Chemical Modifications of Polymers

by

Li Jin

(Department of Chemical Engineering and Applied Chemistry)

A Thesis Submitted for the Degree of Doctor of Philosophy

at

The University of Aston in Birmingham

April 1988

THE UNIVERSITY OF ASTON IN BIRMINGHAM

Chemical Modifications of Polymers

Li Jin

A Thesis Submitted for the Degree of Doctor of Philosophy

April 1988

Summary

Based on the knowledge of PVC degradation and stabilisation, chemical modifications were imposed on degraded PVC and raw PVC with the aim of obtaining non-migrating additives. The modifications were carried out mainly in the presence of dibutyl maleate(DBM), and the resulting polymer contained dibutyl maleic residues. Such modifications result in a polymer which contain substantive additives which resist migration under aggressive environments.

Previous studies have shown that stable nitroxyl radicals function as stabilisers in polymer during processing (e.g. PP, PVC) by deactivating a large number of kinetic chains via a redox process whereby the concentrations of the nitroxyl and its reduced form, the hydroxylamine, fluctuate reciprocally and rhythmically. In order to understand the major reactions involved in such systems, a simulation method was used which resulted in a mathematical model and some rate constants, explaining the kinetic behaviour exhibited by such system. In the process of forming a suitable model, two nonlinear oscillators were proposed, which could be of interest in the study of nonlinear phenomenon because of their chaotic behaviour.

Keywords: PVC, Poly(vinyl chloride), Polypropylene, Thermal Degradation, Chemical Modification, Chemical Oscillation

Acknowledgement

I would like to express my deepest gratitude to Professor G. Scott and Dr S. Al-Malaika for their supervision and constant encouragement during this research. Especially, in the second part of the work, it might not have been possible without their tolerance and help.

I am grateful to the Ministry of Education of the People's Republic of China and the British Council for financial support throughout my research work.

I acknowledge the help of the technical staff in the Chemistry Department, the help from Dr J. Jones and Miss I. Whittier of Engineering Computing Office, and mathematical encouragement from Dr B. Martin of Department of Computer Science and Applied Mathematics.

To my wife Suibo, so much is owed, and no written words can express my profound gratitude for her patience and sacrifice.

Declaration

The work described herein was carried out at the University of Aston in Birmingham between April 1985 and April 1988.

It has been done independently and submitted for no other degrees.

April 1988

Contents

List of Figures

Part I

Chemical Modifications of PVC by Plasticiser

Chapter1

Introduction

PVC is one of the most studied subject in the field of polymer degradation and stabilisation. In spite of this, the basic details[1~9] of the mechanism of PVC degradation have not been elucidated clearly and a fundamental understanding of the stabilisation methods[10], some of which have been used empirically for many years, still needs to be clarified. However, on the basis of what is known about the degradation and stabilisation of PVC, it is possible to apply this knowledge to study the modification of PVC. This study is mainly concerned with chemical modification of PVC and degraded PVC by plasticisers. Therefore, in the following sections, the general chemical methods of polymer modification will be discussed with particular emphasis on the reactions of degraded PVC.

1.1 General Aspects of PVC Degradation

The main outward manifestations of PVC degradation (at temperatures sensibly below those of pyrolytic decomposition and combustion) are the evolution of hydrogen chloride (HCl), discolouration and deterioration of physical, chemical and electrical properties. The thermal degradation of PVC starts[11] at about 110 ^{0}C, which is much lower than the degradation temperatures for low molecular weight analogues used as models for PVC structures,e.g. 2,4,6-trichloroheptane[12]. It has therefore been assumed that the low stability of PVC under thermostatic conditions is caused by irregular structures in the polymer chain, although during processing[13] degradation is largely caused by thermomechanical action. Therefore, the following factors have been considered to be important for the understanding of PVC degradation:

Initiator residues;

Unsaturation chain ends;

Internal unsaturation;

Branches;

Head-to-head structures;

Oxidation structures;

Chain scission caused by mechanical shear.

Although there has not been a single widely agreed factor responsible for the initiation of PVC degradation, it is commonly recognised that once degradation starts, the major process afterwards involves the loss of HCl from PVC molecules with formation of polyene. It is the formation of such conjugated structures and their reactions that will be our major concern.

1.1.1 Polyene Formation During PVC Degradaton

Irrespective of the nature of the initiation site, the evolution of HCl is accompanied by the formation of polyenes in PVC. The discolouration of degraded PVC has been considerea to be due to polyene formation. One important feature of polyene generation is the restricted length of the resulting polyene sequences. The average length of conjugated double bonds has been reported to be 4~6 units[14,15~17]. The sequence distribution does not gradually change towards higher mean values. Instead, long sequences are observed even at low levels of degradation[18] and the distribution is shifted towards shorter sequences at higher conversions[14,16,18]. The fast development of long sequences has been ascribed to a rapid 'zipper' process for the growth of the polyenes.

Over the years, several different mechanisms for the formation of conjugated double bonds have been discussed. The proposed mechanisms include radical, ionic and unimolecular (concerted) paths.

1.1.1.1 Radical Mechanism

This mechanism[19,20] can be illustrated in Scheme1.1. As support for a radical mechanism, the following observations have been made:

(1) increased rate of degradation in the presence of radical initiators[21,22], γ-irradiation[11,23~25] and UV irradiation[26~28].

(2) the presence of radicals as detected by ESR spectroscopy[15,29~33].

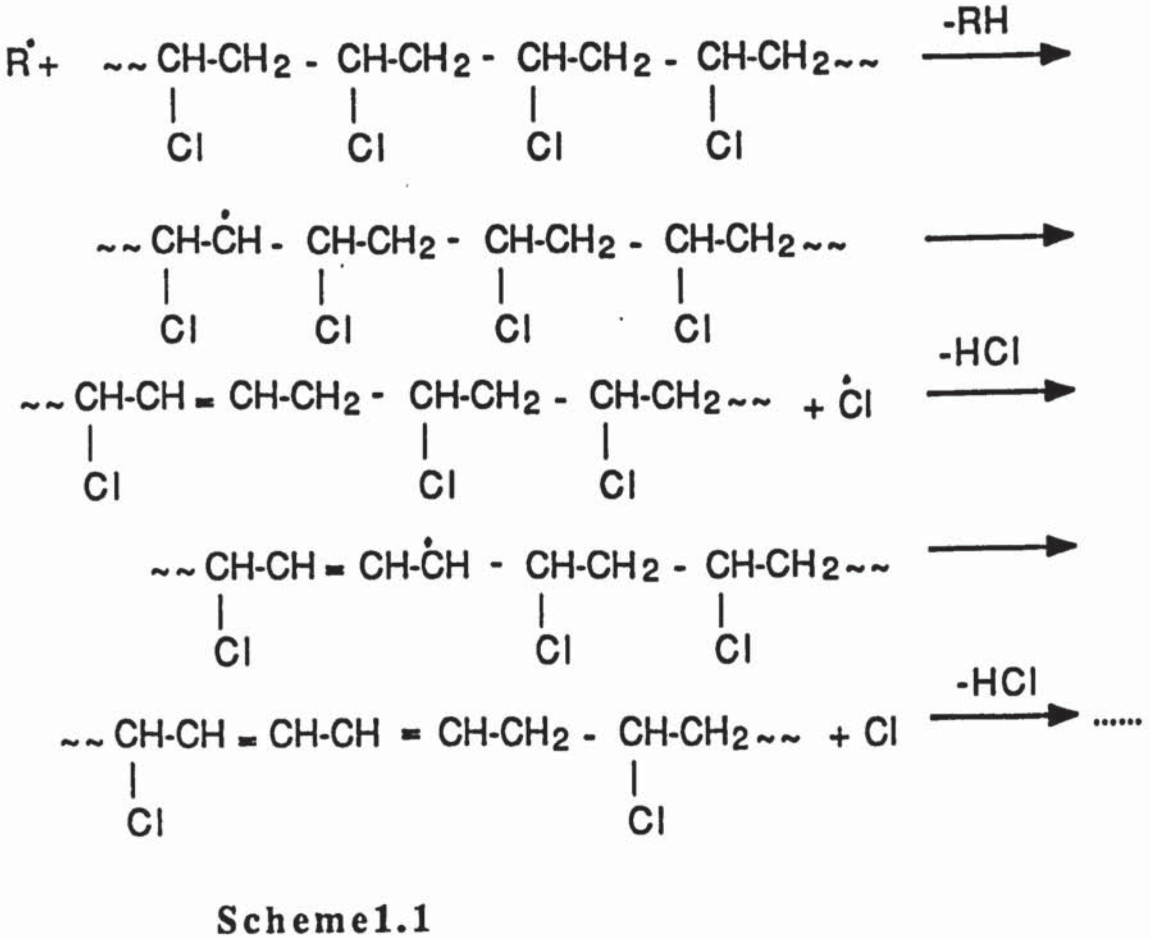
(3) grafting of methylmethacrylate to PVC during degradation[34].

(4) the retarding effect of radical traps[35,36].

(5) increased rate of degradation of polymethylmethacrylate[37~39] and polypropylene[35] in the presence of PVC.

(6) exchange reaction[34] with labelled ^{14}C toluene.

An immediate question is where does R˙ come from at the beginning of this propagation process. Under thermostatic condition, it is hard to find a reason. But under processing condition, this may well be attributed to thermomechanical effect, i.e. shear-induced polymer chain scission[13]. Although there have been so many observations to support the radical mechanism, using the same experimental techniques, there have been some other studies which are contradictary with the above.

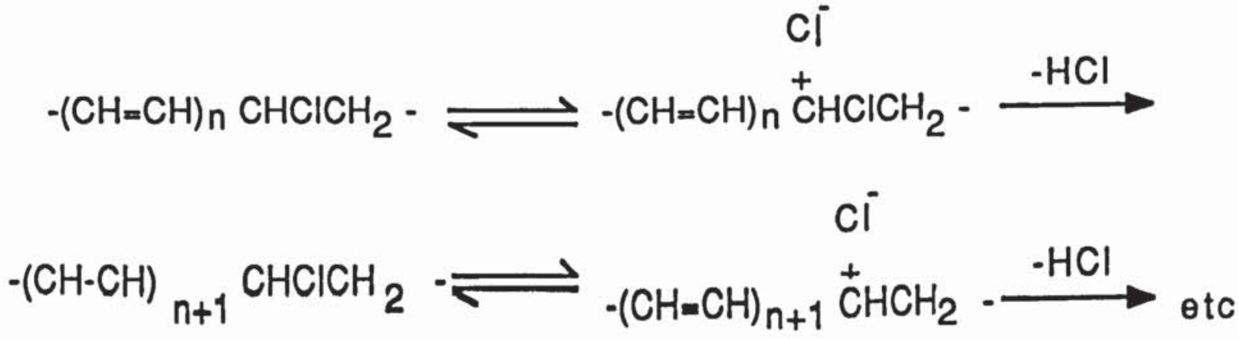


Scheme1.1

For example, it was reported that the presence of radical initiators[40,41] or inhibitors[5,42] have no effect on the rate of degradation of PVC. It has also been claimed that the observed ESR-signals originate from species which are not active in the polyene propagation but are due to thermoactivated electrons in the polyene systems[43] or to radicals connected with aromatic structures formed in secondary reactions[44] (see section1.1.2). Although it has been argued that the catalytic effect of HCl is consistent with radical mechanisms, many workers consider the influence of HCl as strong evidence against a radical mechanism[25], and ionic processes have been suggested.

1.1.1.2 Ionic Mechanism

The ionic mechanism assumes the following process for the growth of polyenes during PVC degradation:



Scheme1.2

The observations which support this mechanism are:

(1) The rate of dehydrochlorination in solution is influenced by the nature and dielectric constant of the solvent[41,45,46].

(2) The degradation is catalysed by HCl[47,48], other acids[49], strong bases[42,50], and some metal salts[40,45,51,52].

(3) The degradation behaviour of low molecular weight compounds[53].

(4) The degradation of PVC follows first order kinetics[4].

Some workers find support for a concerted mechanism from the elimination behaviour of low molecular weight compounds[2,54]. The observation of first order kinetics is also in accordance with a concerted reaction mechanism.[41,42]

Resonance stabilisation has been assumed to be an important factor in explaining the restricted length of the polyene sequences[4,17,20,55]. At increasing length an additional HCl elimination will be less and less favourable. Molecular orbital calculation showed that this assumption is not valid for neutral polyenes and that the incremental conjugation energy is constant[56,57]. This calculation also revealed that the positive charge shows a strong tendency to concentrate in the center of polyenyl cation. This effect is considered to be an important factor in restricting the length of polyene[9,56,57]. The limitation of polyene growth may also be due to other reasons, e.g. the presence of chloromethyl branches[14].

Stereoregularity is another important structural feature which influences the propagation and the termination of poylene sequences.

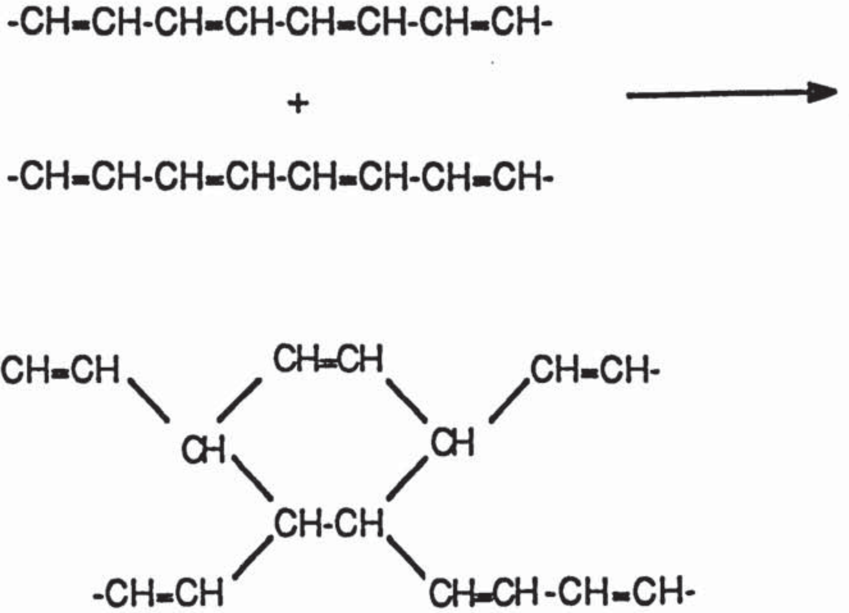
syndiotacticity, both in thermal[58~62] and UV degradation[63]. The increased length may be due to faster propagation along favoured (trans-trans) conformation[59~63].

The above discussion shows that the views concerning the initiation of PVC degradation vary widely. But the formation of poylene sequences in the degraded PVC is a widely agreed phenomenon, and these structures are able to undergo various secondary reactions.

