZE*
4               &0.00018240507)))/(1.0+ZE*(0.65097426+ZE*(0.22948482+ZE*
5               &0.0340301B2)))
6                TE=1.7724539+ZE*(2.0-DE)
7                QU=EXP(-ZE**2)*TE/(1.7724539*(ZE*TE+1.0))
8                C=-0.5
9                IF(Z.LT.0.0) C=0.5
10               RES=(QU-1.0)*C
11               RETURN
12               END
```

--VARIABLE MAP--(LO=A)

-NAME-	--ADDRESS--	--BLOCK-----	----PROPERTIES-------	----TYPE---------	--SIZE
C	102B			REAL	
DE	77B			REAL	
QU	101B			REAL	
RES	2	DUMMY-ARG		REAL	
TE	100B			REAL	
Z	1	DUMMY-ARG		REAL	
ZE	76B			REAL	

--PROCEDURES--(LO=A)

-NAME------	----TYPE--------	ARGS------	--CLASS-----
ABS	GENERIC	1	INTRINSIC
EXP	GENERIC	1	INTRINSIC

--ENTRY POINTS--(LO=A)

-NAME---	--ADDRESS--	ARGS---
ERF	5B	2

--STATISTICS--

PROGRAM-UNIT LENGTH	105B =	69
SCM STORAGE USED	61700B =	25536
COMPILE TIME	0.192 SECONDS	

```
  FUNCTION ETS          74/176  OPT=0,ROUND= A/ S/ M/-D,-DS      FTN 5.1+577          84/08/10. 11.44.44        PAGE      1
DC=-LONG/-DT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I=AD,L=L,B=ADB.

      1                FUNCTION ETS(X)
      2                IF(X.LE.-675.0) THEN
      3                  ETS=0.0
      4                ELSE
      5                  ETS=EXP(X)
      6                ENDIF
      7                RETURN
      8                END


--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  ETS         278                             REAL
  X             1   DUMMY-ARG                 REAL


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  EXP        GENERIC         1       INTRINSIC


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  ETS          60      1


--STATISTICS--

  PROGRAM-UNIT LENGTH              328 =  26
  SCM STORAGE USED               61700B =  25536
  COMPILE TIME                   0.073 SECONDS
12.05.22.UCLP, AA, P04     ,       1.194KLNS.
```

```
      PROGRAM CALIBR (INPUT,OUTPUT,CIK,TAPE5=INPUT,TAPE6=OUTPUT,
     &TAPE11=CIK)
C*************************************************************************
C*     PROGRAM BUCAASA/CALIBR                                            *
C*     AUTHOR : DR. OZER CIFTCIOGLU, ITU (1979)                          *
C*     REVISED AND DEVELOPED BY: LEVENT AKIN, BU (1984)                  *
C*************************************************************************
      DIMENSION PK(150),ENGY(150),EENGY(150),EN(15),EF(15),ENL(15),
     *EFL(15),EFL2(150),AREA(150),EAREA(150),EPK(150),EEF(150),ERL(150)
      DIMENSION RCCP(50),ACE(50),W(50),C(50),CE(10),XI(10,10),WE(50)
      CHARACTER*7 NAM,NAM2
      CHARACTER*10 TIME,DATE,GUN,SAAT
      CHARACTER*50 JOB
      LOGICAL JXST
      GUN=DATE()
      SAAT=TIME()
      WRITE(6,*) 'NAME OF JOB?'
      READ(5,75)JOB
   75 FORMAT(A50)
      WRITE(11,80) GUN,SAAT,JOB
   80 FORMAT(1H1,6('*'),' BUCAASA/CALIBR REV 3.1 ',2A10,6('*')//
     &5X,'JOB: ',A50)
      WRITE(6,*)'NP,MP,SYSER'
      READ(5,*)NP,MP,SYSER
      WRITE(6,*)'ENTER CALIBRATION DATA'
      WRITE(6,*)'CHANNEL ENERGY'
      DO 10 I=1,NP
      READ(5,*) RCCP(I),ACE(I)
   10 W(I)=1.
      WRITE(11,3)
    3 FORMAT(10X,'ENERGY CALIBRATION DATA'/17X,'CHANNEL',6X,
     &'ENERGY(KEV)'/)
      DO 15 I=1,NP
   15 WRITE(11,4) I,RCCP(I),ACE(I)
    4 FORMAT(10X,I2,5X,F8.3,5X,F8.3)
      WRITE(6,*)'LP SYSEF EFIN'
      READ(5,*) LP,SYSEF,EFIN
      IF(EFIN.EQ.1) THEN
      WRITE(6,*)'ENERGY EFFICIENCY EFF-ERR'
      DO 24 I=1,LP
   24 READ(5,*) EN(I),EF(I),ENL(I)
      WRITE(11,1)
    1 FORMAT(//10X,'EFFICIENCY CALIBRATION DATA'/14X,'ENERGY(KEV)',
     &5X,'EFFICIENCY',10X,'EFF-ERR'/)
      DO 22 I=1,LP
   22 WRITE(11,2) I,EN(I),EF(I),ENL(I)
    2 FORMAT(10X,I2,5X,F8.3,5X,1PE14.7,3X,1PE14.7)
      ELSE
      WRITE(6,*)'ENTER CALIBRATION DATA'
      WRITE(6,*)'ENERGY  EFFICIENCY'
      DO 25 I=1,LP
      READ(5,*) EN(I),EF(I)
      ENL(I)=ALOG(EN(I))
      EFL(I)=ALOG(EF(I))
   25 WE(I)=1.
```

```
      WRITE(11,5)
    5 FORMAT(//10X,'EFFICIENCY CALIBRATION DATA'/14X,'ENERGY(KEV)',
     &5X,'EFFICIENCY'/)
      DO 17 I=1,LP
   17 WRITE(11,6) I,EN(I),EF(I)
    6 FORMAT(10X,I2,5X,F8.3,5X,1PE14.7)
      ENDIF
      WRITE(6,*)'NAME OF THE FIT FILE?'
   55 READ(5,82)NAM
   82 FORMAT(A7)
      INQUIRE(FILE=NAM,EXIST=JXST)
      IF(.NOT.JXST) THEN
        WRITE(6,*)'THERE IS NO SUCH FILE !'
        GO TO 55
      ENDIF
      OPEN(17,FILE=NAM,ACCESS='DIRECT',RECL=30)
      WRITE(11,*)'DATA INPUT FROM FILE ',NAM
      WRITE(6,*)'NAME OF THE TRANSFER FILE?'
      READ(5,82)NAM2
      OPEN(15,FILE=NAM2)
      READ(17,REC=1)NX
      MX=1
      DO 68 I=2,NX+1
      READ(17,REC=I)MUL,(PK(J),EPK(J),AREA(J),EAREA(J),
     &J=MX,MX+MUL-1)
   68 MX=MX+MUL
      CALL POLIN(NP,MP,RCCP,ACE,W,C,ENGY,XI,ES,CE)
      M=MP+1
      IF(MP.LE.0) STOP
      CALL POLIX(M,MX,PK,C,XI,ENGY,EENGY,ES,1)
      DO 32 I=1,MX
   32 EENGY(I)=SQRT((EENGY(I)*ENGY(I)*0.01)**2+SYSER**2+
     &(EPK(I)/PK(I)*ENGY(I))**2)
      IF(EFIN.EQ.1) THEN
         CALL INTRP(EN,EF,ENL,ENGY,EFL2,EEF,EAREA,LP,MX,SYSEF)
         GO TO 100
      ENDIF
      CALL POLIN(LP,1,ENL,EFL,WE,C,EFL2,XI,ES,CE)
      DO 99 I=1,MX
   99 ERL(I)=ALOG(ENGY(I))
      CALL POLIX(2,MX,ERL,C,XI,EFL2,EEF,ES,2)
      DO 60 I=1,MX
      EFL2(I)=EXP(EFL2(I))
      EEF(I)=SQRT((EFL2(I)*EXP(EEF(I))*0.01)**2+SYSEF**2)
   60 CONTINUE
  100 WRITE(11,7)
      WRITE(15,84) MX
   84 FORMAT(I3)
    7 FORMAT(//5X,'CALIBRATION RESULTS'/' PEAK#  CHANNEL  ERROR-CHN'
     *,'  ENERGY(KEV)  ERROR(KEV)     AREA     ERROR(%)  INTENSITY',
     *'   ERROR(%)'/)
      DO 78 I=1,MX
      WRITE(11,8)I,PK(I),EPK(I),ENGY(I),EENGY(I),AREA(I),EAREA(I),
     &AREA(I)/EFL2(I),EEF(I)
    8 FORMAT(1X,I3,3X,F8.3,3X,F5.3,3X,F9.3,8X,F5.3,3X,E11.4,3X,F7.3,
     &3X,E11.4,3X,F7.3)
      WRITE(15,85)PK(I),ENGY(I),AREA(I)/EFL2(I),EEF(I)
```