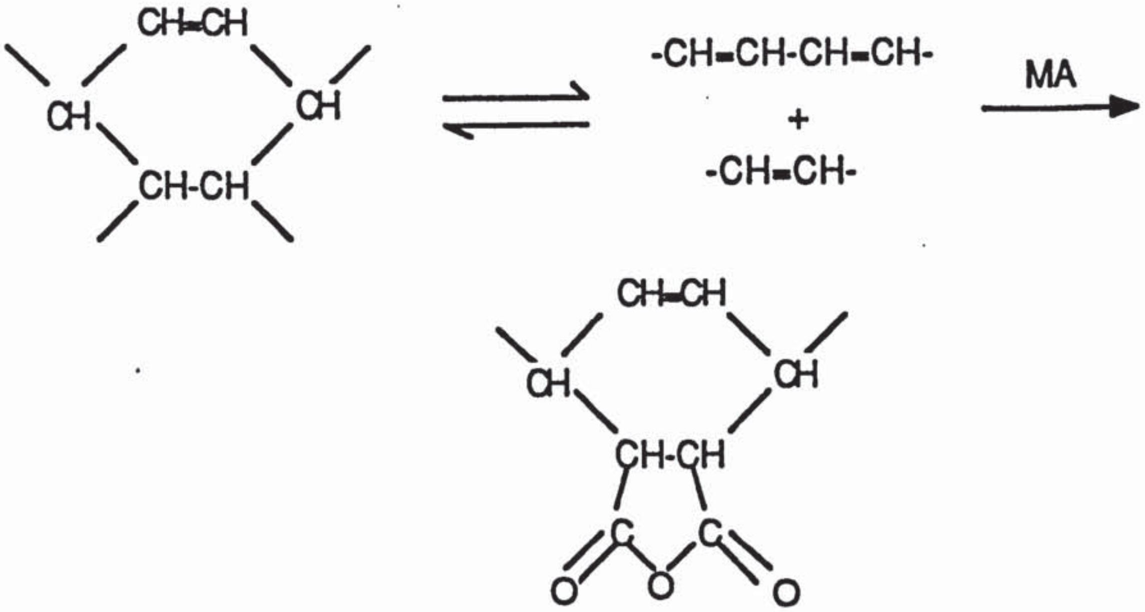
1.1.2 Secondary Reactions

The early work[64] in PVC degradation showed that insoluble material was formed during degradation. The tendency to molecular enlargment was also indicated by increased solution viscosity before gel point[21,40]. Using GPC in combination with viscosity measurement the molecular weight increases could be followed in more detail[14,65,66]. In solid state degradation, gel formation started around 1% dehydrochlorination and became extensive[65,66] after 2%. After the gel point, the molecular weight and the viscosity of the soluble material decreases rapidly[66,68].

The gel formation was assumed to occur by intermolecular HCl elimination between growing poylene sequences and ordinary methylene groups[64,65]. This mechanism would account for the increased crosslinking and decreased mean poylene length observed at increased degradation temperature[65]. It has also been considered to explain the decrease in polyene length at increasing degree of degradation[65]. But this assumption is questionable. It has been suggested that Diels-Alder cycloaddition between two polyenes[69~73] is more likely to agree with experimental facts(see Scheme1.3). It has been shown that this mechanism accounts for a major part of crosslinking[69]. By treating degraded, but still soluble PVC with maleic anhydride(MA), a decrease in the high molecular weight material formed during degradation was observed. For samples degraded beyond the gel point, a considerable decrease in the gel content was also observed. These observations were explained by the displacement in the Diels-Alder/retro-Diels-Alder equilibrium in Scheme1.4. The existence of other corsslinking mechanisms was not excluded, but it was said that at least 60-70% of crosslinking during thermal degradation of PVC is due to Diels-Alder reactions between different chains which contain conjugated unsaturation.

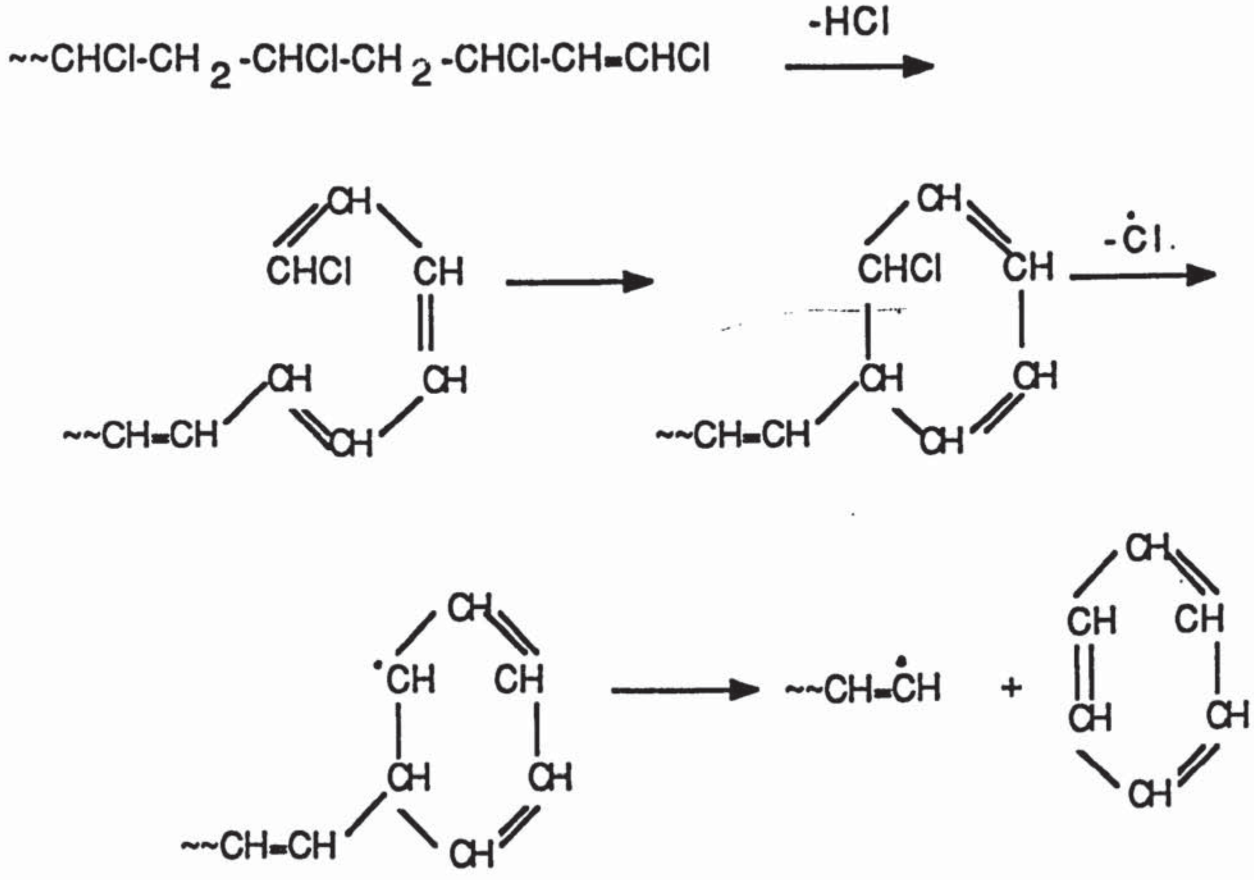


Scheme1.3



Scheme1.4

The formation of volatile aromatic pyrolysates, mainly benzene during thermal degradation of PVC is another result of secondary reactions[74]. At the conditions characteristic of PVC processing and stability test, about 0.01 mole benzene per mole evolved HCl was reported to be formed[68]. It was pointed out[75] that the cyclic product in Scheme1.3 could be a precursor of benzene. However, further investigations with deuterium-labelled PVC have shown that benzene must be formed via an intramolecular route. Degradation of mixtures of PVC and PVC-d_3 results in benzene-h_6 and benzene-d_6 with traces of other labelled benzenes[75~77]. Four different intramolecular routes[77~80] have been suggested to explain the formation of aromatic compounds. One of them involves cyclisation of polyenes formed at the end of polymer chain. It was found[78] that the evolution of benzene was dependent on the molecular weight, i.e. the concentration of end-groups, and several benzene molecules were evolved per initial chain after extensive degradation. This process is shown in Scheme1.5:



Scheme1.5

However, several other facts make this mechanism questionable. The content of 1-chlorovinyl ends is very low and the formation of vinyl radicals is unlikely[80]. The most frequent unsaturated end-groups in raw PVC molecules were found to remain unchanged during degradation[81].

1.2 Chemical Modifications of Polymers

The chemical modification of polymers together with polymerisation and ploycondensation is one of the ways to synthesize new high molecular weight compounds (see Fig.1.1):

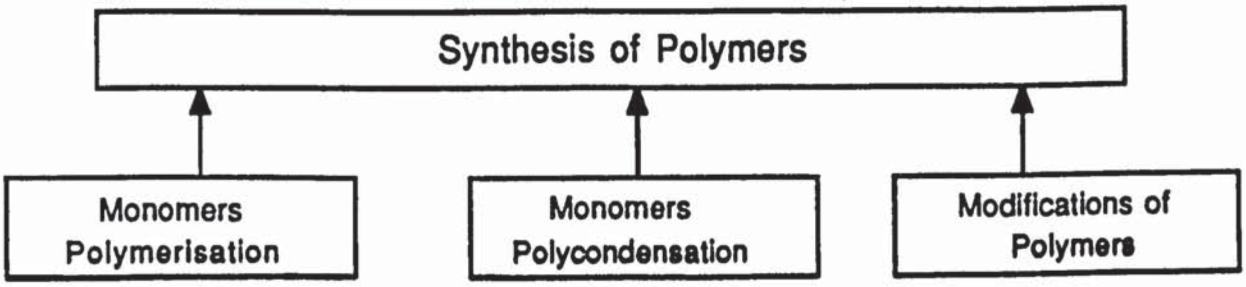


Fig.1.1 Different processes leading to the synthesis of polymers

Chemical modification of polymers is a process in which the initial polymer is subjected to chemical influences, as a result of which it is converted into a new polymer with a different chemical structure.

The first polymer modification reaction[82] can be dated back to 1833, when nitrocellulose was produced by treating cellulose with nitric acid. Similar reactions were later carried out on a series of natural polymers. Subsequently synthetic polymers such as PVC, polystyrene, polyethylene and others were subjected to chemical modifications[82].

Such use of the modification methods originates from the fact that the initial polymers do not meet all the required practical properties. Thus many insoluble polymers can be converted into soluble products by modification. A striking example of this kind is provided by cellulose, whose esters are readily soluble in organic solvents. The introduction of carboxy- and amino- groups into polystyrene imparts to it acid or basic properties respectively. Such products are used as ion-exchange resins. The introduction of nitrate ester group into cellulose imparts explosive properties to the latter. The introduction of phosphate groups as well as chlorine, fluorine, bromine and certain other elements into polymer renders them incombustible[85].

Thus the modification of polymers constitutes an extremely general method whereby it is possible to vary within a wide range of their physical and chemical properties to meet special needs. Because of this, the modification of polymers has become one of the most important procedures for the synthesis of new polymers. In industry, modification processes have found extensive applications in the synthesis of many polymers, produced on a large scale, such as nitrocellulose, acetylcellulose, other cellulose esters and ethers, poly(vinyl alcohol), etc. There is no doubt that in the future modification will acquire still greater importance in industry.

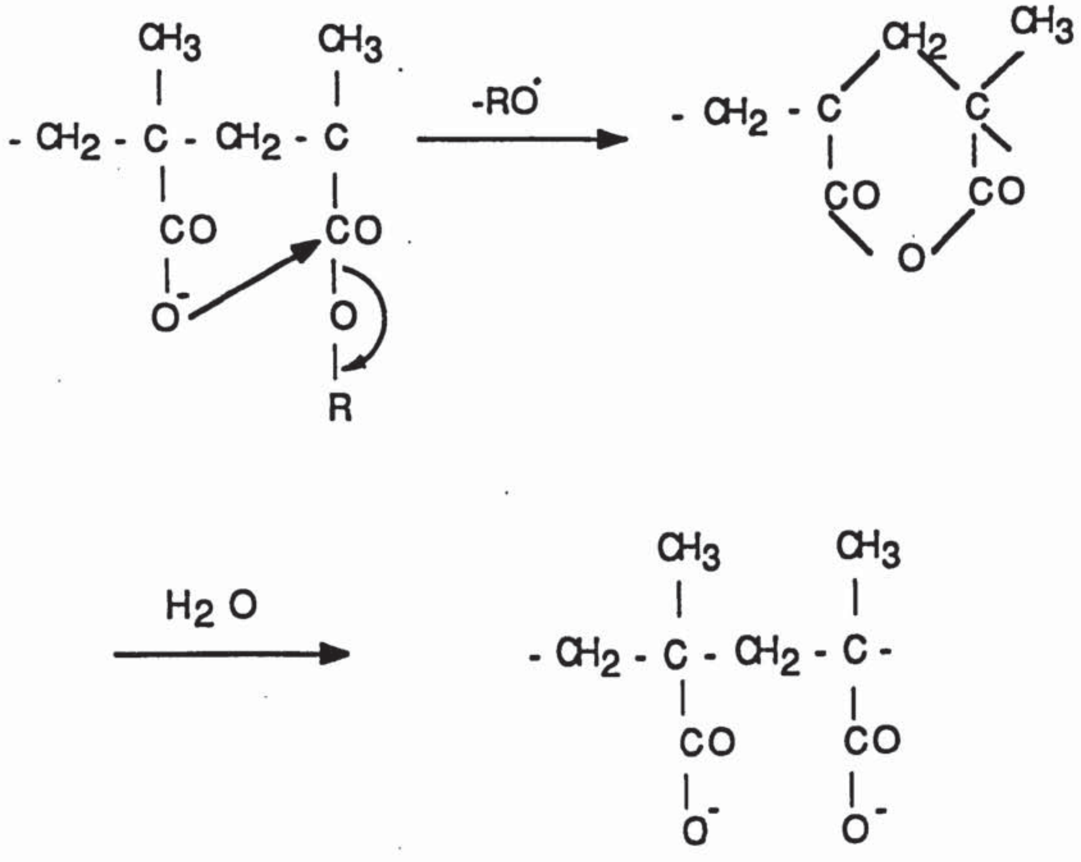
1.2.1 Characteristics of Polymer Modifications

A characteristic feature of the polymer modification processes is that all reactions of this kind do not as a rule go to completion and are accompanied by side reactions, which has a significant influence on the process and its results. This is caused by the influence of a number of physical and chemical factors, manifested in a specific manner in various polymer modification reactions. Such factors include the influence of the nature of the reagents and catalysts, the structure and degree of crystallinity of polymers, temperature, the nature of solvent, etc. Ultimately, the reactions result in a mixture of macromolecules of different structures, each of which contains different units in a wide variety of combinations. The final polymer is chemically inhomogeneous, or has a heterounit structure.

Consequently reactions in the polymer chain obey much more complex rules than the analogous reactions of low molecular weight compounds. The reason is that not only the structure of the reacting groups but also a number of specific influences, determined by the macromolecular nature of one or both reactants, play a significant role here. The chain effect, the effect of neighbouring groups and configurational, conformational, electrostatic, and supermolecular effects are usually considered to be the most important factors.

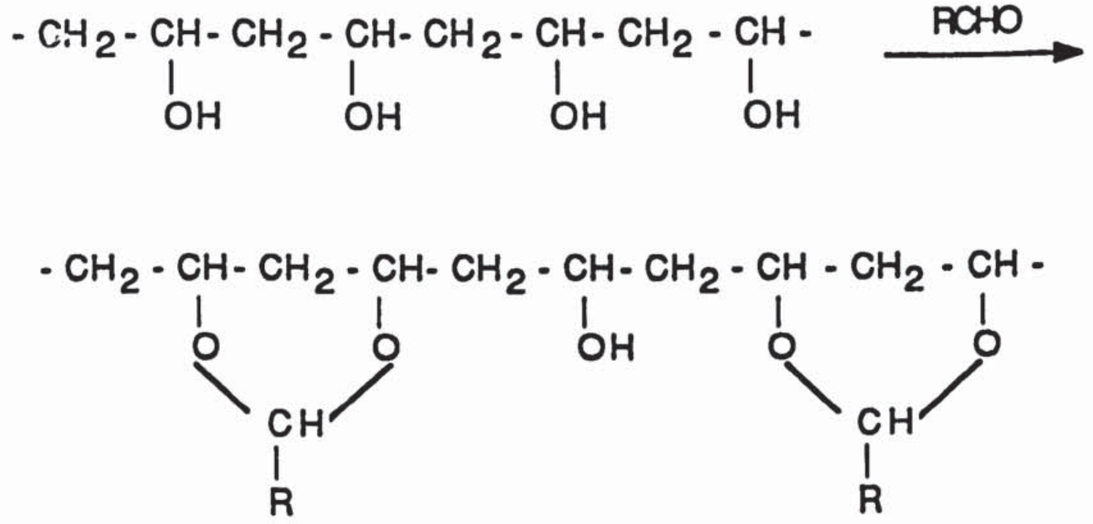
The relative steric disposition of different groups in the polymer chain has a significant steric or electrostatic (polar) influence on their reactivity. This influence can be shown both in the activation and deactivation by neighbouring groups. The increase of the rate of reaction under the influence of a neighbouring group, known as 'anchimeric assistance', is exhibited mainly when five- or six-membered rings can be formed. Thus it is observed for succinate esters. In exactly the same way, owing to the steric proximity of a neighbouring group, isotactic poly(methyl mehtacrylate) hydrolyses faster than the syndiotactic and atactic polymers, because in the isotactic polymer the neighbouring functional groups are distributed more favourably for their interactions with one another, leading to the formation of an intermediate cyclic anhydride. The distribution function of the compositions of polymers has been considered in a number of investigations[83,84].

As already stated, one of the causes of the formation of anomalous units in polymer modification processes is the influence of neighbouring units which have already reacted (the "neighbouring effect") [85]. A characteristic example of the acceleration of the reaction by the "neighbouring effect" is the hydrolysis of the methacrylate-methacrylic acid copolymer[85]:



Scheme1.6

However, as a result of the "neighbouring effect", not all the initial hydrolysis products in poly(vinyl alcohol) are converted into acetal by treatment with aldehydes. According to calculation[86], approximately 13.53% of the hydroxyls should remain unsubstituted or a statistical basis:



Scheme1.7

Examples are known in which different effects are superimposed[87]. Thus the alkaline hydrolysis of polymethacrylamide does not go to completion. The loss of reactivity of the amide groups occurs both as a consequences of their being blocked by two ionised carboxyl groups and as a result of the electrostatic repulsion by the latter of hydroxide ions which catalyse the hydrolysis.

Attention should be drawn to the fact that, in order to characterise fully the polymer formed as a result of chemical modification, it is important to establish not only the presence of anomalous units and their number but also their sequences in the polymeric molecules. Thus the physical properties of vulcanised rubbers are determined not only by the ratio of the cis- and trans-units but also by their sequences in the macromolecules.

When the causes responsible for the chemical inhomogeneity of modified polymer through the formation of heterounit polymers, are considered, it is important to recoginse that the physical structure of the initial polymer plays a by no means subsidiary role. Its role is particularly important when the reaction takes place in a two-phase system, for example, there is no suitable solvent for the reaction system or when one has to retain the initial form of the polymer, as, in the treatment of fibers and films. The tendency towards the manufacture of polymers with the greatest possible elimination of all solvents has been recently assuming an increasing importance. Together with economical and ecological causes, this is associated with the fact that reactions in solution are sensitive to the nature of the solvent and to the properties of the solution, which can be reflected in the degree of the conversion and in the structure of modification products formed.

The solubility of the polymer in the solvent is also significant, because, as the reaction proceeds, their composition changes very considerable with a corresponding changes of their properties. Incompatibility is due to phase separation and results in the localisation of the reaction on the surfaces of compact coils. The accessibility of the functional groups of the polymer to the low molecular weight reagent is thus limited by diffusion, which affects the degree of the modification reaction and the structure of the products[88].

Factors such as the supermolecular structure of the polymers, the orientation of the macromolecules in solution and the solubility and compatibility of the polymer and reagents in solution also exert a considerable influence on the result of reactions involving substitution in macromolecules. When the solid polymer is acted upon by liquid reagents, the rate and degree of conversion is dependent on the rate of diffusion of the liquid reagent into the bulk of the polymer structure. Diffusion takes place more readily in an amorphous polymer and in the amorphous region of crystalline polymers. Macromolecules in the amorphous regions react faster and are more fully acted upon by reagents than macromolecules in the ordered crystalline state.

In view of the above characteristics of the reactions in heterogeneous system, in chemical modification reactions, attempts should be made to dissolve the polymer or at least to obtain them in swollen state since this will facilitates the reaction process.

As mentioned above, the polymer modification reactions never go to completion and they therefore lead to polymers containing the initial units as well as the final and intermediate unit. The study of the resulting polymers shows that the substituents are usually distributed at random along the polymer chain[85]. Fig1.2 diagramatically illustrates the distribution of substituents for different degrees of substitution.

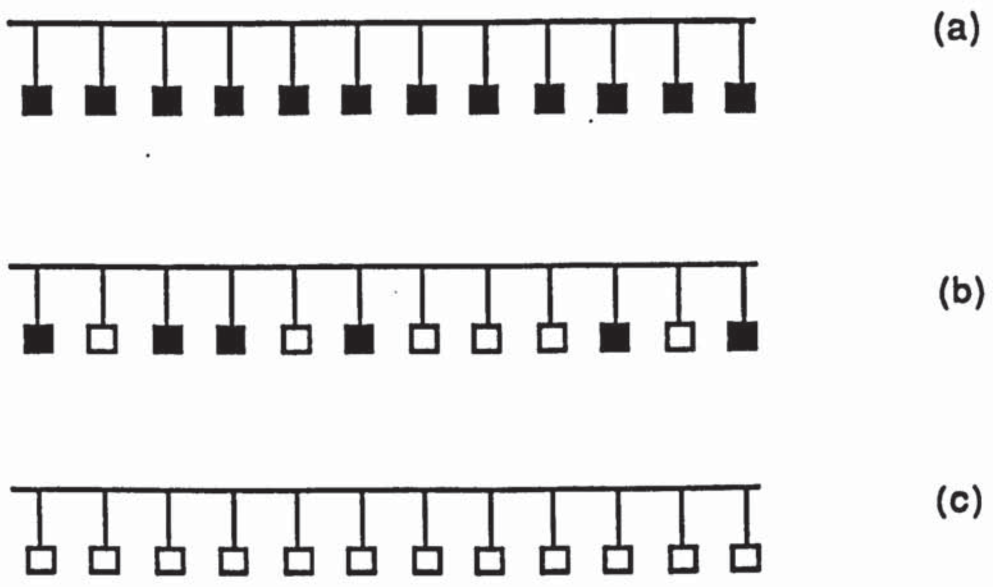


Fig1.2 distribution of substituents in modified polymer: (a) initial polymer; (b) partial and irregular substitutions; (c) ideal case of complete substitution.

1.2.2 The Heterounit Structure of Modified Polymer

One of the characteristic features of polymer modification reactions as shown in Fig1.2 is the incompletness of the reactions, as a consequence of which some of the initial functional groups do not react and remain in the unchanged state. At the same time, there is a possibility of a wide variety of side reactions, which also lead to the formation of units of a different chemical type and give rise to chemical heterogeneity in the macromolecules, usually referred to as the heterounit structure of polymers[90].

The heterounit polymer obtained by various modification reactions differ due to the presence of anomalous units in the macromolecule, i.e., they are chemically heterogeneous. Thus unlike the structure of a normal formula $-\!\!(\text{-M-})\!\!-_x$, the macromolecule of a heterounit polymer contains anomalous unit A:

$$\text{-M-M-M-A-M-M-}$$

Consequently the structure of a heterounit polymer can be represented by the following formula in general cases:

$$-\!\!\left[(\text{-M-})_m\text{-}(\text{-A-})_n\right]\!\!-_x$$

where M is normal unit, A is anomalous unit, m and n are coefficients indicating a statistical distribution of the units in the macromolecule of the heterounit polymer obtained by modification[90].

Fig1.3 illustrates the potential reactions - both normal and secondary reactions - which can occur in the modification of polymers.

1.2.3 Typical Chemical Modification Reaction of Polymer

Polymer modification processes are based on the use of a variety of chemical reactions, each of which has a specific influence on the structure of the polymer chain structure. Fig1.4 illustrates shcematically the changes in the structure of macromolecules occuring as a result of the modification reactions.

It is necessary to bear in mind not only the chemical nature of the reaction but also the way in which it is carried out (in solution, in melt, in emulsion state or in solid), since the result of the process depends on the physical state of the polymer.

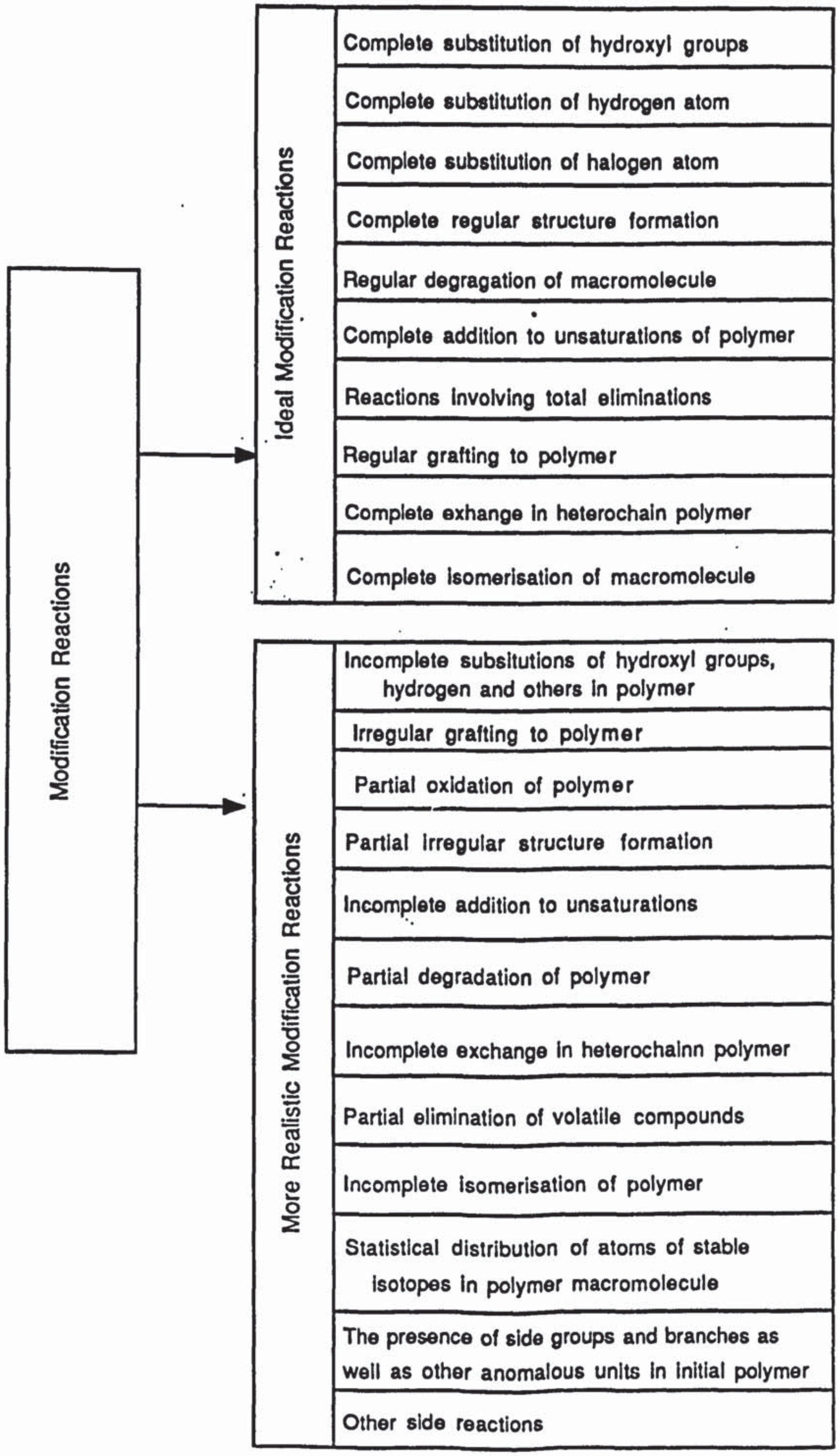

Modification Reactions
Ideal Modification Reactions
Complete substitution of hydroxyl groups
Complete substitution of hydrogen atom
Complete substitution of halogen atom
Complete regular structure formation
Regular degragation of macromolecule
Complete addition to unsaturations of polymer
Reactions involving total eliminations
Regular grafting to polymer
Complete exhange in heterochain polymer
Complete isomerisation of macromolecule
More Realistic Modification Reactions
Incomplete subsitutions of hydroxyl groups, hydrogen and others in polymer
Irregular grafting to polymer
Partial oxidation of polymer
Partial irregular structure formation
Incomplete addition to unsaturations
Partial degradation of polymer
Incomplete exchange in heterochainn polymer
Partial elimination of volatile compounds
Incomplete isomerisation of polymer
Statistical distribution of atoms of stable isotopes in polymer macromolecule
The presence of side groups and branches as well as other anomalous units in initial polymer
Other side reactions

Fig1.3 Potential and actual modification reactions of polymer.