```
      85  FORMAT(4E14.7)
   78     CONTINUE
          STOP
          END
```

--VARIABLE MAP--(LO=A)

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
ACE	4217B			REAL	50
AREA	2557B			REAL	150
C	4363B			REAL	50
CE	4445B			REAL	10
EAREA	3005B			REAL	150
EEF	3461B			REAL	150
EENGY	2007B			REAL	150
EF	2254B			REAL	15
EFIN	4727B			REAL	
EFL	2312B			REAL	15
EFL2	2331B			REAL	150
EN	2235B			REAL	15
ENGY	1561B			REAL	150
ENL	2273B			REAL	15
EPK	3233B			REAL	150
ERL	3707B			REAL	150
ES	4742B			REAL	
GUN	4707B			CHAR*10	
I	4722B			INTEGER	
J	4740B			INTEGER	
JOB	4711B			CHAR*50	
JXST	4716B			LOGICAL	
LP	4725B			INTEGER	
M	4743B			INTEGER	
MP	4720B			INTEGER	
MUL	4737B			INTEGER	
MX	4735B			INTEGER	
NAM	4705B			CHAR*7	
NAM2	4706B			CHAR*7	
NP	4717B			INTEGER	
NX	4734B			INTEGER	
PK	1333B			REAL	150
RCCP	4135B			REAL	50
SAAT	4710B			CHAR*10	
SYSEF	4726B			REAL	
SYSER	4721B			REAL	
W	4301B			REAL	50
WE	4623B			REAL	50
XI	4457B			REAL	100

--PROCEDURES--(LO=A)

NAME	TYPE	ARGS	CLASS
ALOG	REAL	1	INTRINSIC
DATE	CHAR*10	0	FUNCTION
EXP	GENERIC	1	INTRINSIC
INTRP		10	SUBROUTINE
POLIN		10	SUBROUTINE
POLIX		9	SUBROUTINE
SQRT	GENERIC	1	INTRINSIC
TIME	CHAR*10	0	FUNCTION

--STATEMENT LABELS--(LO=A)

LABEL	ADDRESS	PROPERTIES	DEF
1	672B	FORMAT	43
2	704B	FORMAT	47
3	656B	FORMAT	31
4	666B	FORMAT	35
5	711B	FORMAT	57
6	722B	FORMAT	61
7	732B	FORMAT	104
8	751B	FORMAT	110
10	INACTIVE	DO-TERM	29
15	INACTIVE	DO-TERM	34
17	INACTIVE	DO-TERM	60
22	INACTIVE	DO-TERM	46
24	INACTIVE	DO-TERM	41
25	INACTIVE	DO-TERM	55
32	INACTIVE	DO-TERM	87
55	2718		64
60	INACTIVE	DO-TERM	100
68	INACTIVE	DO-TERM	81
75	644B	FORMAT	19
78	INACTIVE	DO-TERM	114
80	646B	FORMAT	21
82	726B	FORMAT	65
84	730B	FORMAT	103
85	762B	FORMAT	113
99	INACTIVE	DO-TERM	95
100	504B		101

```
      SUBROUTINE POLIN(N,M,X,Y,W,A,YC,D,S,EC)
      DIMENSION X(100),Y(100),YC(100),W(100),XC(20),YX(10),A(10),B(10)
      DIMENSION C(10,10),D(10,10),EC(100)
      MP1=M+1
      M2=M+M
      DO 20 K=1,M2
      XC(K)=0.0
      DO 15 I=1,N
   15 XC(K)=XC(K)+(W(I)*(X(I)**K))
   20 CONTINUE
      YX1=0.0
      WI=0.0
      DO 22 I=1,N
      WI=WI+W(I)
   22 YX1=YX1+W(I)*Y(I)
      DO 30 K=1,M
      YX(K)=0.0
      DO 25 I=1,N
   25 YX(K)=YX(K)+(W(I)*(Y(I)*(X(I)**K)))
   30 CONTINUE
      DO 40 I=1,MP1
      DO 35 J=1,MP1
      IPJM2=I+J-2
      IF(IPJM2.EQ.0) THEN
        C(1,1)=WI
      ELSE
        C(I,J)=XC(IPJM2)
      ENDIF
   35 CONTINUE
   40 CONTINUE
      B(1)=YX1
      DO 45 I=2,MP1
   45 B(I)=YX(I-1)
      CALL MATIN(C,D,MP1)
      IF(MP1.LE.0) THEN
         M=0
         RETURN
      ENDIF
      DO 55 I=1,MP1
      A(I)=0.0
      DO 54 J=1,MP1
   54 A(I)=A(I)+D(I,J)*B(J)
   55 CONTINUE
      DO 60 I=1,N
      YC(I)=0.
      DO 60 J=1,MP1
   60 YC(I)=YC(I)+(A(J)*(X(I)**(J-1)))
      D1=N-MP1
      S=0
      SDEV=0.0
      DO 70 I=1,N
      DEV=Y(I)-YC(I)
      S=S+W(I)*(DEV**2)
   70 SDEV=SDEV+DEV
      S=S/D1
```

```
      56                    VE=SQRT(S)
      57                    DO 80 J=1,MP1
      58                    T=D(J,J)*S
      59                 80 EC(J)=SQRT(T)
      60                    RETURN
      61                    END
```

--VARIABLE MAP--(LO=A)

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
A	6	DUMMY-ARG		REAL	10
B	461B			REAL	10
C	473B			REAL	100
D	8	DUMMY-ARG		REAL	100
DEV	666B			REAL	
D1	663B			REAL	
EC	10	DUMMY-ARG		REAL	100
I	643B			INTEGER	
IPJM2	655B			INTEGER	
J	653B			INTEGER	
K	641B			INTEGER	
M	2	DUMMY-ARG		INTEGER	
MP1	637B			INTEGER	
M2	640B			INTEGER	
N	1	DUMMY-ARG		INTEGER	
S	9	DUMMY-ARG		REAL	
SDEV	664B			REAL	
T	671B			REAL	
VE	667B		*S*	REAL	
W	5	DUMMY-ARG		REAL	100
WI	646B			REAL	
X	3	DUMMY-ARG		REAL	100
XC	423B			REAL	20
Y	4	DUMMY-ARG		REAL	100
YC	7	DUMMY-ARG		REAL	100
YX	447B			REAL	10
YX1	645B			REAL	

--PROCEDURES--(LO=A)

NAME	TYPE	ARGS	CLASS
MATIN		3	SUBROUTINE
SQRT	GENERIC	1	INTRINSIC

--STATEMENT LABELS--(LO=A)

LABEL	ADDRESS	PROPERTIES	DEF
15	INACTIVE	DO-TERM	9
20	INACTIVE	DO-TERM	10
22	INACTIVE	DO-TERM	15
25	INACTIVE	DO-TERM	19
30	INACTIVE	DO-TERM	20
35	INACTIVE	DO-TERM	29
40	INACTIVE	DO-TERM	30
45	INACTIVE	DO-TERM	33
54	INACTIVE	DO-TERM	42
55	INACTIVE	DO-TERM	43
60	INACTIVE	DO-TERM	47
70	INACTIVE	DO-TERM	54
80	INACTIVE	DO-TERM	59

--ENTRY POINTS--(LO=A)

NAME	ADDRESS	ARGS
POLIN	5B	10

```
DG=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I=CC,L=LCAL,B=CALIBR.
```



```
      1                SUBROUTINE INTRP(XX,YY,YYE,X,Y,YE,EA,N,M,SYS)
      2                DIMENSION XX(N),YY(N),YYE(N),X(M),Y(M),YE(M),EA(M)
      3                DO 10 J=1,M
      4                DO 10 I=1,N-1
      5                IF(X(J).GT.XX(I).AND.X(J).LE.XX(I+1))THEN
      6                  Y(J)=EXP((ALOG(X(J))*ALOG(YY(I)/YY(I+1))+
      7               &      ALOG(XX(I))*ALOG(YY(I+1))-ALOG(XX(I+1))*ALOG(YY(I)))/
      8               &          ALOG(XX(I)/XX(I+1)))
      9                  YE(J)=EXP((ALOG(X(J))*ALOG(YYE(I)/YYE(I+1))+
     10               &      ALOG(XX(I))*ALOG(YYE(I+1))-ALOG(XX(I+1))*ALOG(YYE(I)))/
     11               &          ALOG(XX(I)/XX(I+1)))
     12                 ENDIF
     13                 YE(J)=SQRT(YE(J)**2+EA(J)**2+SYS**2)
     14           10    CONTINUE
     15                 RETURN
     16                 END
```

--VARIABLE MAP--(LO=A)

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
EA	7	DUMMY-ARG		REAL	ADJ-ARY
I	2358			INTEGER	
J	2338			INTEGER	
M	9	DUMMY-ARG		INTEGER	
N	8	DUMMY-ARG		INTEGER	
SYS	10	DUMMY-ARG		REAL	
X	4	DUMMY-ARG		REAL	ADJ-ARY
XX	1	DUMMY-ARG		REAL	ADJ-ARY
Y	5	DUMMY-ARG		REAL	ADJ-ARY
YE	6	DUMMY-ARG		REAL	ADJ-ARY
YY	2	DUMMY-ARG		REAL	ADJ-ARY
YYE	3	DUMMY-ARG		REAL	ADJ-ARY

--PROCEDURES--(LO=A)

NAME	TYPE	ARGS	CLASS
ALOG	REAL	1	INTRINSIC
EXP	GENERIC	1	INTRINSIC
SQRT	GENERIC	1	INTRINSIC

--STATEMENT LABELS--(LO=A)

LABEL	ADDRESS	PROPERTIES	DEF
10	INACTIVE	DO-TERM	14

--ENTRY POINTS--(LO=A)

NAME	ADDRESS	ARGS
INTRP	58	10

```
DO=-LONG/-UT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I=CC,L=LCAL,B=CALIBR.
```

```
      SUBROUTINE MATIN(A,B,N)
      DIMENSION A(10,10),B(10,10)
      DATA EPS/1.E-10/
      DO 6 I=1,N
      DO 6 J=1,N
    6 B(I,J)=0.0
      DO 7 I=1,N
    7 B(I,I)=1.0
      DEL=1.
      DO 45 K=1,N
      IF (K.GE.N) GO TO 30
      IMAX=K
      AMAX=ABS(A(K,K))
      KP1=K+1
      DO 20 I=KP1,N
      IF(AMAX.LT.ABS(A(I,K))) THEN
        IMAX=I
        AMAX=ABS(A(I,K))
      ENDIF
   20 CONTINUE
      IF(IMAX.EQ.K) GO TO 30
      DO 29 J=1,N
      ATMP=A(IMAX,J)
      A(IMAX,J)=A(K,J)
      A(K,J)=ATMP
      BTMP=B(IMAX,J)
      B(IMAX,J)=B(K,J)
   29 B(K,J)=BTMP
      DEL=-DEL
   30 CONTINUE
      IF(ABS(A(K,K)).LE.EPS) GO TO 93
      DEL=A(K,K)*DEL
      DIV=A(K,K)
      DO 38 J=1,N
      A(K,J)=A(K,J)/DIV
   38 B(K,J)=B(K,J)/DIV
      DO 43 I=1,N
      AMULT=A(I,K)
      IF(I.EQ.K) GO TO 43
      DO 42 J=1,N
      A(I,J)=A(I,J)-AMULT*A(K,J)
   42 B(I,J)=B(I,J)-AMULT*B(K,J)
   43 CONTINUE
   45 CONTINUE
      RETURN
   93 WRITE(6,*)'SINGULAR MATRIX EXECUTION ENDS'
      N=0
      RETURN
      END
```

```
FTN5,I=CC,L=LCAL,B=CALIBR.