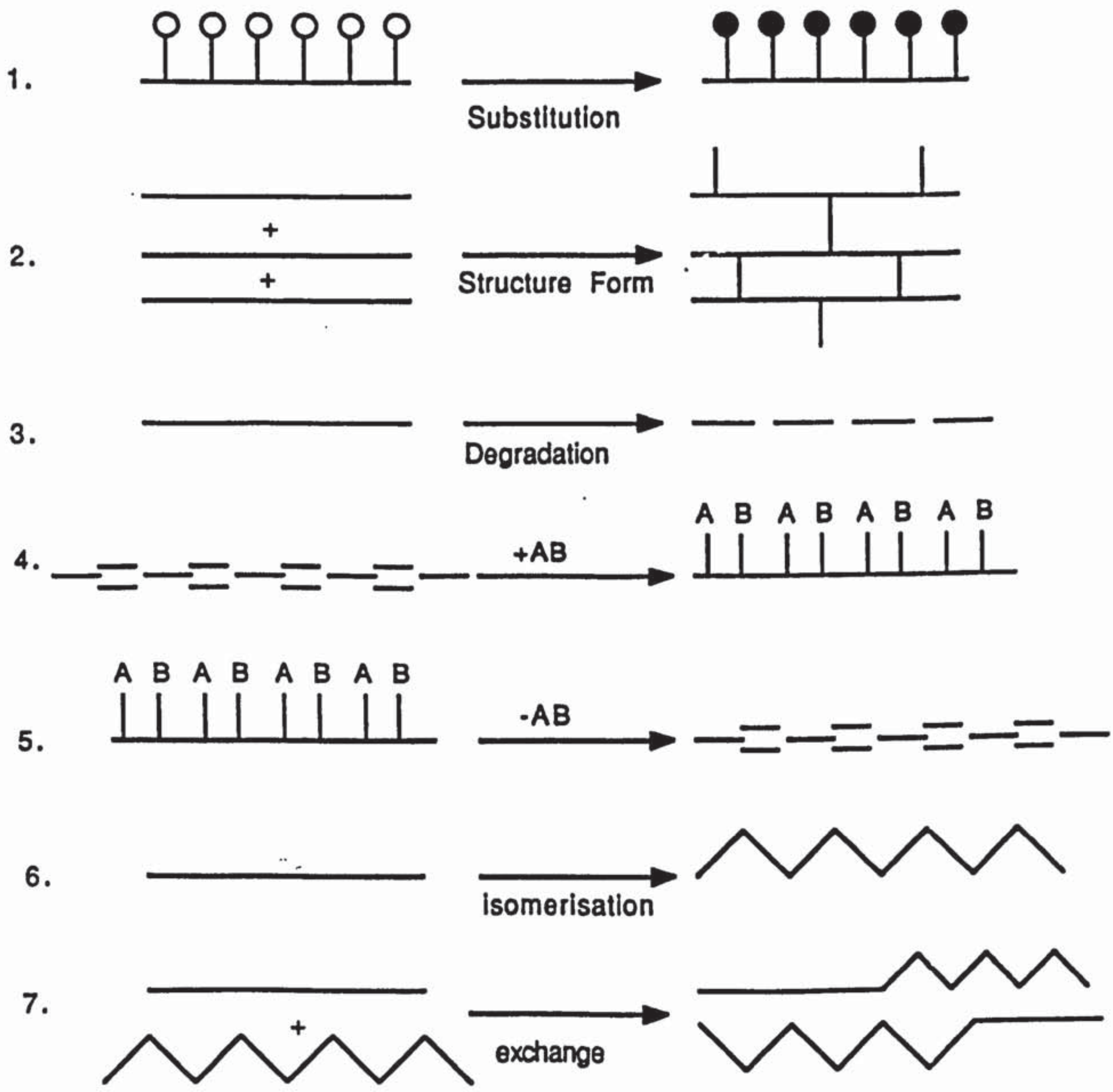


Fig1.4 Schematic representation of changes in the structures of macromolecules after modification reactions

1.2.3.1 Polymer Modification Reactions Involving Substitution[90-93]

The substitution reaction can be divided into the following groups:

(1) substitution of a hydroxyl by acyl and alkyl

groups as well as inorganic acid ;

(2) substitution of a hydrogen atom by a hetero group;

(3) hydrolysis of derivatives of polymeric carboxylic acids;

(4) substitution of a halogen and other groups for a hydroxy- or amino-group;

(5) oxidation of side group.

These reactions have certain common features consisting in the nature of the alteration of the structure and the properties of the resulting polymers. We may note that the replacement of a hydroxyl group in a polymer by an acid residue or by an alkyl makes the polymer soluble in organic solvents. In contrast, the substitution of an alkyl group in an ether by a hydroxy group impairs the solubility of such a polymer in organic solvents.

The majority of polymer modification reactions involving substitution belong to the type designated as polymer-analogue reactions. Their characteristic feature is retention of the main macromolecular skeleton, only the nature of the substituents along the main chains being changed.

1.2.3.2 Polymer Modification Reactions Involving Structural Changes[93-96]

The modification of polymers by cross-linking formation is a process in which the macromolecules pass through a stage with branched structures and are ultimately converted into infusible and insoluble three-dimensional structures. In the first stage of this process the polymer interacts with the cross-linking agent, which leads to the formation of bonds between the macromolecules. A typical industrial structure formation process is vulcanisation, which is widely used to give improved physical behaviour in the processing of polymers, particularly elastomer. Vulcanisation imparts to rubbers resistance to wear and mechanical action and resistance to solvents.

1.2.3.3 Polymer Modification Reactions Involving Degradation[97,98]

Degradation is a process directly opposed to structure formation and is used to decrease the molecular weight of polymers or to introduce reactive sites on the polymeric molecules. Degradation is particularly used in the processing of natural polymers such as rubber and cellulose to reduce the molecular weight. With the aid of degradation by mechanochemical and oxidative methods, it is possible to obtain polymers with the optimum size of the macromolecules for specific applications.

1.2.3.4 Polymer Modification Reactions Involving Addition[99~102]

Addition reactions are used as for the modification of polymers when polymers contain multiple carbon-carbon bonds. Thus halogen, hydrogen, and other reagents may be added to them. At the same time, there also exist metathetical reactions resulting rearrangement of carbon-carbon double bonds in unsaturated cyclo-olefines.

1.2.3.5 Polymer Modification Involving Elimination[103~106]

The elimination of low molecular weight substances such as water, alcohol, hydrogen halides, etc. is one of the important modification methods. As a result of the elimination of low molecular weight compounds, a double or triple bond or even a heterocycle is formed in the macromolecule of the initial polymer.

.As an example of a reaction leading to the formation of a heterounit polymer in such processes, we may quote the dehydochlorination of PVC:

$$\left[\mathrm{CH_2{-}CHCl}\right]_x \xrightarrow[x=m+n]{-\mathrm{HCl}} \left[\mathrm{CH_2{-}CHCl}\right]_m\left[\mathrm{CH{=}CH}\right]_n$$

This process is of considerable importance in the study of PVC degradation and stabilisation (see section1.1), and it is the subject of present work(see section1.3).

1.2.3.6 Polymer Modification Involving Isomerisation[107,108]

In all the polymer modification reactions considered above, the composition of the resulting product differs from the initial polymer. However, in the isomerisation reactions, the modificatin process proceeds without changing the elementary composition of the polymer and consists in the rearrangement of the basic skeleton on the macromolecule. Isomerisation requires a vigorous treatment, such as heating, sometimes combined with high pressure.

1.2.3.7 Polymer Modification Involving Exchange[109]

All the modification methods explained so far led to the formation of polymers in which the chain structure of the initial polymer remained unchanged in most cases. In contrast to this, there is another modification method, based on the breaking and reforming of bonds in the polymer backbone, which leads to the formation of a polymer containing the units of both initial polymers in the chain. The macromolecules of this polymer therefore constitute in essence a block copolymer whose overall composition corresponds to that of the initial mixture of polymers. Consequently, this modificatica method also constitutes an unusual copolymerisation method.

1.3 Objectives of the Present Work

From the discussions of the previous sections (see section1.1), it is clear that an enormous amount of research has been done on the chemical behaviours of PVC under variours conditions. The major concern of PVC research has been to improve the thermal and photochemical instability. To achieve this, a range of additives have been developed, such as stabilisers, plasticisers, lubricants, etc. This has however been associated with a growing concern because the migration of additives, especially stabilisers and plasticisers from PVC composition. There are basically two ways to prevent the loss of additives from the polymer bulk:

(1) by binding additives to the polymer chain.

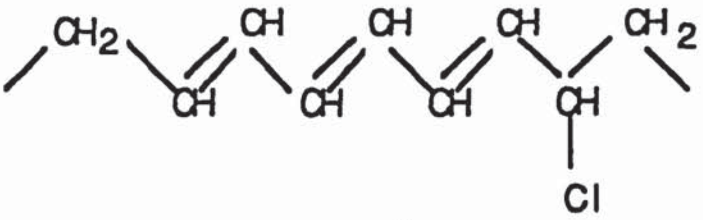
(2) by synthesizing polymeric additives. This kind of additive may have increased substantivity in certain polymers.

The first method requires the additives to have multifunctional structures. The second method requires the additives to be polymerisable. In most polymers, this leads to compatibility problems.

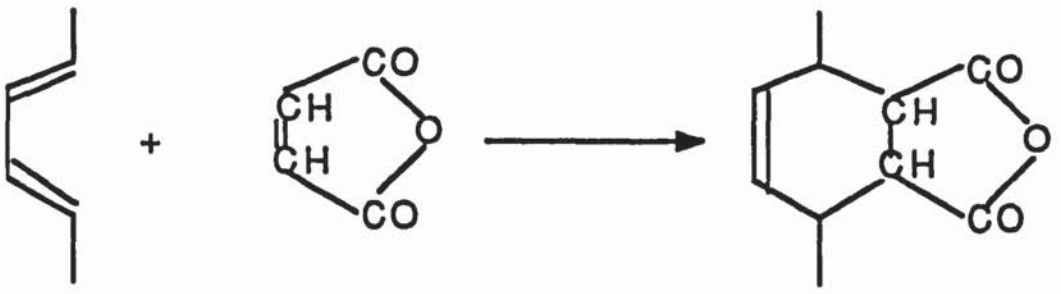
The purpose of this work is to find a method which can be used to form non-migrating additives for PVC. Possible ways of achieving this available to us are as follows:

1. Reactions through Conjugated Double Bonds in PVC

Publications dealing with the degradation of PVC (in an inert atmosphere) mostly agree that a dehydrochlorination process and the simultaneous formation of conjugated double bond sequences in the polymer chain are involved. As a result, PVC after degradation would very much look like:



It is known[110,111] that the Diels-Alder reaction between conjugated dienes and dienophiles is one way of removing conjugated unsaturation in degraded PVC (see Scheme1.8). Obviously, because the degraded PVC contains many conjugated dienes on its chain, it is possible to bond an additive which may act as dienophile for the degraded PVC chain. The additive chosen to experiment with for this purpose is maleic anhydride and its butyl ester, which is a plasticiser:



Scheme1.8

2. Radical Initiated Modification of PVC

It is also known that free radicals play an important role, which may lead to the evolution of HCl and the cross-linking of PVC molecules. The formation of reactive sites provides us an opportunity to add an additive molecule to the PVC chain.

To summarise, the objectives of this work are to modify PVC and degraded PVC by maleic derivatives through Diels-Alder reactions and radical initiated bonding of additive to PVC. It is anticipated that these methods can also be extended to other additives, especially stabilisers.

Chapter2
Experimental Procedures

2.1 Materials

General reaction reagents and solvents were supplied by the chemical store of the Chemistry Department. They were used without additional purifications.

PVC resin: PVC homopolymer was supplied by BP. It is unstabilised, free of additives. Gel permeation chromotography of the polymer gave the following characteristics (provided by supplier):
Mn=30750, Mw=75000, Mz=188000, Mv=67400
Mw/Mn=2.44, Mz/Mn=6.11

Maleic Anhydride (MA): purchased from Aldrich Chemicals Co., 99%, bp 200 ^{0}C, FW 98.06.

Dibutyl Maleate (DBM): purchased from Aldrich Chemicals Co., 98%, bp 281 ^{0}C, FW 228.29.

Azobisiobutyronitrile (AIBN): purchased from Alfa Chemicals, 99%, mp 107 ^{0}C, FW 162.21.

2.2 Degradation of PVC

2.2.1 PVC Powder Degradation

1.0g PVC resin was heated in a flask sealed with a stopper which has nitrogen inlet and outlet. The flask was preheated to different temperatures according to experimental requirement. PVC resin was kept in the flask for different duration of time under nitrogen protection. The degraded PVC resin was collected for later use.

The conversion to unsaturation was calculated by the weight decrease in the sample.

2.2.2 PVC Film Degradation

A piece of PVC resin film (see section2.4 for the making of film) weighed was added into a flask sealed with a stopper which has nitrogen inlet and outlet. The flask was preheated to different temperatures according to experimental requirement. PVC resin was kept in the flask for different periods of time under nitrogen protection. The degraded PVC film was collected for later use.

The conversion to unsaturated PVC was calculated by the weight decrease in the sample.

2.3 Processing

PVC formulations for processing were prepared by carefully mixing accurately weighed additives with the polymer. Processing the formulations was carried out in a RAPRA torque rheometer which is used as reactor simulator.

A processing temperature 170^0C and rotor speed of 60 rpm were used throughout and the variation of torque during processing was recorded. The chamber, the rotors, the ram and the hopper were carefully cleaned prior to the sample processing. The processed composition was rapidly withdrawn, chilled in clean water, dried and stored in sealed plastic bags.

2.4 Film Preparation

2.4.1 Film Casting

PVC resin or degraded PVC powder was dissolved in THF to form a clean solution. The solution was carefully poured onto a shiny glass surface. The glass was kept in a dasecater for 48 hours. The the film was removed from the glass surface and dried. Uniform films of thickness $3x10^{-3}$ cm were chosen for later usage.

2.4.2 Compression Moulding of Film

The processed polymer was powdered by gentle abrasion and the powder was compression moulded between stainless steel plates at 170 ^{0}C for 75 seconds, using 550 PU grade cellophane to separate the polymer from the metal. The polymer was preheated between the plates of the press for 45 seconds prior to application a ram pressure of 100 kg cm^{-2}. Uniform films of thickness $3x10^{-3}$ were chosen for analysis, thermal and UV exposure studies.

2.5 Modification Reactions

2.5.1 Reaction between Degraded PVC (dPVC) and DBM

Degraded PVC was added to dibutyl maleate in a round bottomed reaction flask, according to the ratio of 1:10 by weight. The mixture was heated to different reaction temperatures according to experimental requirement, and dPVC dissolved in DBM. The reaction was carried out for period of time under continous stirring and nitrogen protection. After reaction, methanol was used to separate polymer and dibutyl maleate. The polymer sample was then dried and kept for analysis.

2.5.2 Radical Initiated Reactions between PVC and DBM

PVC (undegraded) was added into dibutyl maleate in a round bottomed reaction flask according to the ratio of 1:10 by weight. The mixture was heated to the required temperature. The the radical initiator AIBN was distributedly added into the reaction system in three separate steps during the whole reaction time. The reaction was carried out under continous stirring and nitrogen protection.

The polymer sample was separated from the rest by treating the whole mixture with methanol. The polymer sample was then dried and weighed for later use.

2.6 Extraction

Polymer films and powdered polymer were suitably contained in a cellulose thimble and subject to continous extraction. The apparatus used was a Soxhlet extraction unit. The extraction was carried out over various time by hexane as extracting solvent to remove dibutyl maleate.

2.7 Colour Development

Colour formation in PVC films of uniform thickness $3x10^{-3}$ cm, was expressed as the total colour difference relative to the calibrated reference films cast from a 5% THF solution of the unprocessed polymer. The measurement was carried out using the MEECO Colormaster ModeV in transmission. Readings were obtained by using red, green and blue filters and the total colour differences ΔT was calculated by following equation:

$$\Delta T = \sqrt{\Delta R^2 + \Delta G^2 + \Delta B^2}$$

where ΔR, ΔG and ΔB are the differences between the instrument readings of the test sample and the reference sample using the red, green and blue filters respectively.

Chapter3
Results and Discussions

According to the objectives of the project (see section1.3), the work was carried out in two steps. The first step was to investigate the degradation of PVC under controlled condition in order to follow the development of polyenes in the PVC molecules. The second step was to carry out modification reactions on PVC itself and degraded PVC under controlled conditions. The results and discussions are given in this chapter.

3.1 Thermal Degradation of PVC

3.1.1 Polyene Development

The degradation of PVC is marked by the evolution of HCl and the development of colour (see section1.1). The discolouration has been attributed to the formation of conjugated double bonds (polyene sequences), which are formed through a 'zipper' mechanism[112]. It is thought that discolouration occurs when the sequence length exceeds about five units[113].

The development of polyene sequences during thermal degradation has been often followed by ultraviolet-visible spectroscopy. It is well known that absorption increases with increasing conversion of PVC. A typical UV spectrum of degraded PVC is shown in Fig3.1.

It has been shown that every peak in the absorption spectrum

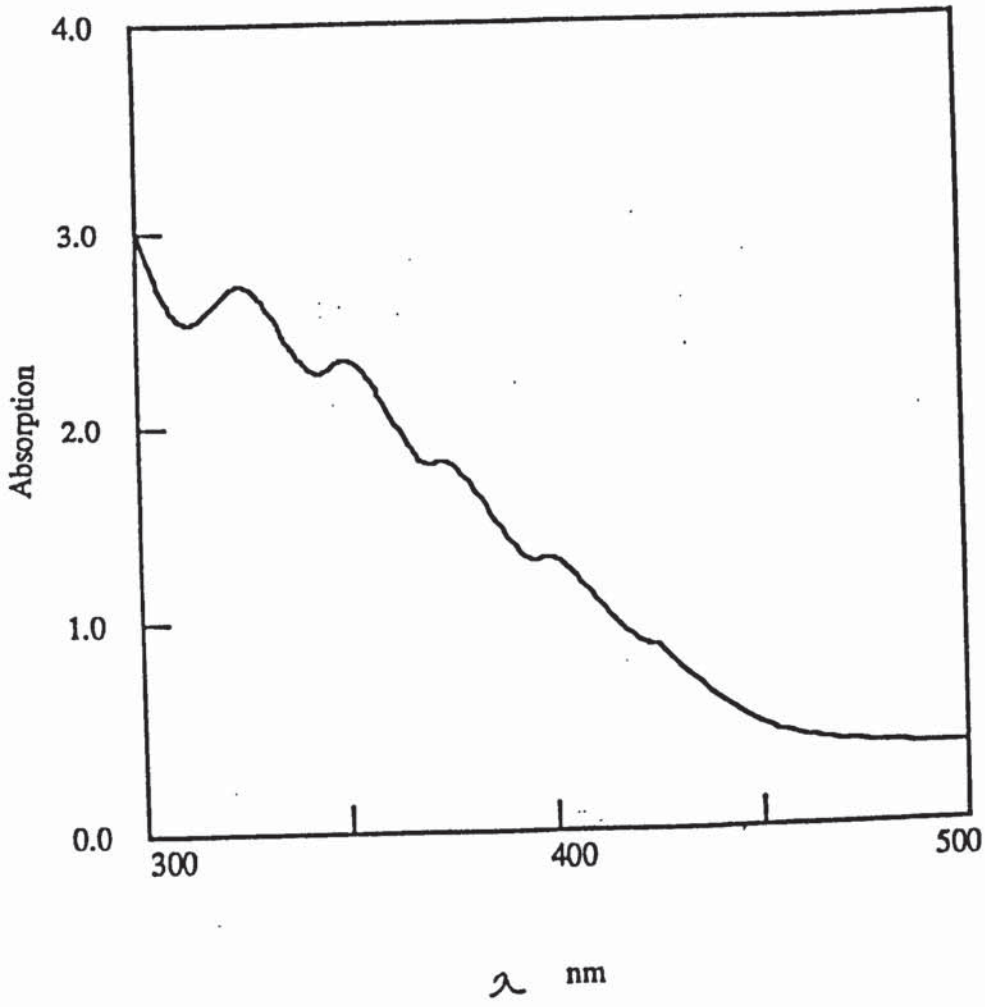


Fig.3.1 UV spectrum of dPVC of 1.0% conversion at 180 C

could, to a good approximation, be related to a certain sequence length of polyene[114], although it has been argued that the peak absorption might be the result of the superimposing effect of different length of polyenes[115]. The logrithm of polyene concentration versus degradation time is shown in Fig3.2. It can be seen that at the early stage, the change is linear, but deviates from linearity at later stage of degradation, which may indicate the onset of secondary reactions between degraded PVC molecules[116] (also see section1.1.2). The concentration of polyene denoted by Hn in Fig3.2 was calculated from equation(3.1)[117]:

$$Hn=(A/m)\varepsilon_n \qquad (3.1)$$

where n is the number of conjugated double bonds, A is the absorption of the examined solution of the wavelength corresponding to the polyene sequence of length n, m is the concentration of the examined solution in monomeric units mol/litre, and ε_n is the molar extintion coefficient related to the polyene sequence length n. ε_n was calculated[117] by

$$\varepsilon_n=10,000 + (n+1)x20,000 \qquad (3.2)$$

where ε_n is expressed in litre/mol cm[118]. From the values of Hn, it is possible to calculate an average polyene sequence length, n. A value of about 6 was obtained in our case.

From Fig3.2, the ratio:

$$H_2:H_4:H_6:H_8:H_{10}$$

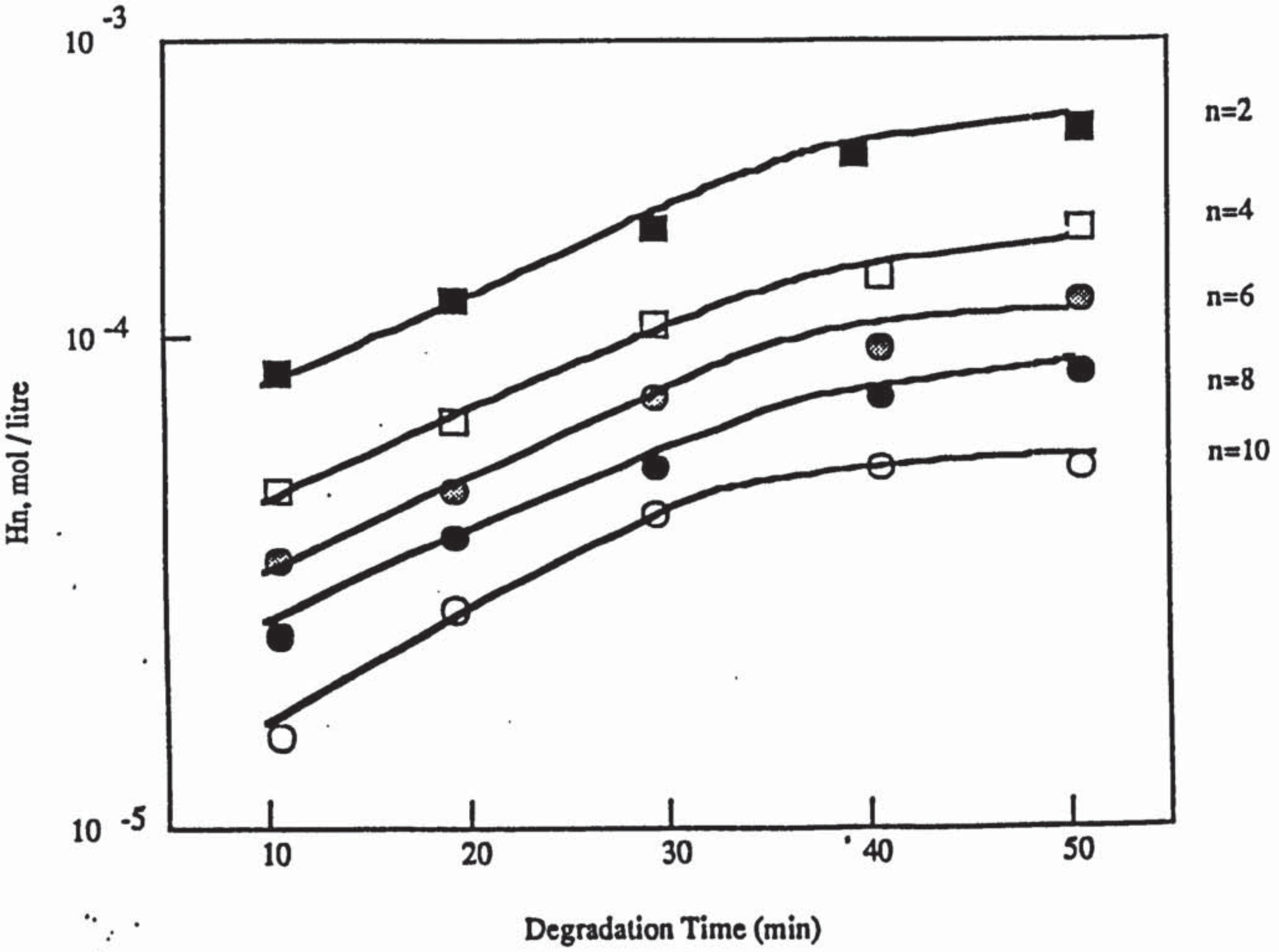


Fig.3.2 Log concentration of even polyene lengths as a function of degragation time for PVC degraded at 180 C under nitrogen protection

was calculated to show the relative amounts of polyenes of different length. It was found that this value was almost constant throughout the whole period of degradation. A typical ratio was found to be:

$$H_2:H_4:H_6:H_8:H_{10}=1: 0.6 : 0.5 : 0.3 : 0.1$$

This phenomenon suggests that polyenes with different length have an equal probability to develop. Although the total numbers of the polyene of different length increases with the degradation time, their relative concentrations remain constant. This ratio also shows that the longer the polyene length the less amount there is in the mixture. The ratio is shown in Fig3.3.

The colour formation in PVC films was measured by the total colour difference (see section2.7), the result is shown in Fig3.4.

3.1.2 Simulation of Different Length Polyene Formation

From section3.1.1, it is clear that the development of polyene sequence has a feature of randomness. Many attempts have been made to study the kinetic process of PVC degradation[119]. Because of the complexity of the process, it is only possible to study some very comprehensive aspects. As regards to the development of different length polyenes, it is almost impossible to analyse this by conventional kinetic method. Therefore, it is necessary to seek other ways, especially computer simulation.

The degraded PVC has the following molecular structure(see section1.1.1):

$$\sim CH\text{-}CH{=}CH\text{-}[CH{=}CH]_n\text{-}CH{=}CH\text{-}\underset{\substack{| \\ Cl}}{CH}\text{-}CH_2\sim$$