      1                SUBROUTINE POLIX(M,K,XD,C,XI,YD,YE,S,IPIS)
      2                DIMENSION XD(150),YD(150),C(10),XI(10,10),YE(150),G(20)
      3                MM=M
      4                IF(IPIS.EQ.2)C(1)=EXP(C(1))
      5                DO 10 I=1,K
      6                YD(I)=0.0
      7                DO 10 J=1,MM
      8           10   YD(I)=YD(I)+C(J)*(XD(I)**(J-1))
      9                DO 30 I=1,K
     10                YE(I)=0.0
     11                DO 20 J=1,MM
     12           20   G(J)=XD(I)**(J-1)
     13                DO 30 L=1,MM
     14                DO 30 N=1,MM
     15           30   YE(I)=YE(I)+G(L)*XI(L,N)*G(N)
     16                DO 40 I=1,K
     17           40   YE(I)=SQRT(YE(I)*S)*100/YD(I)
     18                RETURN
     19                END
```

--VARIABLE MAP--(LO=A)

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
C	4	DUMMY-ARG		REAL	10
G	2008			REAL	20
I	2258			INTEGER	
IPIS	9	DUMMY-ARG		INTEGER	
J	2278			INTEGER	
K	2	DUMMY-ARG		INTEGER	
L	2338			INTEGER	
M	1	DUMMY-ARG		INTEGER	
MM	2248			INTEGER	
N	2358			INTEGER	
S	8	DUMMY-ARG		REAL	
XD	3	DUMMY-ARG		REAL	150
XI	5	DUMMY-ARG		REAL	100
YD	6	DUMMY-ARG		REAL	150
YE	7	DUMMY-ARG		REAL	150

--PROCEDURES--(LO=A)

NAME	TYPE	ARGS	CLASS
EXP	GENERIC	1	INTRINSIC
SQRT	GENERIC	1	INTRINSIC

--STATEMENT LABELS--(LO=A)

LABEL	ADDRESS	PROPERTIES	DEF
10	INACTIVE	DO-TERM	8
20	INACTIVE	DO-TERM	12
30	INACTIVE	DO-TERM	15
40	INACTIVE	DO-TERM	17

```
FTN5,I=IDEN,L=LID,B=NUCLIDE.

      PROGRAM NUCLIDE(INPUT,OUTPUT,CIK,TAPE6=OUTPUT,TAPE5=INPUT,
     &                TAPE11=CIK)
*************************************************************************
*     PROGRAM BUCAASA/NUCLIDE  (SAMPO80 PART III)                       *
*     AUTHOR : KOSKELO ET AL.                (1982)                     *
*     REVISED AND ADAPTED BY : LEVENT AKIN, BU (1984)                   *
*************************************************************************
      CHARACTER*7 INF,LIBF
      CHARACTER*50 JOB
      CHARACTER*10 TIME,DATE,GUN,SAAT
      LOGICAL JXST
      COMMON/TIMES/TCO,TWA,TEX,TLIVE,ETOL
      GUN=DATE()
      SAAT=TIME()
      WRITE(6,*)'NAME OF JOB  '
      READ(5,10)JOB
      WRITE(11,30)  GUN,SAAT,JOB
      WRITE(6,*) 'TCO TEX  TWA TLIVE ETOL '
      READ(5,*)TCO,TEX,TWA,TLIVE,ETOL
      IF(TCO.LE.0.0) TCO=1.0
      IF(TEX.LE.0.0) TEX=1.0
      IF(TWA.LE.0.0) TWA=1.0
      WRITE(11,40)TCO,TWA,TEX,TLIVE,ETOL
  100 WRITE(6,*)'NAME OF THE INPUT FILE?'
      READ(5,10)INF
      INQUIRE(FILE=INF,EXIST=JXST)
      IF(.NOT.JXST) THEN
         WRITE(6,*)'NO SUCH FILE'
         GO TO 100
      ENDIF
      OPEN(14,FILE=INF,RECL=80)
  200 WRITE(6,*)'NAME OF THE LIBRARY FILE?'
      READ(5,10)LIBF
      INQUIRE(FILE=INF,EXIST=JXST)
      IF(.NOT.JXST) THEN
        WRITE(6,*)'NO SUCH FILE'
        GO TO 200
      ENDIF
      OPEN(2,FILE=LIBF)
      CALL IDENT
      STOP
   10 FORMAT(A7)
   20 FORMAT(A50)
   30 FORMAT(1H1,1X,10('*'),' BUCAASA/NUCLIDE REV 1.0',2A10,
     &10('*')//'JOB : ',A50/)
   40 FORMAT(1X,'COUNT. TIME= ',F11.2,'(MIN)','  WAIT. TIME= ',
     &F11.2,'(MIN)',' EXSPSR. TIME= ',F11.2,' (MIN)',
     &'LIVE C. TIME= ',F11.2,'(SEC)'/1X,'EN. TOL= ',F4.2,'(KEV)')
      END
```

```
…COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I=IDEN,L=LID,B=NUCLIDE.

      SUBROUTINE CORR(JB,JA,AL2,NUMF,JG,ICHEC)
      CHARACTER*8 COL,ITUNN
      COMMON/FITIN/FITCH(100),ENERG(100),EFFSUM(100),EFER(100)
      COMMON/COR/B(80,30),ROW(30),X(30),IL(30),CONF(30),
     &HALF(30),ACOL5(30),XINT(30),XCONF(30),XER(30),
     &ACOL2(30),ACOL3(30)
      COMMON/TIMES/TCO,TWA,TEX,TLIVE,ETOL
      COMMON/INTF/A(80,30),JFL(30),KFL(30)
      COMMON COL(30),ITUNN(30)
      DO 183 K=1,JB
      KK=KFL(K)
      IF(JA.NE.1)GO TO 1801
      JJ=JFL(1)
      IF(EFER(JJ).GT.50.0.OR.EFFSUM(JJ).LE.0.0)GO TO 176
      ERROR=EFER(JJ)
      GC TO 1198
 1801 ERROR=100.*SQRT(B(K,K))/X(K)
 1198 CCOR=(1.-EXP(-AL2*TCO/HALF(KK)))/(AL2*TCO/HALF(KK))
      TCHECK=365.*24.*60.
      IF(HALF(KK).GT.TCHECK) CCOR=1.
      WCOR=EXP(-AL2*TWA/HALF(KK))
      WTEST=AL2*TWA/HALF(KK)
      IF(WTEST.GT.0.0001) GO TO 591
      KERR=1.
      WCOR=0.
      DO 6311 IND=1,5
      KERR=KERR*FLOAT(IND)
      WCOR=WCOR+(-WTEST)**IND/KERR
 6311 CONTINUE
      WCOR=1.-WCOR
  591 CONTINUE
      ECOR=1.-EXP(-AL2*TEX/HALF(KK))
      ETEST=AL2*TEX/HALF(KK)
      IF(ETEST.GT.0.0001) GO TO 6120
      KERR=1.
      ECOR=0.
      DO 6119 IND=1,5
      KERR=KERR*FLOAT(IND)
      ECOR=ECOR-(-ETEST)**IND/KERR
 6119 CONTINUE
 6120 IF(JA.NE.1)GO TO 293
      XE=EFFSUM(JJ)/(A(JJ,KK)*TLIVE*3.7E-02*WCOR*CCOR)
      ASAT=XE/ECOR
      IF(JB.EQ.1)GO TO 210
      WRITE(11,1116) COL(KK)
      GO TO 209
 293  XE=X(K)/(WCOR*CCOR*TLIVE*3.7E-02)
      ASAT=XE/ECOR
 210  WRITE(11,1013)COL(KK)
 209  DO 1181 I=1,NUMF
      IF(A(I,KK).LT.1.E-36)GO TO 1181
      WRITE(11,1014)I,FITCH(I),ENERG(I)
 1181 CONTINUE
      CALL ORDER(6,JG,ACOL2(KK),ACOL3,XE,XINT,CONF(KK),XCONF,
     1ERROR,XER,ASAT,ACOL5,COL(KK),ITUNN)
```