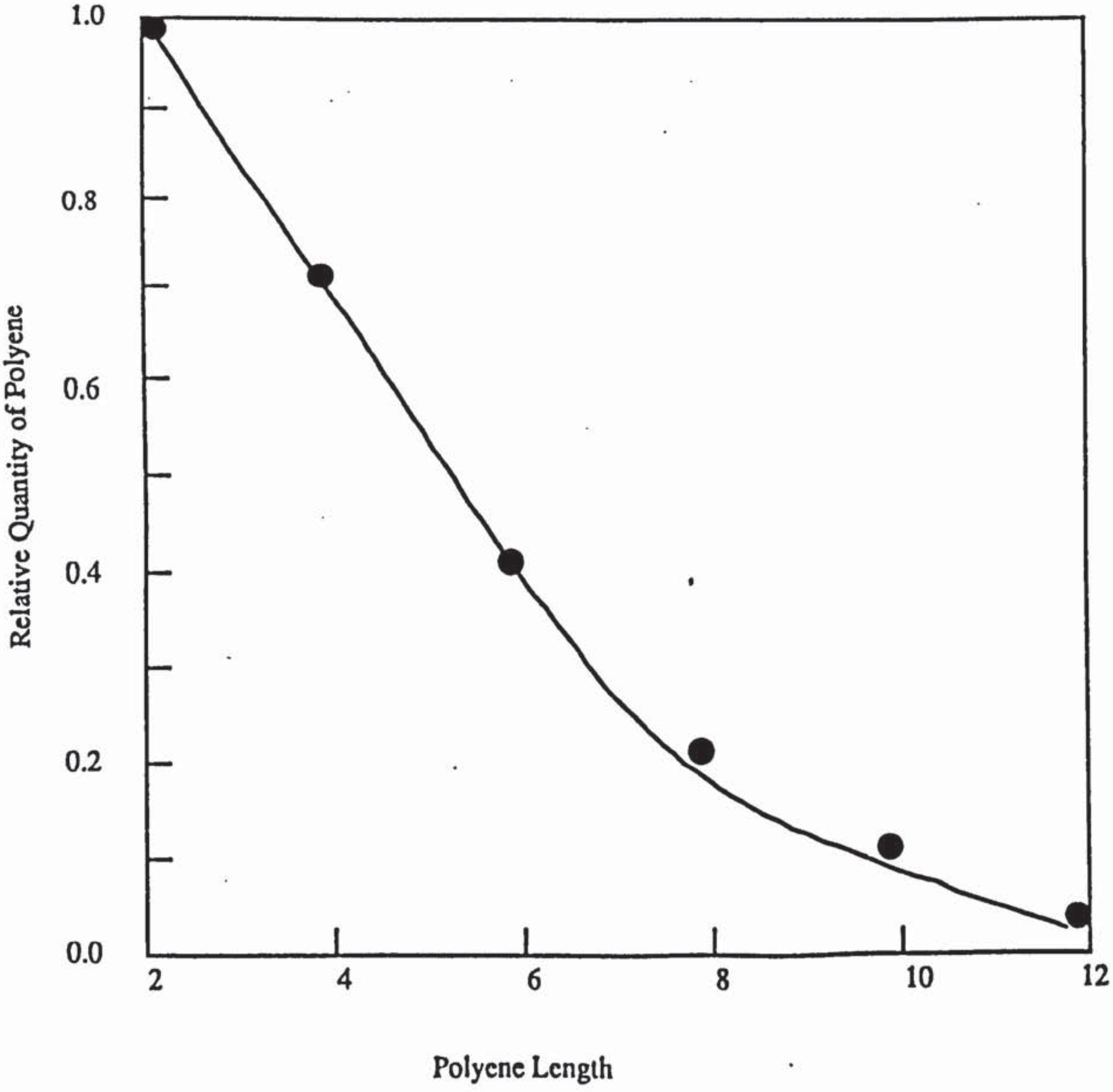


Fig.3.3 Relation between the length of polyene and the relative quantity

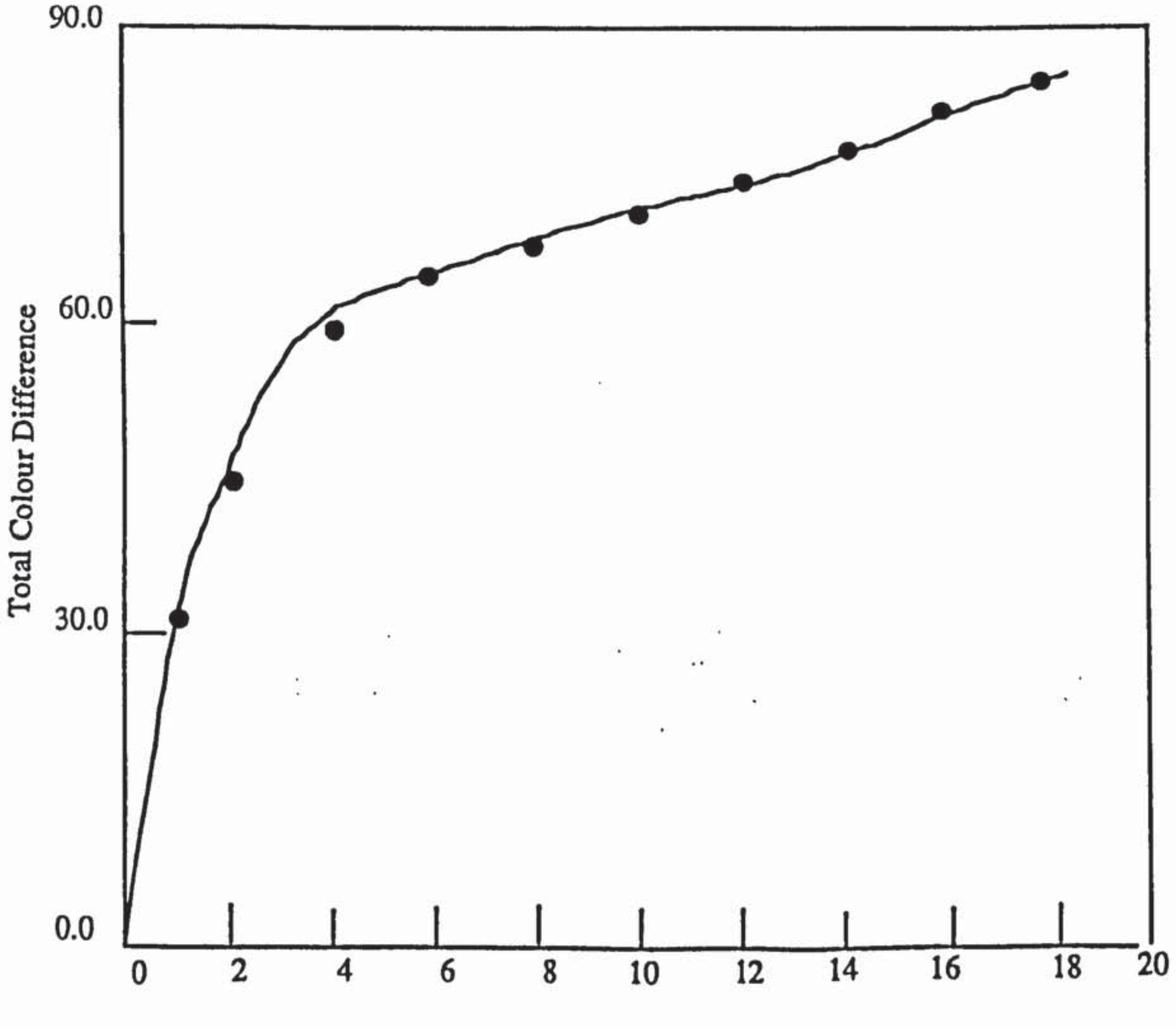
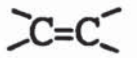
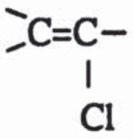


Fig.3.4 Effect of Processing on the formation of colour when PVC was processed at 180°C

If this molecule is considered as a polymer itself instead of a product of PVC degradation, then it is a copolymer between the following two monomers:

 and 

Although there has been no direct experimental copolymerisation of acetylene and vinyl chloride, it is still possible in principle. Such as the development of polyenes during degradation of PVC is random, the same is true for the copolymerisation of the above two monomers. If we carry out a numeric simulation of copolymerisation, then it is possible to have some knowledge of the distribution of the polyenes in the PVC molecule.

It is known[120] that when a monomer M1 is copolymerised with another monomer M2 the relation between the composition of the copolymer and the composition of the monomer mixture is given by

$$\frac{dm1}{dm2} = \frac{M1(r_{12}M1+M2)}{M2(r_{21}M2+M2)} \qquad (3.3)$$

where

m 1	moles of monomer M1 entering copolymer
m 2	moles of monomer M2 entering copolymer
M 1	moles of monomer M1 in monomer mixture
M 2	moles of monomer M2 in monomer mixture
r_{12},r_{21}	monomer reactivity ratios

The monomer reactivity ratios r_{12}, r_{21} , for any monomer pair are the ratios of the rate constants of different propagation reactions as defined by the following equations:

$$\sim M\dot{1} + M1 \xrightarrow{k_{11}} \sim M1M\dot{1}$$

$$\sim M\dot{1} + M2 \xrightarrow{k_{12}} \sim M1M\dot{2}$$

$$\sim M\dot{2} + M1 \xrightarrow{k_{21}} \sim M2M\dot{1}$$

$$\sim M\dot{2} + M2 \xrightarrow{k_{22}} \sim M2M\dot{2}$$

$$r_{12} = k_{11}/k_{12} \qquad (3.4)$$

$$r_{21} = k_{22}/k_{21}$$

The above processes can be generalised as

$$\sim M\dot{i} + Mj \xrightarrow{k_{ij}} \sim MiM\dot{j} \qquad (i,j=1,2)$$

$\sim M^{\bullet}$ represents a polymer chain ending in a radical derived from monomer M. The reactivity ratio is

$$r_{ij} = k_{ii}/k_{ij} \qquad (3.5)$$

which is based on the experimental data k_{ii} and k_{ij}.

The probability of a given chain end Mi reacting with a particular monomer Mj is given by

$$P_{ij} = \frac{1}{1 + r_{ij}x_i/(1 - x_i)} \qquad (3.6)$$

where x_i is the mole fraction of Mi in the monomer mixture feed. Because there is only two monomers,

$$P_{ii} + P_{ij} = 1 \qquad (3.8)$$

Then it is possible to devise a numerical procedure to carry out the simulation of copolymerisation.

The basic idea in the simulation is to relate the probability of copolymerisation propagation with the random number generated by computer. Therefore, the mechanism of using this random number facility decides the statistical reliability. The flow chart[121] for simulation is shown in Fig3.5. This flow chart only simulates a single copolymer chain, and does the calculation on this single chain. If the number of units is large enough (usually larger then 1,000 units), the result is considered reliable[121]. However, if we modify the procedure in Fig3.5, so that, instead of synthesizing a single chain of fixed units each time, we synthesize a fix number of chains whose units are randomly generated by the computer in every run, then, we use random number to generate the number of every chain and simulate the growth of the chain. The modified version is shown in Fig3.6. The implementation of it is given in Appendix A.

Fig3.7 gives the molar ratios of different length polyenes in the copolymerised "degraded PVC" molecules, which is very similar to

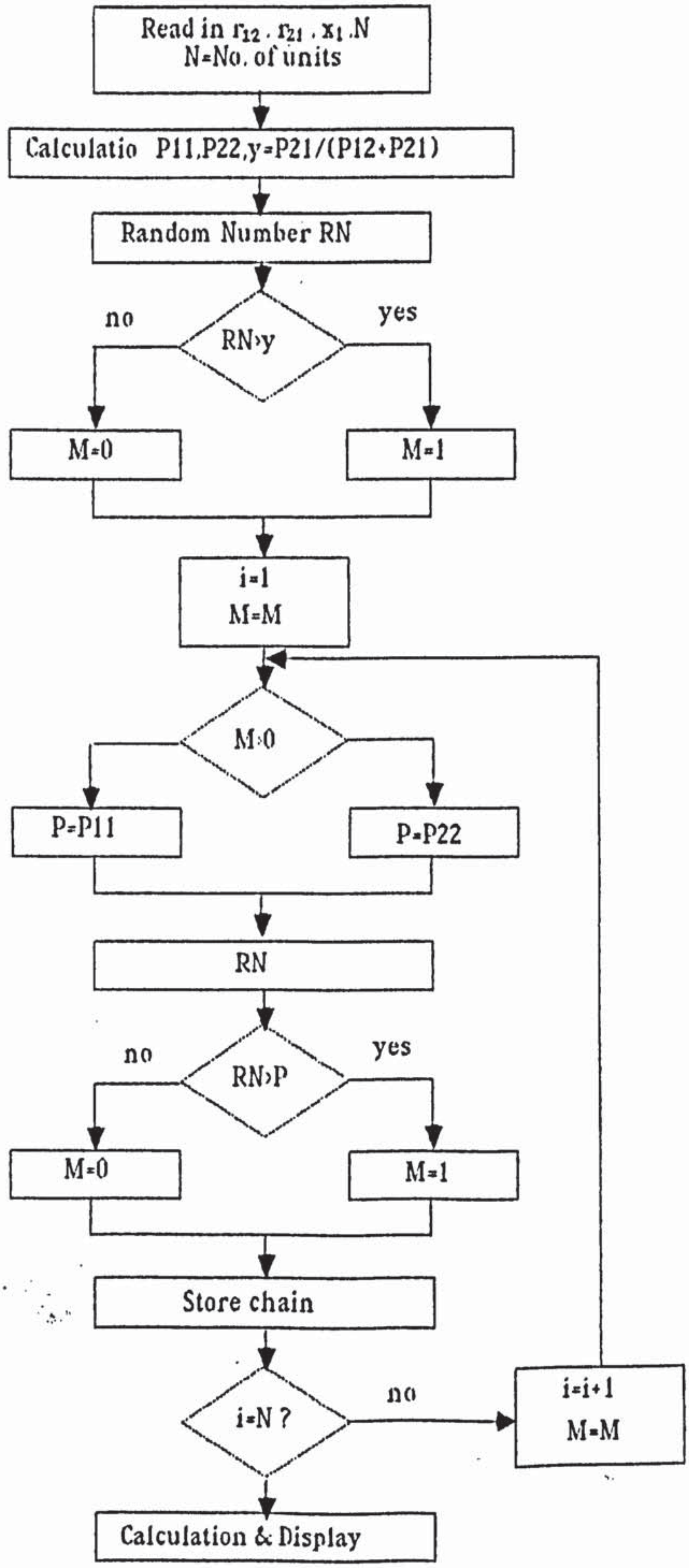


Fig.3.5 Flow chart for simulation of binary copolymerisation

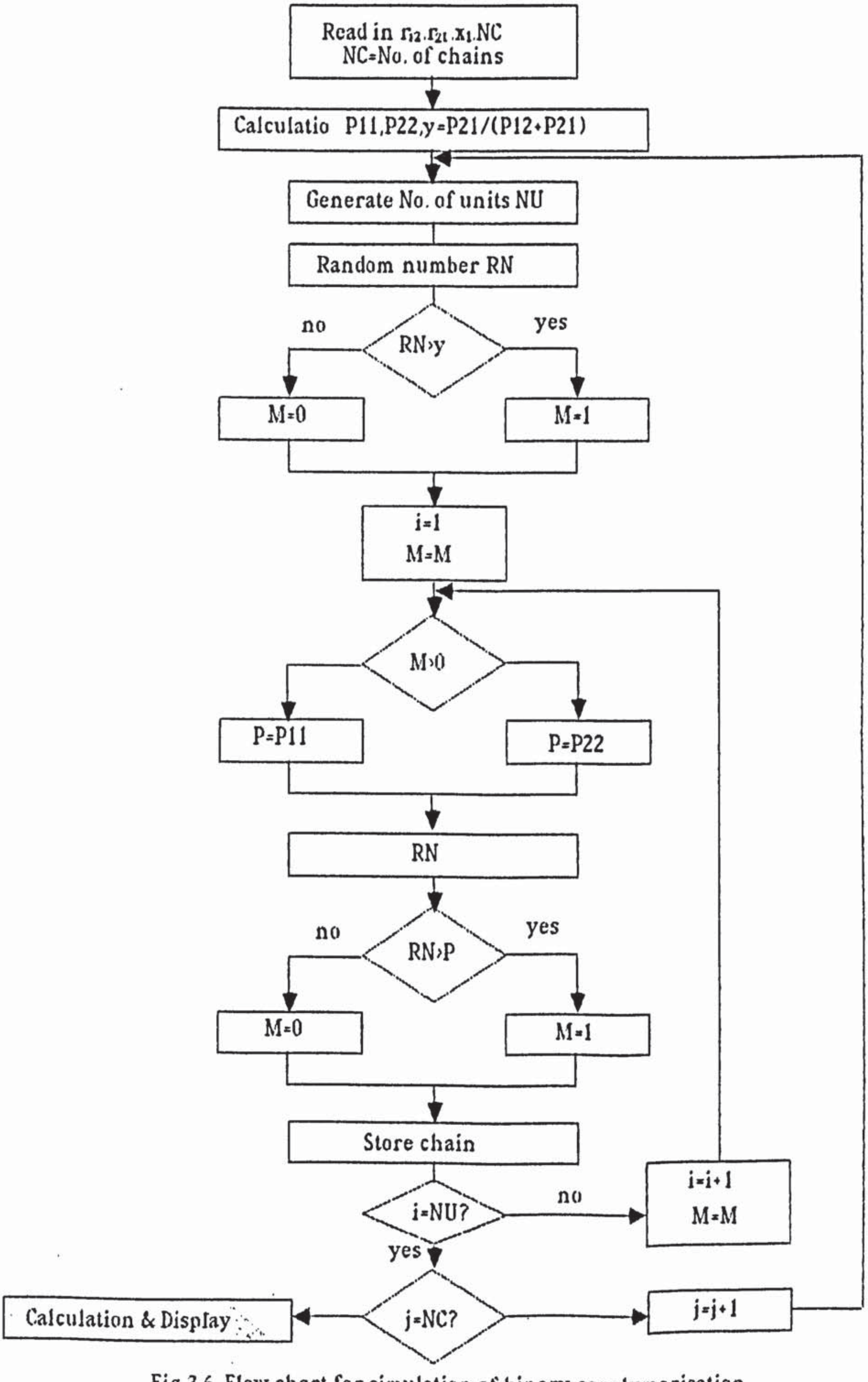


Fig.3.6 Flow chart for simulation of binary copolymerisation

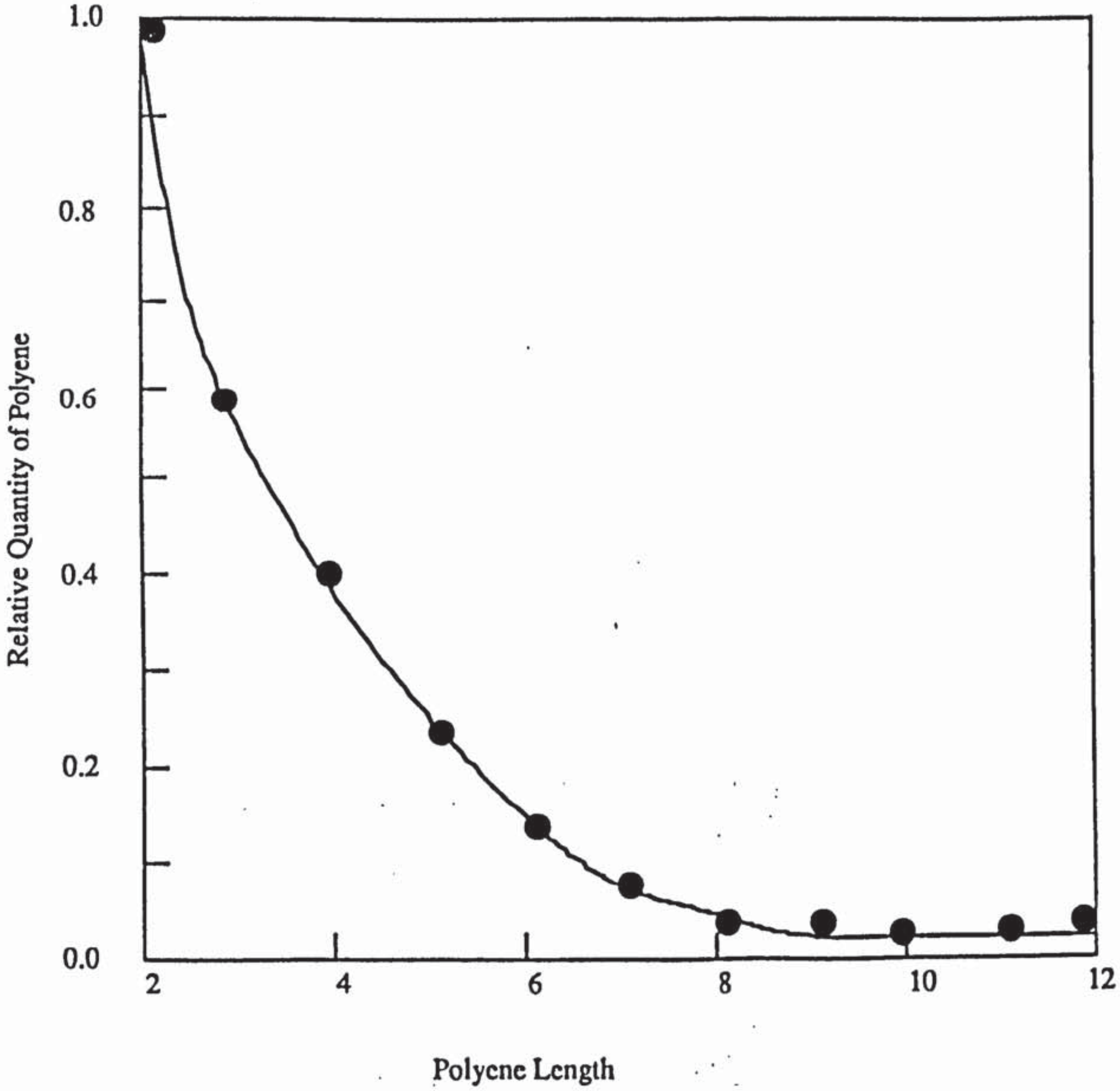


Fig.3.7 Simulated relation between the length of polyene and the relative quantity

the experimental observation for experimental degradation (see Fig3.3).

A few more points about the simulating program. This program stores the polymer chain as it is produced so that not only the copolymer composition can be calculated, but also the distribution can be computed. Furthermore, visual display of the chain is also possible.

When analytical treatment of a problem is almost impossible, it is useful to apply the techniques we used (often referred to as a Monte Carlo simulation). The application of computer and numerical methods in physical sciences is a rapidly expanding technique and polymer science is no exception[122,123]. In fact, because the mathematical treatment of polymeric problem involves large magnitude of numbers, it seems more necessary here than in many others. The above treatment is only an elementary attempt to use the techniques available, further work is obviously possible.

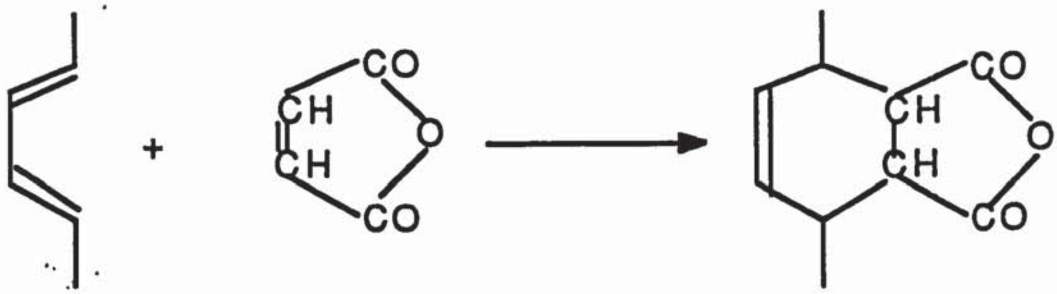
The method used above is not a direct treatment of PVC degradation. It is clear that more convinsing simulation of polyene growth should be based on degradation process. However, because the complexity and the lack of kinetic data of PVC degradation reactions, the work done in this project tried another way. The validity of it, of course, remains to be argued. The important idea here is not the simulation itself, but the idea of viewing degraded PVC as a copolymer. Compared with the degradation process of PVC, the copolymerisation suggested above is more controllable, which may provide a method to study PVC degradation more accurately.

3.2 Chemical Modification of PVC by Plasticisers

The modification of PVC and its degraded products were carried out separately. They will be discussed in the following sections.

3.2.1 Reaction of Degraded PVC with Dibutyl Maleate

It was seen above (see section 1.1.2) that conjugated unsaturation and dienophile can react:



by the Diels-Alder reaction. When PVC is degraded for a long time, it is observed that there is cross-linking in the polymer. There might be several explanations for this, but the reactions between two PVC chain through the Diels-Alder mechanism cannot be ruled out (see section1.1.2). That is to say, the conjugated unsaturation developed on PVC has the potential to react with a dienophile. This is the basis of our modification reaction.

Fig3.8 gives the UV spectra of degraded PVC after reacting with dibutyl maleate for different period of time, which shows the

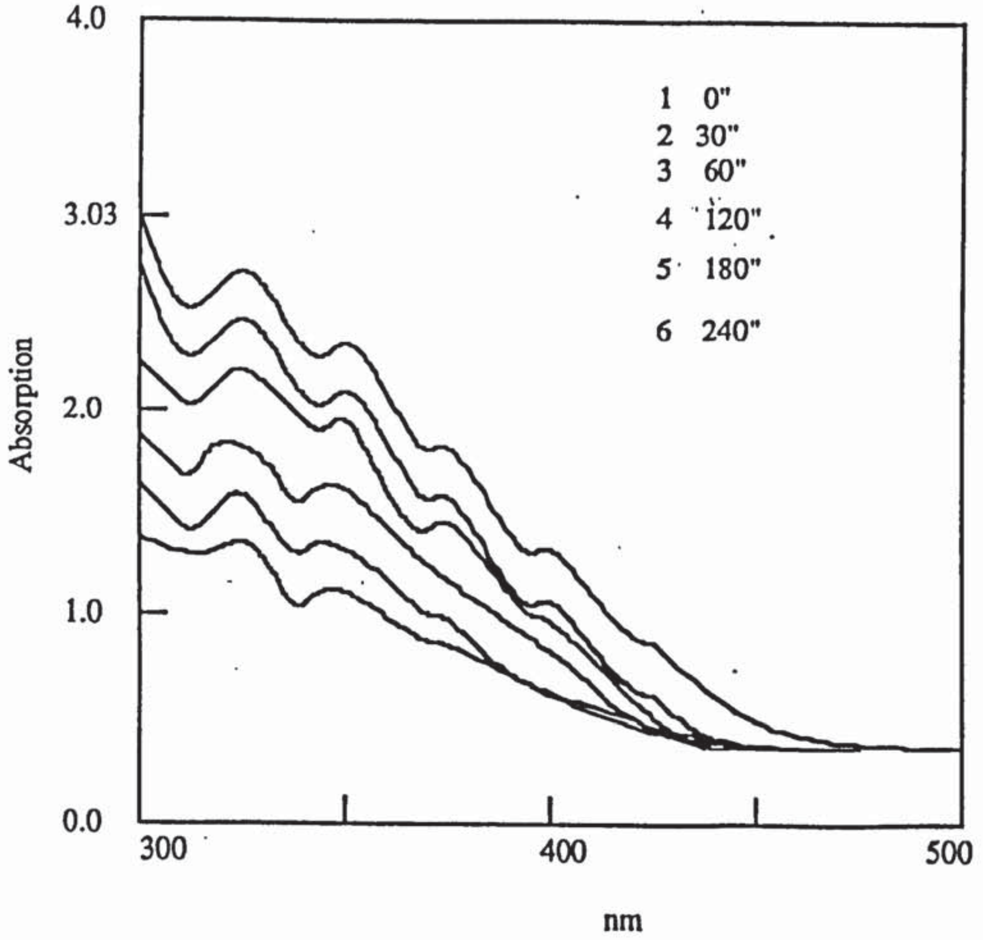


Fig.3.8 Decreases of UV absorption of polyenes due to the reaction with malcic anhydride at 120°C for different reaction time

reduction of conjugated unsaturations as a result of the reaction between them and DBM.

Fig3.9 shows the IR spectra of degraded PVC and the degraded PVC which was reacted with DBM at 120^{0}C for 3 hours and then extracted by hexane for 24 hours. It can be seen that the degraded PVC reacted with the maleate has a clear absorption at 1740 cm^{-1}, which indicates the existence of carbonyl group in PVC molecule. It can also be noted that both spectra have an absorption at 1600 cm^{-1}, which is usually due to the absorption of unsaturation, but the unreacted sample has a more intense unsaturation absorption than the reacted one. This suggests reaction between the conjugated unsaturation of degraded PVC and the maleate. However, due to the nature of the Diels-Alder reaction, there is always isolated unsaturation left. As in other polymer modification reactions (see section1.2), the reaction cannot go to completion for various reasons. In the reaction between dPVC and DBM, the incompletness of the reaction may be expressed in Scheme3.1. This may explain why the reacted sample still has absorption at 1600 cm^{-1}, and the reaction can not go to completion.

Fig3.10 gives the result of colour change of degraded PVC as a result of reaction with dibutyl maleate. It is clear, as the reaction time increases, the colour reduces. But, after certain time, it reaches an equilibrium level.

From the appearance of the sample, it was observed that the degraded PVC was rigid and deeply colour, and after reaction with

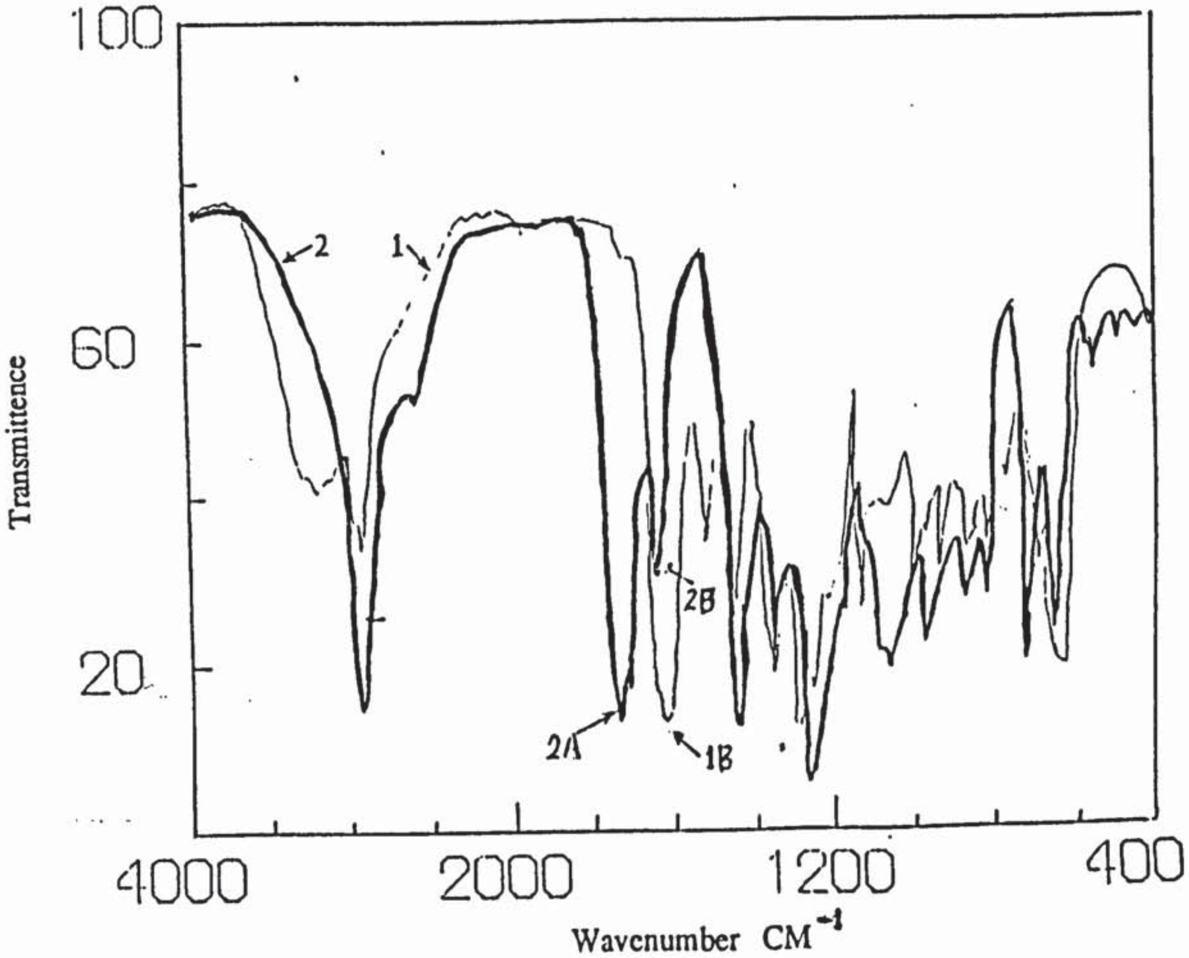


Fig3.9 Comparison of IR spectra of dPVC and dPVC reacted with DBM
1: dPVC, 1B: 1600 cm
2: dPVC after reaction(120 C for 3 hrs), 2A: 1740 cm 2B: 1600 cm