```
      56            183  CONTINUE
      57                 RETURN
      58             176 ICHEC=1
      59                 RETURN
      60            1013 FORMAT("THE FOLLOWING PEAKS ARE ATTRIBUTED TO ISOTOPE",2X,
      61           &A8,/1H ,5X,6HNUMBER,5X,7HCHANNEL,5X,6HENERGY)
      62            1014 FORMAT(1H ,6X,I4,6X,F7.2,4X,F7.2)
      63            1116 FORMAT("THE FOLLOWING PEAKS ARE ATTRIBUTED TO ISOTOPE",2X,A8,
      64                1'      ** CHECK **',/1H ,5X,6HNUMBER,5X,7HCHANNEL,5X,6HENERGY)
      65                 END
```

--VARIABLE MAP--(LO=A)

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
A	0B	/INTF/		REAL	2400
ACOL2	51568	/COR/		REAL	30
ACOL3	52148	/COR/		REAL	30
ACOL5	47668	/COR/		REAL	30
AL2	3	DUMMY-ARG		REAL	
ASAT	4558			REAL	
B	0B	/COR/		REAL	2400
CCOR	4428			REAL	
COL	0B	//		CHAR*8	30
CONF	46728	/COR/		REAL	30
ECOR	4518			REAL	
EFER	4548	/FITIN/		REAL	100
EFFSUM	3108	/FITIN/		REAL	100
ENERG	1448	/FITIN/		REAL	100
ERROR	4418			REAL	
ETEST	4528			REAL	
ETOL	48	/TIMES/		REAL	
FITCH	0B	/FITIN/		REAL	100
HALF	47308	/COR/		REAL	30
I	4568			INTEGER	
ICHEC	6	DUMMY-ARG		INTEGER	
IL	46348	/COR/		INTEGER	30
IND	4478			INTEGER	
ITUNN	308	//		CHAR*8	30
JA	2	DUMMY-ARG		INTEGER	
JB	1	DUMMY-ARG		INTEGER	
JFL	45408	/INTF/		INTEGER	30
JG	5	DUMMY-ARG		INTEGER	
JJ	4408			INTEGER	
K	4358			INTEGER	
KERR	4468			INTEGER	
KFL	45768	/INTF/		INTEGER	30
KK	4378			INTEGER	
NUMF	4	DUMMY-ARG		INTEGER	
ROW	45408	/COR/		REAL	30
TCHECK	4438			REAL	
TCO	0B	/TIMES/		REAL	
TEX	2B	/TIMES/		REAL	
TLIVE	3B	/TIMES/		REAL	
TWA	1B	/TIMES/		REAL	
WCOR	4448			REAL	
WTEST	4458			REAL	
X	45768	/COR/		REAL	30
XCONF	50628	/COR/		REAL	30
XE	4548			REAL	
XER	51208	/COR/		REAL	30
XINT	50248	/COR/		REAL	30

--PROCEDURES--(LO=A)

NAME	TYPE	ARGS	CLASS
EXP	GENERIC	1	INTRINSIC
FLOAT	REAL	1	INTRINSIC
ORDER		14	SUBROUTINE
SQRT	GENERIC	1	INTRINSIC

```
DC=-LONG7-DT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I=IDEN,L=LID,B=NUCLIDE.

      1                SUBROUTINE DIAG(N,A)
      2                DIMENSION A(80,30)
      3                DIMENSION R(80)
      4                IF(N.EQ.1) GO TO 80
      5                SUM=0.
      6                DO 10 I=1,N
      7                R(I)=0.
      8                R(I)=A(I,I)
      9             10 CONTINUE
     10                DO 20 I=1,N
     11                DO 20 J=1,N
     12                A(I,J)=A(I,J)/R(I)
     13             20 CONTINUE
     14                K=N+1
     15             30 K=K-1
     16                I=K
     17             40 I=I-1
     18                J=I
     19             50 J=J+1
     20                SUM=SUM+A(I,J)*A(K,J)
     21                IF(J.LT.K) GO TO 50
     22                A(K,I)=-SUM
     23                SUM=0.
     24                IF(I.GT.1) GO TO 40
     25                IF(K.GT.2) GO TO 30
     26                DO 70 I=1,N
     27                DO 60 J=I,N
     28                SUM=SUM+(A(J,I)/R(J))**2
     29             60 CONTINUE
     30                A(I,I)=SUM
     31                SUM=0.
     32             70 CONTINUE
     33                RETURN
     34             80 A(1,1)=1./A(1,1)**2
     35                RETURN
     36                END


--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE      -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  A            2   DUMMY-ARG                    REAL          2400    N            1   DUMMY-ARG                    INTEGER
  I         3228                                INTEGER               R         2018                                REAL          80
  J         3258                                INTEGER               SUM       3218                                REAL
  K         3278                                INTEGER


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF       -LABEL-ADDRESS-----PROPERTIES----DEF

     10   INACTIVE   DO-TERM          9          50     768                          19
     20   INACTIVE   DO-TERM         13          60   INACTIVE   DO-TERM          29
     30     708                     15          70   INACTIVE   DO-TERM          32
     40     738                     17          80    1708                        34
```

```
      SUBROUTINE GLSQ(X,IL,N,M,ALPHA,E1,E2,A)
      DIMENSION A(80,30)
      DIMENSION X(30),IL(30)
      MM=M+1
      LL=1
      DO 60 J=1,MM
60    IL(J)=0
      I=1
      DO 3 K=1,MM
      II=I+1
      DO 4 J=II,N
      IF(ABS(A(J,K))-E1) 4,4,6
6     T1=SQRT ((A(J,K))**2+(A(I,K))**2)
      S=A(J,K)/T1
      C=A(I,K)/T1
      DO 5 L=K,MM
      T2=C*A(I,L)+S*A(J,L)
      A(J,L)=-S*A(I,L)+C*A(J,L)
5     A(I,L)=T2
      LL=LL+1
4     CONTINUE
      IF (ABS (A(I,K))-E2)3,3,8
8     IL(K)=I
      I=I+1
3     CONTINUE
      X(MM)=-1.0
      II=M
      DO 35 I=1,M
35    X(I)=0.
      DO 30 J=1,M
      IF (IL(II))30,30,31
31    S=0.
      LL=II+1
      I=IL(II)
      DO 32 K=LL,MM
32    S=S+A(I,K)*X(K)
      X(II)=-S/A(I,II)
30    II=II-1
      IF (IL(MM))50,51,50
51    ALPHA=0.
      GO TO 52
50    I=IL(MM)
      ALPHA=A(I,MM)
52    RETURN
      END
```

--VARIABLE MAP--(LO=A)

-NAME-	--ADDRESS--	-BLOCK-	----PROPERTIES----	-TYPE-	---SIZE
A	8	DUMMY-ARG		REAL	2400
ALPHA	5	DUMMY-ARG		REAL	
C	3408			REAL	
E1	6	DUMMY-ARG		REAL	
E2	7	DUMMY-ARG		REAL	
I	3318			INTEGER	
II	3348			INTEGER	
IL	2	DUMMY-ARG		INTEGER	30
J	327B			INTEGER	
K	332B			INTEGER	

```
      1               SUBROUTINE IDENT
      2               CHARACTER*8 COL,ISOR,BLNK,ITUNN
      3               COMMON/FITIN/FITCH(100),ENERG(100),EFFSUM(100),EFER(100)
      4               COMMON/LIBR/THAR,APRO,ANEU,CRO,THR,EREF(30),GREF(30)
      5               COMMON/LIB2/ISOR
      6               COMMON/COR/B(80,30),ROW(30),X(30),IL(30),CONF(30),
      7              &HALF(30),ACOL5(30),XINT(30),XCONF(30),XER(30),
      8              &ACOL2(30),ACOL3(30)
      9               COMMON/TIMES/TCO,TWA,TEX,TLIVE,ETOL
     10               COMMON/INTF/A(80,30),JFL(30),KFL(30)
     11               COMMON COL(30),ITUNN(30)
     12               READ(14,101)NUMF
     13           101 FORMAT(I3)
     14               IF(NUMF.LE.0) THEN
     15                 WRITE(6,*)'NO PEAKS ... CASE TERMINATED.'
     16                 STOP
     17               ENDIF
     18               DO 1000 I=1,NUMF
     19               READ(14,200) FITCH(I),ENERG(I),EFFSUM(I),EFER(I)
     20           200 FORMAT((4E14.7))
     21          1000 CONTINUE
     22               WRITE(11,1015)
     23          1015 FORMAT('
     24              15X,'PEAK NO.    CHANNEL        ENERGY          INTENS.       PC-',
     25              2'ERROR'/)
     26               WRITE(11,1016)(I,FITCH(I),ENERG(I),EFFSUM(I),EFER(I),I
     27              X=1,NUMF)
     28          1016 FORMAT(1H ,6X,I7,2F12.4,1E12.5,F12.4)
     29               IF(ETOL.EQ.0.)ETOL=2.0
     30               DO 10 I=1,30
     31               DO 8 J=1,80
     32               A(J,I)=0.
     33               B(J,I)=0.
     34             8 CONTINUE
     35               CONF(I)=1.
     36               ACOL2(I)=0.
     37               ACOL3(I)=0.
     38            10 CONTINUE
     39               AL2=ALOG(2.)
     40               DT=TCO+TWA
     41          C
     42          C    START SEARCHING THE LIBRARY
     43          C
     44                 JMAT=0
     45               IP=0
     46             9 CONTINUE
     47               CALL LIBRAR(IP,LINE,BLNK,NR1,NOISO,LINES)
     48               IP=1
     49               IF(ISOR.EQ.BLNK) GO TO 70
     50          C
     51          C      SEARCH FOR MATCHES BETWEEN LIBRARY AND SPECTRUM
     52          C
     53                 IROW=0
     54               JA=1
     55               CALL MATRIX(LINE,NUMF,ETOL,JMAT,DT,BLNK)
```

```
      GO TO 9
   70 JG=0
      WRITE(11,7070) NR1,NOISO,LINES
 7070 FORMAT(1X,/' LIBRARY NUMBER',I3,',',I4,' NUCLIDES,',I4,' GAMMA
     1LINES')
      WRITE(11,2001)
      IF(JMAT.LE.0) GO TO 220
C
C       DETERMINE THE UNIDENTIFIED PEAKS
C
        DO 110 J=1,NUMF
      DO 100 K=1,JMAT
      IF(A(J,K).GT.1.E-36)GO TO 110
  100 CONTINUE
      CALL ORDER(1,IROW,FLOAT(J),ROW,DUM,DUM,DUM,DUM,DUM,DUM,
     1DUM,DUM,DUM1,DUM1)
  110 CONTINUE
   75 IF(JMAT.LE.0)GO TO 220
      ICHEC=0
      CALL INTRF(NUMF,JMAT,JB,JA)
  150 IF(JB-1)187,175,151
C
C       SOLVE FOR ISOTOPES. LEAST SQUARES SOLUTION
C
  151   IF(JA.EQ.1) GO TO 198
  152 DO 156 I=1,JA
      JJ=JFL(I)
      DO 155 K=1,JB
      KK=KFL(K)
      B(I,K)=100.*A(JJ,KK)/(EFER(JJ)*EFFSUM(JJ))
      B(I,K)=AMAX1(B(I,K),1.0E-38)
      B(I,JB+1)=100./EFER(JJ)
  155 CONTINUE
  156 CONTINUE
      E1=1.E-38
      E2=1.E-38
      CALL GLSQ(X,IL,JA,JB,ALPHA,E1,E2,B)
      CALL DIAG(JB,B)
C
C       CHECK FOR NEGATIVE INTENSITIES
C
        XPREV=1.E+06
      I=1
  173 IF(X(I).LT.0.)GO TO 177
  174 IF(I.GE.JB)GO TO 300
      I=I+1
      GO TO 173
  177 IF(X(I).LT.XPREV)GO TO 178
      GO TO 174
  178 XPREV=X(I)
      ICHEC=I
      GO TO 174
C
C       CHECK FOR LARGE ERRORS
C
  180   DO 181 I=1,JB
      BC=1./SQRT(B(I,I))
```

```
      IF(BC.LE.1.E-30.OR.X(I).LE.1.E-30)GO TO 6000
      ERROR=100.*SQRT(B(I,I))/X(I)
      IF(ERROR.GT.50.)GO TO 182
  181 CONTINUE
      CALL CORR(JB,JA,AL2,NUMF,JG,ICHEC)
      GO TO 184
 6000 CONTINUE
  182 ICHEC=I
  300 IF(ICHEC.EQ.0)GO TO 180
C
C       CHECK FOR NEW UNIDENTIFIED PEAKS
C
        M1=KFL(ICHEC)
      DO 302 JJ=1,NUMF
      IF(A(JJ,M1).LT.1.E-36)GO TO 302
      DO 304 I=1,JMAT
      IF(I.EQ.M1)GO TO 304
      IF(A(JJ,I).GT.1.E-36)GO TO 302
  304 CONTINUE
      CALL ORDER(1,IROW,FLOAT(JJ),ROW,DUM,DUM,DUM,DUM,
     1DUM,DUM,DUM,DUM,DUM1,DUM1)
  302 CONTINUE
C
C       REMOVE THE DISCARDED ISOTOPE FROM THE WORKING MATRIX
C
        DO 311 J=M1,JMAT
      DO 310 II=1,NUMF
      A(II,J)=A(II,J+1)
  310 CONTINUE
      COL(J)=COL(J+1)
      CONF(J)=CONF(J+1)
      ACOL2(J)=ACOL2(J+1)
      HALF(J)=HALF(J+1)
  311 CONTINUE
      JMAT=JMAT-1
      GO TO 75
C
C       SOLVE FOR A SINGLE ISOTOPE
C
  175   IF(JA.GT.1)GO TO 152
   198 CALL CORR(JB,JA,AL2,NUMF,JG,ICHEC)
      IF(ICHEC.NE.0)GO TO 300
  184 DO 186 I=1,JA
      JJ=JFL(I)
      DO 185 K=1,JB
      KK=KFL(K)
      A(JJ,KK)=0.
  185 CONTINUE
  186 CONTINUE
      IF(JG.LT.JMAT)GO TO 75
  187 CONTINUE
      WRITE(11,1003)
      WRITE(11,1004)(I,ITUNN(I),XCONF(I),XINT(I),XER(I),ACO
     XL5(I),I=1,JG)
      SUM=0.
      DO 1191 I=1,JG
      SUM=SUM+XINT(I)
```

```
 1191 CONTINUE
      WRITE(11,1020)SUM
 1020 FORMAT(1H ,1X,/,'  TOTAL ACTIVITY ACCOUNTED FOR',1PE10.4,' PCI'//)
      WRITE(11,1012)
      IF(IROW.EQ.0)RETURN
  190 WRITE(11,1007)
      DO 295 I=1,IROW
      NE=INT(ROW(I))
      WRITE(11,1008) NE,FITCH(NE),ENERG(NE),EFFSUM(NE),EFER(NE)
  295 CONTINUE
      RETURN
  220 WRITE(11,1001)
      DO 222 I=1,NUMF
  222 ROW(I)=FLOAT(I)
      IROW=NUMF
      GO TO 190
 1001 FORMAT(50H NONE OF THE PEAKS WERE MATCHED WITH LIBRARY DATA[//)
 1002 FORMAT(1X,A72)
 1003 FORMAT(1X,39H THE FOLLOWING ISOTOPES WERE IDENTIFIED/,
     11X,6HNUMBER,3X,7HNUCLIDE,2X,10HCONF.VALUE,1X,11HSAMPLE ACT.,2X
     2,8HPC-ERROR,1X,9HSAT. ACT./1H ,36X,3HPCI,19X,3HPCI)
 1004 FORMAT(1H ,1X,I4,5X,A8,3X,0PF7.4,1X,1PE10.4,4X,0PF7.2,2X
     X,1PE10.4)
 1007 FORMAT(1H ,1X,40H THE FOLLOWING PEAKS WERE NOT IDENTIFIED/7H NUMBER,2
     1X,7HCHANNEL,5X,6HENERGY,5X,9HINTENSITY,4X,8HPC-ERROR/)
 1008 FORMAT(1H ,1X,I4,2X,0PF9.3,3X,0PF9.3,3X,1PE10.3,4X,0PF7.2,4X,A6,4X,A1
     X,A1,1X,A6,4X,A1,A1)
 1012 FORMAT(1X,/)
 2001 FORMAT(1H ,1X,20H ALL PEAKS CHECKED  ,//)
      END
```

--VARIABLE MAP--(LO=A)

-NAME-	--ADDRESS--	BLOCK-----	PROPERTIES-------	TYPE---------	SIZE
A	0B	/INTF/		REAL	2400
ACOL2	5156B	/COR/		REAL	30
ACOL3	5214B	/COR/		REAL	30
ACOL5	4766B	/COR/		REAL	30
ALPHA	1451B		*S*	REAL	
AL2	1421B			REAL	
ANEU	2B	/LIBR/		REAL	
APRO	1B	/LIBR/		REAL	
B	0B	/COR/		REAL	
BC	1454B			REAL	2400
BLNK	1411B			REAL	
COL	0B	//		CHAR*8	
CONF	4672B	/COR/		CHAR*8	30
CRO	3B	/LIBR/		REAL	30
DT	1422B			REAL	
DUM	1437B			REAL	
DUM1	1440B			REAL	
EFER	454B	/FITIN/		REAL	
EFFSUM	310B	/FITIN/		REAL	100
ENERG	144B	/FITIN/		REAL	100
EREF	5B	/LIBR/		REAL	100
ERROR	1455B			REAL	30

-NAME-	--ADDRESS--	BLOCK-----	PROPERTIES-------	TYPE---------	SIZE
ETOL	4B	/TIMES/		REAL	
E1	1447B			REAL	
E2	1450B			REAL	
FITCH	0B	/FITIN/		REAL	100
GREF	43B	/LIBR/		REAL	30
HALF	4730B	/COR/		REAL	30
I	1413B			INTEGER	
ICHEC	1441B			INTEGER	
II	1462B			INTEGER	
IL	4634B	/COR/		INTEGER	30
IP	1424B			INTEGER	
IROW	1431B			INTEGER	
ISOR	0B	/LIB2/		CHAR*8	
ITUNN	30B	//		CHAR*8	30
J	1417B			INTEGER	
JA	1432B			INTEGER	
JB	1442B			INTEGER	
JFL	4540B	/INTF/		INTEGER	30
JG	1433B			INTEGER	
JJ	1444B			INTEGER	
JMAT	1423B			INTEGER	
K	1435B			INTEGER	

```
PINS,I=IDEN,L=LID,B=NUCLIDE.
```

```
1                SUBROUTINE INTRF(NUMF,JMAT,JB,JA)
2                COMMON/INTF/A(80,30),JFL(30),KFL(30)
3                K=1
4                JA=0
5                JB=0
6           113  DO 115 J=1,NUMF
7                IF(A(J,K).LT.1.E-36)GO TO 115
8                JA=JA+1
9                IF(JA.EQ.1)JB=JB+1
10               JFL(JA)=J
11               KFL(JB)=K
12          115  CONTINUE
13               IF(JB.GE.1)GO TO 117
14               IF(K.GE.JMAT)RETURN
15               K=K+1
16               GO TO 113
17          117  K1=K
18          118  K1=K1+1
19               JCH=JA
20               IF(K1.GT.JMAT)RETURN
21               DO 120 JC=1,JA
22               JJ=JFL(JC)
23               IF(A(JJ,K1).LT.1.E-36)GO TO 120
24               DO 122 I=1,JB
25               IF(KFL(I).EQ.K1)GO TO 118
26          122  CONTINUE
27               GO TO 125
28          120  CONTINUE
29               GO TO 118
30          125  JB=JB+1
31               KFL(JB)=K1
32               DO 135 J=1,NUMF
33               IF(A(J,K1).LT.1.E-36)GO TO 135
34               DO 130 I=1,JA
35               IF(JFL(I).EQ.J)GO TO 135
36          130  CONTINUE
37               JA=JA+1
38               JFL(JA)=J
39          135  CONTINUE
40               IF(JA.GT.JCH)GO TO 117
41               GO TO 118
42               END
```

--VARIABLE MAP--(LO=A)

-NAME-	--ADDRESS--	--BLOCK-----	PROPERTIES-------	TYPE---------	SIZE
A	0B	/INTF/		REAL	2400
I	221B			INTEGER	
J	212B			INTEGER	
JA	4		DUMMY-ARG	INTEGER	
JB	3		DUMMY-ARG	INTEGER	
JC	216B			INTEGER	
JCH	215B			INTEGER	

-NAME-	--ADDRESS--	--BLOCK-----	PROPERTIES-------	TYPE---------	SIZE
JFL	4540B	/INTF/		INTEGER	30
JJ	220B			INTEGER	
JMAT	2		DUMMY-ARG	INTEGER	
K	211B			INTEGER	
KFL	4576B	/INTF/		INTEGER	30
K1	214B			INTEGER	
NUMF	1		DUMMY-ARG	INTEGER	

```
      1                SUBROUTINE LIBRAR(IP,NGA,BLNK,NR1,NOISO,LINES)
      2                COMMON/LIBR/THH,APR,ANE,CRO,THR,ENE(30),PCE(30)
      3                COMMON/LIB2/ISO
      4                CHARACTER*1 IHA
      5                CHARACTER*8 ISO,BLNK,UNK
      6                IF(IP.GT.0) GO TO 890
      7                NOISO=0
      8                LINES=0
      9                IF(NR1.EQ.0)NR1=1
     10                READ(2,6000) UNK,BLNK
     11           6000 FORMAT(A8,/,A8,124X)
     12            892 READ(2,7000) LIB,BNR
     13           7000 FORMAT(A8,2X,E10.0,112X)
     14            890 READ(2,7008) ISO,THA,IHA,NGA,APR,ANE,CRO,THR
     15           7008 FORMAT(A8,2X,E10.0,A1,7X,I2,4E10.0,62X)
     16                IF(ISO.EQ.BLNK) RETURN
     17                NOISO=NOISO+1
     18                THH=THA
     19                IF(IHA.EQ.'S') THH=THA/60.
     20                IF(IHA.EQ.'H') THH=THA*60.
     21                IF(IHA.EQ.'D') THH=THA*1440.
     22                IF(IHA.EQ.'Y') THH=THA*1440.*365.25
     23                IF(CRO.EQ.0.) CRO=1.
     24                READ(2,7009) (ENE(I),PCE(I),I=1,NGA)
     25           7009 FORMAT(8E10.0)
     26                LINES=LINES+NGA
     27                RETURN
     28                END
```

--VARIABLE MAP--(LO=A)

-NAME-	--ADDRESS--	BLOCK-----	PROPERTIES-------	TYPE---------	SIZE
ANE	28	/LIBR/		REAL	
APR	18	/LIBR/		REAL	
BLNK	3	DUMMY-ARG		CHAR*8	
BNR	264B		*S*	REAL	
CRO	38	/LIBR/		REAL	
ENE	58	/LIBR/		REAL	30
I	266B			INTEGER	
IHA	261B			CHAR*1	
IP	1	DUMMY-ARG		INTEGER	
ISO	08	/LIB2/		CHAR*8	

-NAME-	--ADDRESS--	BLOCK-----	PROPERTIES-------	TYPE---------	SIZE
LIB	263B		*S*	INTEGER	
LINES	6	DUMMY-ARG		INTEGER	
NGA	2	DUMMY-ARG		INTEGER	
NOISO	5	DUMMY-ARG		INTEGER	
NR1	4	DUMMY-ARG		INTEGER	
PCE	43B	/LIBR/		REAL	30
THA	265B			REAL	
THH	08	/LIBR/		REAL	
THR	48	/LIBR/		REAL	
UNK	262B			CHAR*8	

--STATEMENT LABELS--(LO=A)

-LABEL-	--ADDRESS-----	PROPERTIES----	DEF
890	33B		14
892	*NO REFS*		12
6000	145B	FORMAT	11
7000	150B	FORMAT	13
7008	153B	FORMAT	15
7009	160B	FORMAT	25

```
FTN5,I=IDEN,L=LID,B=NUCLIDE.

      SUBROUTINE MATRIX(LINE,NUMF,ETOL,JMAT,DT,BLNK)
      CHARACTER*8 ISOR,BLNK,COL,ITUNN
      COMMON/FITIN/FITCH(100),ENERG(100),EFFSUM(100),EFER(100)
      COMMON/LIBR/THAR,APRO,ANEU,CRO,THR,EREF(30),GREF(30)
      COMMON/COR/B(80,30),ROW(30),X(30),IL(30),CONF(30),
     &HALF(30),ACOL5(30),XINT(30),XCONF(30),XER(30),
     &ACOL2(30),ACOL3(30)
      COMMON/INTF/A(80,30),JFL(30),KFL(30)
      COMMON/LIB2/ISOR
      COMMON COL(30),ITUNN(30)
      EFMIN=1.E+38
      DO 11 I=1,NUMF
      EFMIN=AMIN1(EFMIN,EFFSUM(I))
   11 CONTINUE
      TEST1=0.
      DO 12 I=1,LINE
      DO 12 J=1,NUMF
      DIFF=ABS(ENERG(J)-EREF(I))
      IF(DIFF.GT.ETOL) GO TO 12
C
C       FORM THE WORKING MATRIX
C
        JPREV=1000
      PREVC=1000.
      JMAT=JMAT+1
      CHECK=1000.
      COL(JMAT)=ISOR
      ACOL2(JMAT)=APRO*1000.+ANEU
      HALF(JMAT)=THAR
      EFREF=100.*EFFSUM(J)/GREF(I)
      DIFF1=ENERG(1)-ETOL
      DIFF2=ENERG(NUMF)+ETOL
      DO 65 J1=1,LINE
      IF(EREF(J1).LT.DIFF1.OR.EREF(J1).GT.DIFF2) GO TO 65
      DO 50 JJ2=1,NUMF
      DIFF=ABS(ENERG(JJ2)-EREF(J1))
      IF(DIFF.GE.CHECK) GO TO 50
      JCH2=JJ2
      CHECK=DIFF
   50 CONTINUE
      IF(CHECK.GT.ETOL) GO TO 60
   51 IF(JPREV.NE.JCH2) GO TO 56
      IF(CHECK.LT.PREVC) GO TO 56
      GO TO 65
   60 CHEC2=GREF(J1)*EFREF/100.
      IF(CHEC2.GT.EFMIN) GO TO 51
      GO TO 65
   56 A(JCH2,JMAT)=GREF(J1)/100.
      TESTD=(0.16/ETOL)*CHECK*CHECK
      TEST1=TEST1+TESTD
      JPREV=JCH2
      PREVC=CHECK
      CHECK=1000.
   65 CONTINUE
      IF(TEST1.GT.70.) GO TO 6101
```

```
                CONF(JMAT)=CONF(JMAT)*EXP(-TEST1)
                GO TO 6102
 6101           CONF(JMAT)=0.
 6102           CONTINUE
                TEST1=0.
                TEST2=5.116856E-03*(DT/THAR)**2.0
                IF(TEST2.GT.70.)GO TO 6201
                CHECK=EXP(-TEST2)
                GO TO 6202
 6201           CHECK=0.0
 6202           CONTINUE
                CONF(JMAT)=CONF(JMAT)*CHECK
                IF(CONF(JMAT).GE.0.1) RETURN
                HALF(JMAT)=0.
                ACOL2(JMAT)=0.
                COL(JMAT)=BLNK
                CONF(JMAT)=1.
                DO 44 KK=1,NUMF
                A(KK,JMAT)=0.
   44           CONTINUE
                JMAT=JMAT-1
     12         CONTINUE
                RETURN
                END
```

--VARIABLE MAP--(LO=A)

-NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
A	0B	/INTF/		REAL	2400
ACOL2	5156B	/COR/		REAL	30
ACOL3	5214B	/COR/		REAL	30
ACOL5	4766B	/COR/		REAL	30
ANEU	2B	/LIBR/		REAL	
APRO	1B	/LIBR/		REAL	
B	0B	/COR/		REAL	2400
BLNK	6	DUMMY-ARG		CHAR*6	
CHECK	424B			REAL	
CHEC2	435B			REAL	
COL	0B	//		CHAR*8	30
CONF	4672B	/COR/		REAL	30
CRO	3B	/LIBR/		REAL	
DIFF	421B			REAL	
DIFF1	426B			REAL	
DIFF2	427B			REAL	
DT	5	DUMMY-ARG		REAL	
EFER	454B	/FITIN/		REAL	
EFFSUM	310B	/FITIN/		REAL	100
EFMIN	412B			REAL	100
EFREF	425B			REAL	
ENERG	144B	/FITIN/		REAL	
EREF	5B	/LIBR/		REAL	100
ETOL	3	DUMMY-ARG		REAL	30
FITCH	0B	/FITIN/		REAL	
GREF	43B	/LIBR/		REAL	100
HALF	4730B	/COR/		REAL	30
				REAL	30

-NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
I	413B			INTEGER	
IL	4634B	/COR/		INTEGER	30
ISOR	0B	/LIB2/		CHAR*8	
ITUNN	30B	//		CHAR*8	30
J	417B			INTEGER	
JCH2	434B			INTEGER	
JFL	4540B	/INTF/		INTEGER	30
JJ2	432B			INTEGER	
JMAT	4	DUMMY-ARG		INTEGER	
JPREV	422B			INTEGER	
J1	430B			INTEGER	
KFL	4576B	/INTF/		INTEGER	30
KK	440B			INTEGER	
LINE	1	DUMMY-ARG		INTEGER	
NUMF	2	DUMMY-ARG		INTEGER	
PREVC	423B			REAL	
ROW	4540B	/COR/		REAL	30
TESTD	436B			REAL	
TEST1	415B			REAL	
TEST2	437B			REAL	
THAR	0B	/LIBR/		REAL	
THR	4B	/LIBR/		REAL	
X	4576B	/COR/		REAL	30
XCONF	5062B	/COR/		REAL	30
XER	5120B	/COR/		REAL	30
XINT	5024B	/COR/		REAL	30

```
      SUBROUTINE ORDER(K,N,XNEW,X,YNEW,Y,ZNEW,Z,WNEW,W,TNEW,T,
     XSNEW,S)
      CHARACTER*8 SNEW,S
      DIMENSION X(30),Y(30),Z(30),W(30),T(30),S(30)
      IF(N.LT.0.OR.N.GT.151)STOP
      IF(N.GT.0)GO TO 20
      M=N
      GO TO 80
   20 IF(XNEW.LT.X(N))GO TO 40
      M=N
      GO TO 80
   40 I1=0
   50 I1=I1+1
      IF(XNEW.GE.X(I1))GO TO 50
      I2=N+2
   60 I2=I2-1
      GO TO(70,69,68,67,66,64),K
   64 S(I2)=S(I2-1)
   66 T(I2)=T(I2-1)
   67 W(I2)=W(I2-1)
   68 Z(I2)=Z(I2-1)
   69 Y(I2)=Y(I2-1)
   70 X(I2)=X(I2-1)
      IF(I2.GT.I1+1)GO TO 60
      M=I1-1
   80 M=M+1
      GO TO(90,89,88,87,86,84),K
   84 S(M)=SNEW
   86 T(M)=TNEW
   87 W(M)=WNEW
   88 Z(M)=ZNEW
   89 Y(M)=YNEW
   90 X(M)=XNEW
      N=N+1
      RETURN
      END
```

--VARIABLE MAP--(LO=A)

-NAME-	--ADDRESS--	--BLOCK-----	PROPERTIES-------	TYPE---------	SIZE
I1	2708			INTEGER	
I2	2718			INTEGER	
K	1	DUMMY-ARG		INTEGER	
M	2678			INTEGER	
N	2	DUMMY-ARG		INTEGER	
S	14	DUMMY-ARG		CHAR*8	30
SNEW	13	DUMMY-ARG		CHAR*8	
T	12	DUMMY-ARG		REAL	30
TNEW	11	DUMMY-ARG		REAL	
W	10	DUMMY-ARG		REAL	30
WNEW	9	DUMMY-ARG		REAL	
X	4	DUMMY-ARG		REAL	30
XNEW	3	DUMMY-ARG		REAL	
Y	6	DUMMY-ARG		REAL	30
YNEW	5	DUMMY-ARG		REAL	
Z	8	DUMMY-ARG		REAL	30
ZNEW	7	DUMMY-ARG		REAL	

APPENDIX-F. OUTPUT EXAMPLES

PEAK SEARCH BETWEEN 80 AND 4096 CHANNELS
ATGE= .80 ATLT= .80 FAC= .80 ADEL= .80 ATEST= .80 PTEST= 1.00

SPECTRUM INPUT FROM FILE IAEAG2

THERE MAY BE A PEAK NEAR CHANNEL # 117
THERE MAY BE A PEAK NEAR CHANNEL # 119
THERE MAY BE A PEAK NEAR CHANNEL # 121
THERE MAY BE A PEAK NEAR CHANNEL # 896
THERE MAY BE A PEAK NEAR CHANNEL # 1026
THERE MAY BE A PEAK NEAR CHANNEL # 1324
THERE MAY BE A PEAK NEAR CHANNEL # 2096
THERE MAY BE A PEAK NEAR CHANNEL # 2759
THERE MAY BE A PEAK NEAR CHANNEL # 3268
THERE MAY BE A PEAK NEAR CHANNEL # 3553
THERE MAY BE A PEAK NEAR CHANNEL # 3750

INSPECT THE FIRST PEAK CAREFULLY...

NUMBER OF DETECTED PEAKS IN THIS REGION : 28

*** PEAKS FOUND ***

	MIN.CHN	MAX.CHN	PEAK AREA	COMP.CONT	STD.DEV	SIGMA	PEAK CHANNELS	
1	84	126	2781715.	106205.	.1	2.126	97.895	106.343
2	150	168	139043.	498115.	.8	2.137	161.571	
3	197	255	3826690.	1939408.	.1	2.151	246.203	
4	345	382	5879803.	1210588.	.0	2.172	370.628	
5	402	422	668428.	487835.	.2	2.179	414.249	
	481	494	49887.	271748.	1.5	1.879	487.289	
	671	683	26183.	175671.	2.3	1.877	676.503	
	718	728	10626.	124306.	4.8	1.654	723.146	
6	827	846	353095.	200239.	.2	2.249	837.380	
	887	899	5607.	116141.	8.7	2.088	894.645	
7	904	927	824838.	230736.	.1	2.262	917.378	
	1021	1029	2360.	64143.	15.3	1.683	1024.760	
8	1053	1089	2416940.	309599.	.1	2.289	1078.138	
9	1152	1177	322953.	178488.	.3	2.303	1162.312	
	1318	1328	4362.	59633.	8.1	1.943	1322.487	
10	1533	1560	163485.	176391.	.4	2.367	1546.830	
	1760	1770	4619.	69863.	8.2	1.668	1765.327	
	1838	1851	9694.	99940.	4.7	2.160	1844.448	
	2091	2099	2383.	32633.	10.9	1.720	2094.740	
11	2499	2539	1674852.	192073.	.1	2.529	2526.858	
	2753	2765	3804.	36464.	7.3	2.458	2757.633	
	2928	2940	2473.	19593.	8.3	2.199	2933.249	
	3263	3269	225.	3165.	35.9	.985	3266.173	
12	3347	3389	1244116.	45019.	.1	2.670	3376.816	
	3545	3557	1001.	2789.	8.1	2.553	3551.688	
	3741	3756	1516.	3107.	5.8	2.902	3746.074	
	3850	3867	3171.	2400.	2.8	2.482	3858.600	
	4028	4041	1031.	1246.	5.8	2.353	4034.216	

TOTAL NUMBER OF DETECTED PHOTOPEAK GROUPS 12

```
 SPECTRAL DATA READ IN FROM FILE1REAG2
 INUM INIT IFIN   MUL   II  IVAR IRPT IGUES IDISK ISAVE ISET
    6  827  846    1    3    5    0    0    1     0    0
 INITIAL PARAMETERS:


 ------------------
 COMPTON CONTINUUM
 A0=   11451.5 A1=    86.47 A2=    -6.32

 STEP FUNCTION
 H1=    12.00

 PEAK HEIGHTS (GAUSS)
 AG1=     72006.0

 TAILING HEIGHTS
 AT1=     7200.6

 PEAK CHANNELS
 X01=    11.422

 B=  .350 C= 1.000 SIGMAT= 1.500 SIGMAG= 1.868
 INITIAL CHISQR=    .303231E+02
  GRADIENTS
    .170850E+00   .878090E+00   .540057E+01   .166430E+00   .101907E-01
   -.588223E+03   .225843E-01  -.806221E+03   .163815E+03  -.232799E+02
    .109966E+04
  FUNCTION MINIMUM       IKNT= 12
 GRADIENTS
    .819678E-03  -.819766E-04   .259283E-01  -.474577E-03  -.375340E-04
   -.466249E+00   .316635E-03  -.367606E+00   .246082E+00  -.120980E-01
    .197060E+00
  FINAL CHISQR=    .171182E+02      CHISQR/(DEG.FREE)=    .190203E+01


 FINAL PARAMETERS:


 ------------------
 COMPTON CONTINUUM
 A0=     11447.8 +/-    62.7 A1=    85.05 +/-   23.9 A2=    -6.23 +/-    1.5

 STEP FUNCTION
 H1=    11.90 +/-    2.1

 PEAK HEIGHTS (GAUSS)
 AG1=     72006.0 +/-     1.4

 TAILING HEIGHTS
 AT1=     7199.9 +/-   11.8

 B=  .386 +/- .044 C=  .995 +/- .023 SIGMAT= 1.411 +/- .345 SIGMAG= 1.868 +/- .006 FWHM= 4.399

    PEAK CHNL                  TAILING AREA     GAUSS AREA     TOTAL AREA     % ERROR
 1   837.422 +/-      .012        9543.3        337166.6        346710.0         .603

              ERR.CORR=-1.0935052E+00
```

JOB. TARA G2 PROGRAM #2 SPECTRUM

ENERGY CALIBRATION DATA

	CHANNEL	ENERGY(KEV)
1	370.490	122.060
2	1077.300	356.030
3	2525.200	834.830
4	4037.500	1332.500

EFFICIENCY CALIBRATION DATA

	ENERGY(KEV)	EFFICIENCY	EFF-ERR
1	31.790	5.3000000E-04	3.7900000E+00
2	81.000	4.8600000E-03	5.2700000E+00
3	122.100	7.0200000E-03	4.4000000E+00
4	356.000	2.7300000E-03	5.0000000E+00
5	661.600	1.6300000E-03	3.7000000E+00
6	834.800	1.3200000E-03	3.7000000E+00
7	1173.200	9.2900000E-04	1.9000000E+00
8	1332.500	8.1400000E-04	1.9000000E+00

DATA INPUT FROM FILE FIT1

CALIBRATION RESULTS

PEAK#	CHANNEL	ERROR-CHN	ENERGY(KEV)	ERROR(KEV)	AREA	ERROR(%)	INTENSITY	ERROR(%)
1	97.985	.035	31.800	.017	.7450E+06	1.733	.1405E+10	5.959
2	106.343	.014	34.569	.014	.1253E+07	.173	.1938E+10	3.927
3	161.953	.014	57.989	.013	.1125E+06	.184	.6324E+08	4.616
4	246.250	.031	80.910	.015	.2451E+07	1.756	.5057E+09	7.099
5	370.659	.007	122.111	.010	.5705E+07	.320	.8128E+09	4.458
6	414.292	.023	136.561	.012	.6620E+06	1.421	.1041E+09	5.475
7	837.415	.013	276.645	.008	.3432E+06	.625	.1006E+09	5.049
8	917.376	.009	303.112	.008	.8120E+06	.491	.2581E+09	5.020
9	1078.145	.009	356.318	.008	.2307E+07	.202	.8457E+09	5.014
10	1162.313	.013	384.170	.008	.3085E+06	.503	.1204E+09	4.922
11	1546.815	.021	511.373	.011	.1567E+06	.713	.7750E+08	4.429
12	2526.842	.004	835.367	.010	.1638E+07	.182	.1241E+10	3.704
13	3376.800	.008	1116.119	.008	.1225E+07	.208	.1252E+10	2.118

14.37.55.UCLP, AA, P03 , 0.075KLNS.

********** BUCAASA/NUCLIDE REV 1.0 84/08/10. 15.05.52.**********

DB : IAEA G2 PROGRAM SOURCE #2

COUNT. TIME= 306.62(MIN) WAIT. TIME= 37926.00(MIN) EXSPSR. TIME= 1.00 (MIN)LIVE C. TIME= 20000.00(SEC)
EN. TOL= 0.00(KEV)

TABLE OF RESULTS

PEAK NO.	CHANNEL	ENERGY	INTENS.	PC-ERROR
1	97.9850	31.8004	.14047E+10	5.9592
2	106.3430	34.5690	.19381E+10	3.9369
3	161.9530	52.9892	.63240E+08	4.6105
4	246.2500	80.9097	.50575E+09	7.0991
5	370.6580	122.1112	.81281E+09	4.4579
6	414.2920	136.5607	.10410E+09	5.4755
7	837.4150	276.6455	.10061E+09	5.0489
8	917.3760	303.1116	.25808E+09	5.0262
9	1078.1450	356.3178	.84566E+09	5.0141
10	1162.3130	384.1695	.12038E+09	4.9223
11	1546.8150	511.3733	.77597E+08	4.4291
12	2526.8420	835.3673	.12415E+10	3.7040
13	3376.8600	1116.1180	.12525E+10	2.1155

LIBRARY NUMBER 1, 137 NUCLIDES, 583 GAMMA LINES
ALL PEAKS CHECKED

HE FOLLOWING PEAKS ARE ATTRIBUTED TO ISOTOPE NA-22

NUMBER	CHANNEL	ENERGY
11	1546.82	511.37

HE FOLLOWING PEAKS ARE ATTRIBUTED TO ISOTOPE MN-54

NUMBER	CHANNEL	ENERGY
12	2526.84	835.37

HE FOLLOWING PEAKS ARE ATTRIBUTED TO ISOTOPE CO-57

NUMBER	CHANNEL	ENERGY
5	370.66	122.11
6	414.29	136.56

HE FOLLOWING PEAKS ARE ATTRIBUTED TO ISOTOPE ZN-65

NUMBER	CHANNEL	ENERGY
13	3376.86	1116.12

HE FOLLOWING PEAKS ARE ATTRIBUTED TO ISOTOPE BA-133

NUMBER	CHANNEL	ENERGY
3	161.95	52.99
4	246.25	80.91
7	837.42	276.65
8	917.38	303.11
9	1078.15	356.32
10	1162.31	384.17

THE FOLLOWING ISOTOPES WERE IDENTIFIED

NUMBER	NUCLIDE	CONF.VALUE	SAMPLE ACT. PCI	PC-ERROR	SAT. ACT. PCI
1	NA-22	.9889	5.9050E+04	4.43	1.1739E+11
2	MN-54	.9771	1.7789E+06	3.70	1.1556E+12
3	CO-57	.9991	1.3562E+06	3.46	7.6524E+11
4	ZN-65	.9717	3.6602E+06	2.12	1.8554E+12
5	BA-133	.8219	1.8841E+06	2.18	1.5583E+13

TOTAL ACTIVITY ACCOUNTED FOR8.7385E+06 PCI

THE FOLLOWING PEAKS WERE NOT IDENTIFIED

NUMBE	XCHANNEL	ENERGY	INTENSITY	PC-ERROR
1	97.985	31.800	1.405E+09	5.96
2	106.343	34.569	1.938E+09	3.94

15.15.38.UCLP, AA, P04 , 0.101KLNS.

REFERENCES

1- Hevesy,G., Levi,H., Mathematics, Physics and Medicine, Vol.14, No.5, p.3, 1936.

2- Adams,F., Hoste,J., "Non-destructive Activation Analysis" Atomic Energy Review, Vol.4, No.2, pp.113-180, 1966.

3- Çiftçioğlu,Ö., Ge(Li) Nükleer Radyasyon Detektörlerinden Elde Edilen Fotopikler İçin Analitik Bir Yaklaşım ve Yüksek Doğrulukta Otomatik Gamma Spektrum Analizi, İ.T.Ü., 1981.

4- Wainerdi,R.E., Beau,N.P.Du, Science, Vol.139, p.1027, 1963.

5- Hoste,J., et al., Instrumental and Radiochemical Activation Analysis,

6- Lederer,C.M., Hollander,J.M., Perlman,L., Table of Isotopes, New York: John Wiley, 1968.

7- Crouthamel,C.E., Applied Gamma-Ray Spectrometry, Oxford: Pergamon Press, 1960.

8- Koch,R.C., Activation Analysis Handbook, New York and London: Academic Press, 1960.

9- Laul,J.C., "Neutron Activation Analysis of Geological Materials", Atomic Energy Review, Vol.17, No.3, pp.603-695, 1979.

10- Nargolwalla,S.S., Przybylowicz,E.P., Activation Analysis with Neutron Generators, New York: John Wiley and Sons, 1973.

11- Anon., AQCS Programme, IAEA Lab/243, December 1983.

12- Glasstone, Sesonske, Nuclear Reactor Engineering, 1964.

13- Price,W.J., Nuclear Radiation Detection, McGraw Hill, 1964.

14- Gardner,R.P., Ely,R.L.Jr., Radio isotope Measurement Applications in Engineering, Reinhold, 1967.

15- Borak,F., Nuclear Engineering Laboratory Notes, B.Ü., 1982.

16- Adams,J.A.S., Gasparini,P., Gamma Ray Spectrometry of Rocks, Elsevier, 1970.

17- Ciftcioğlu,Ö., "Further Note on The Analytical Approximation to the Peaks Obtained from Ge(Li) Detectors Used in Gamma Spectrum Analysis", Nuclear Instruments and Methods, Vol. 198, No.2,3, pp.391-398, July 1982.

18- Op de Beeck,J., "Gammaray Spectrometry Data Collection and Reduction by Simple Computing Systems", Atomic Energy Review, Vol.13, No.4, pp.743-806, 1975.

19- Quittner,P., Wainerdi,R.E., "Computer Evaluation of NaI(TR) and Ge(Li) Gamma Ray Spectra", Atomic Energy Review, Vol.8, No.2, pp.361-416, 1970.

20- Savitzky,A., Golay,M.J.E., "Smoothing and Differentiation of Data by Simplified Least Squares Procedures", Analytical Chemistry, Vol.36, No.8, pp.1627-1639, July 1964.

21- Yule,H.P., "Mathematical Smoothing of Gamma Ray Spectra", Nuclear Instruments and Methods, Vol.54, pp.61-65, 1967.

22- Barnes,V., "GASPAN-An Advanced Computer Code for the Analysis of High Resolution Gamma-Ray Spectra", IEEE Transactions on Nuclear Science, Vol.N5-15, No.3, pp.437-455, 1968.

23- Inouye,T., Harper,T., Rasmussen,N.C., "Application of Fourier Transforms to the Analysis of Spectral Data", Nuclear Instruments and Methods, Vol.67, pp.125-132, 1969.

24- Parr,R.M., Houtermans,H., Schaerf,K., "The IAEA Intercomparison of Methods for Processing Ge(Li) Gamma-ray Spectra: A Preliminary Report", Transactions of the American Nuclear Society Conference on Computers in Activation Analysis and Gamma-Ray Spectroscopy, Mayaguez, May 1-4, 1978.

25- Mariscotti,M.A., "A Method for Automatic Identification of Peaks in the Presence of Background and Its Application to Spectrum Analysis", Nuclear Instruments and Methods Vol.50, pp.309-320, 1967.

26- Routti,J.T., Prussin,S.G., "Photopeak Method for the Computer Analysis of Gamma Ray Spectra from Semiconductor Detectors", Nuclear Instruments and Methods, Vol.72, pp.125-142, 1969.

27- Aarnio,P.A., Koskelo,M.J., Zambori,P., "A New Method for the Determination of Goodness-of-fit in Gamma Spectroscopy", Nuclear Instruments and Methods, Vol.184, pp.487-492.

28- Erdtman,G., Soyka,W., "The Radionuclide Gamma Ray Data File GAMDAT-78", KFA, Jul-Spez-34, April 1979.

29- Christensen,G.C., Koskelo,M.J., Routti,J.T., "Gamma Spectrum Storage and Analysis Program SAMPO 76 with Nuclide Identification", HS-RP/015/, European Organization for Nuclear Research, October 1978.

30- Koskello,M.J., Aarnio,P.A., Routti,J.T., "SAMPO80: An Accurate Gamma Spectrum Analysis Method for Microcomputers", Nuclear Instruments and Methods, Vol.190, pp.89-99, 1981.