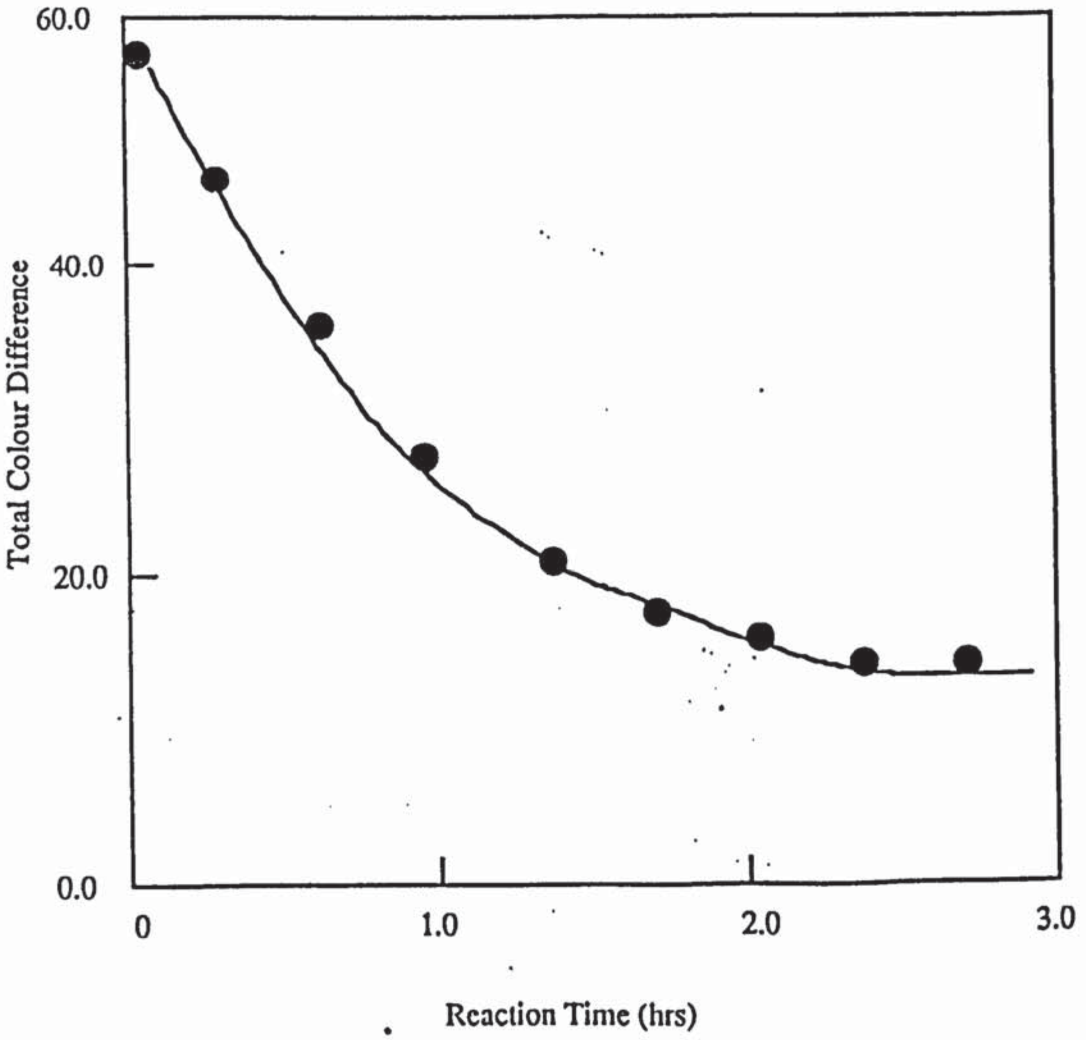
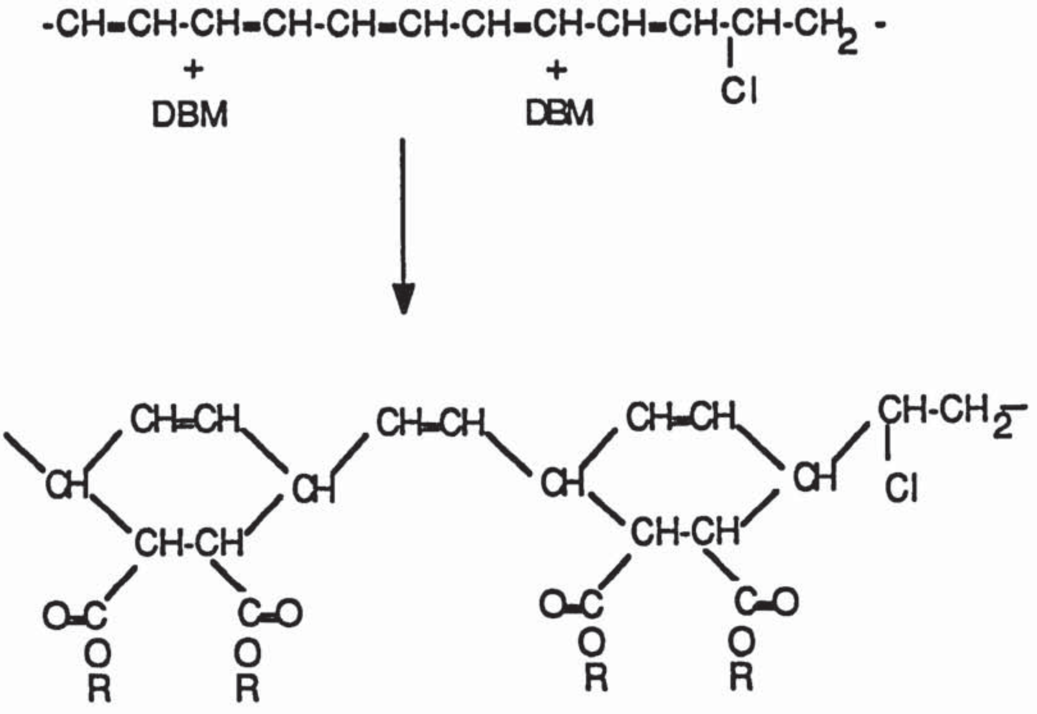


Fig. 3.10 Reduction of total colour difference of dPVC with the time of reaction with dibutyl maleate at 120°C

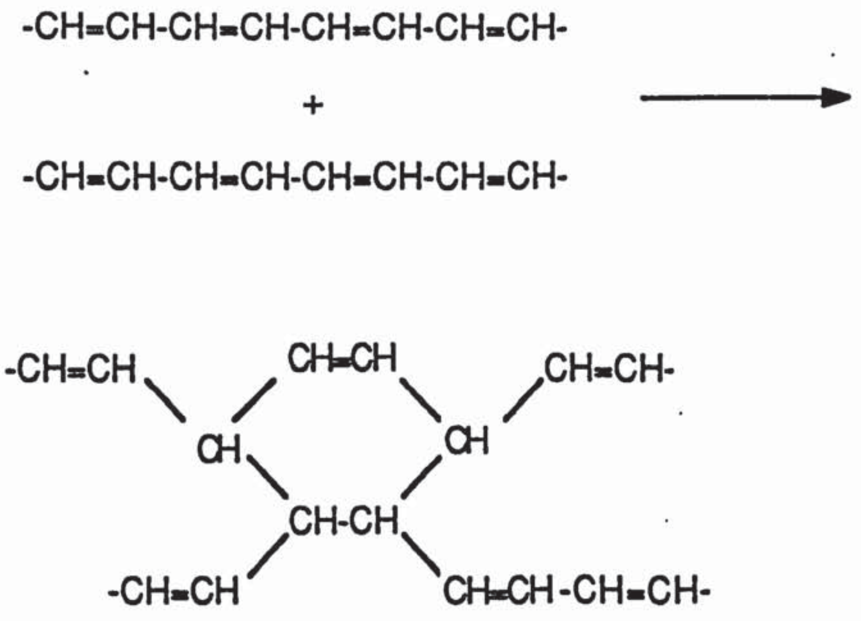
DBM and the removal of unreacted DBM from PVC sample, it became soft and less coloured. The amount of DBM reacted to dPVC reached 20~30%.



R = n-Butyl group

Scheme3.1

The extent of reaction is lower than expected. Apart from the reason presented in Scheme3.1, i.e., the production of isolated double bond, which is not able to react with dibutyl maleate further, other factors , like the reactions between two degraded PVC molecules:

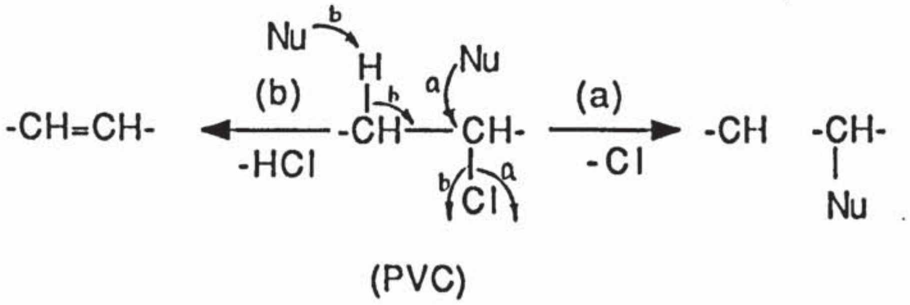


may also be considered. In fact, secondary reactions between degraded PVC molecules were observed experimentally, which lead to the formation of cross-linking.

Although as the experiment has shown that it is possible to bind additives to degraded PVC through Diels-Alder reaction, the double bond left in the product is not desirable when the product is to be used as additive in polymer.

3.2.2 Radical Initiated Modification of Undegraded PVC by DBM

It has long been believed that the nucleophilic substitution of chlorine in PVC is a possible way to introduce modifying groups into PVC, but this reaction is extremely difficult in view of the ease of elimination of HCl which then leads to the degradation of PVC. By using some special reagents[124] (for example, N,N-dialkyldithiocarbomate), it is possible to achieve nucleophlic substitution (see Scheme3.3), but the cost of the special reagents makes it less favourable in practical applications.



Scheme3.2

From the discussion earlier (see section1.1.1), we know that radicals are involved in the degradation process. If we could intentionally generate radicals in PVC and allow the radicals to react with added reagents before it degrades, then modification might be achieved. This technique has been used successfully in polyolefine[125]. It is the purpose of this section to see whether PVC may also be modified in this way.

When we choose a radical intiator, it is necessary to take into account of the efficiency and the working temperature so that the maximum initiation effect can be obtained while the temperature is not high enough to cause severe degradation of PVC. There are many radical initiators with different working temperatures[126]. We know that PVC starts to degrade at temperature above 120 ^{0}C. Therefore, the working temperature of chosen initiator must be lower than that. From the available literature, it seems that AIBN is a potential candidate since its useful initiation temperature range is 40~60^{0}C[127], although it was also reported to be higher than this[126].

The procedure outlined in Fig3.11 was carried out to ensure that after reaction there is no unreacted dibutyl maleate was left in the polymer.

Fig3.12 gives the result of the reaction between PVC and DBM initiated by AIBN and shows that the amount of DBM taken by PVC at different temperature for certain period of time. It can be seen that as the temperature increases, the amount of reaction increases too, but at higher temperatures, the increase is not as large as at lower temperatures. The reaction time is also important. From Fig3.12, it can be seen that, after certain time of reaction, the extent of reaction does not increases any further, and the highest amount of dibutyl maleate taken by PVC only reached about 30.0% of PVC by weight (at 90^{0}C). The proposed mechanism of this modification reaction is presented in Scheme3.3, but according to this free radical reaction mechanism, the extent of reaction of DBM with PVC should theoretically be much higher. One reason for this lower uptake actually observed by experiment is probably due to the neighbouring

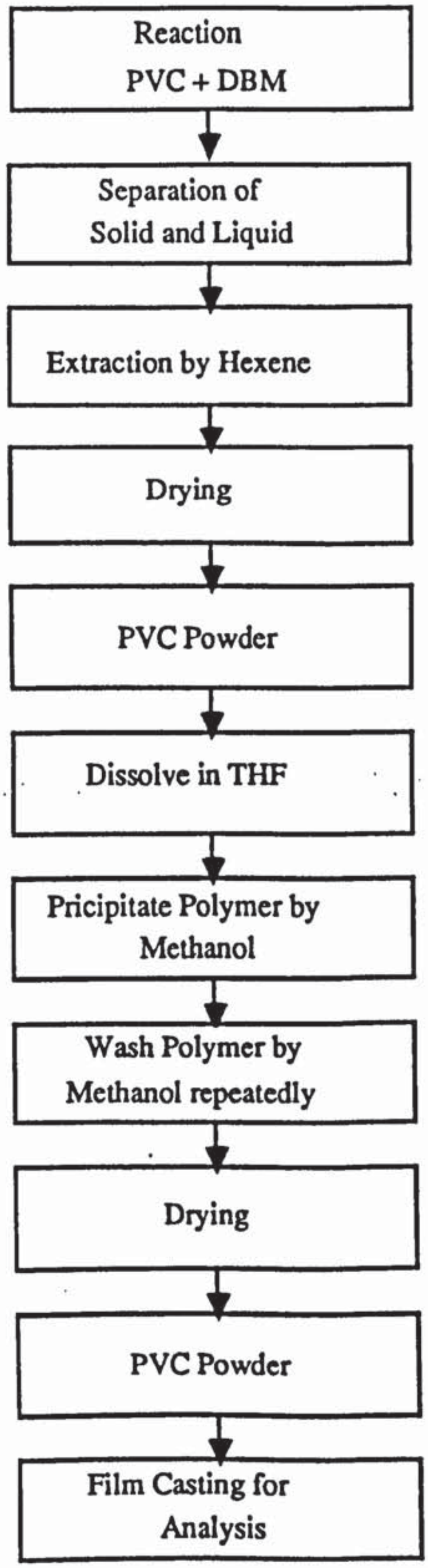


Fig.3.11 Procedure for Purification of Modified PVC

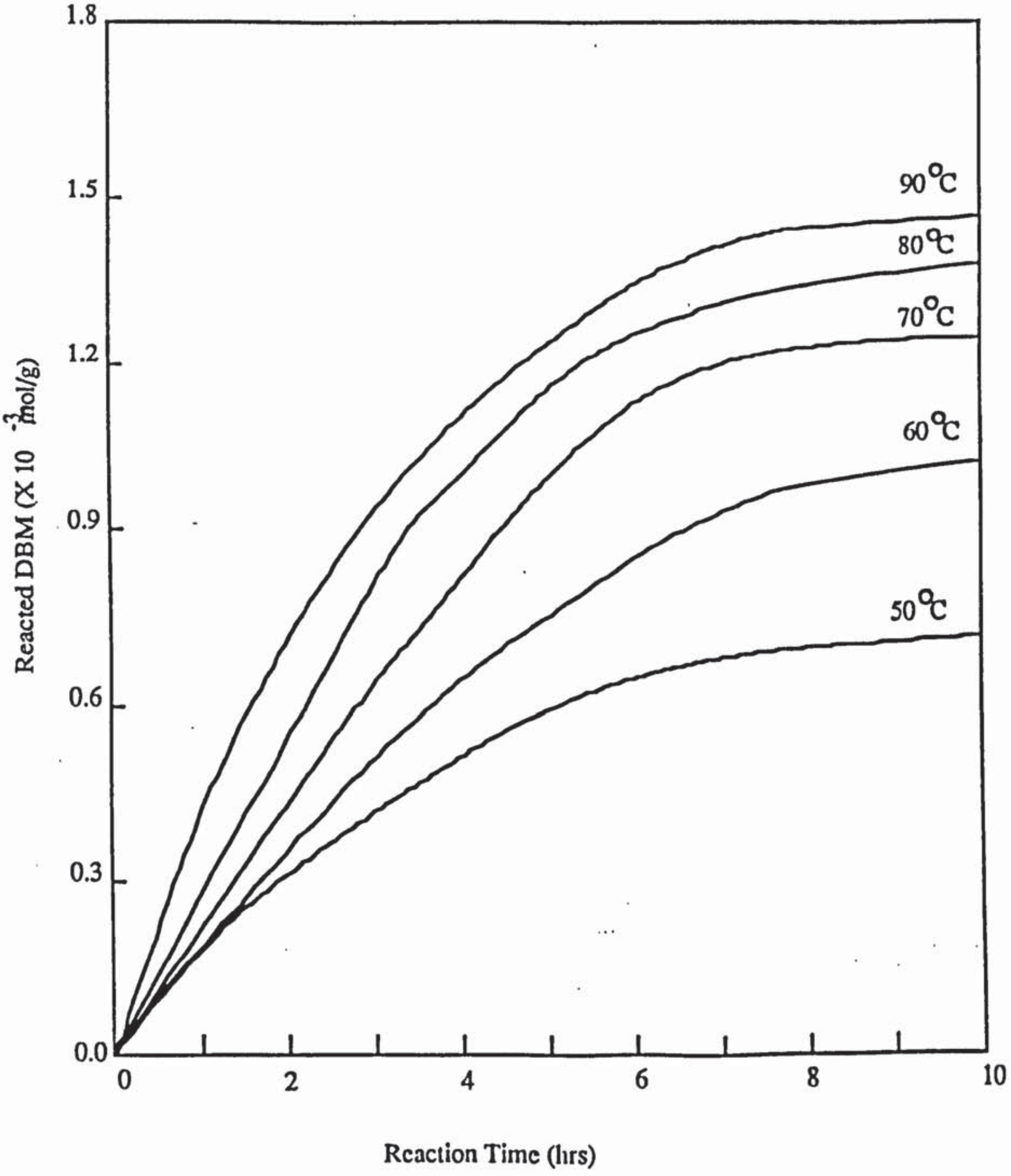
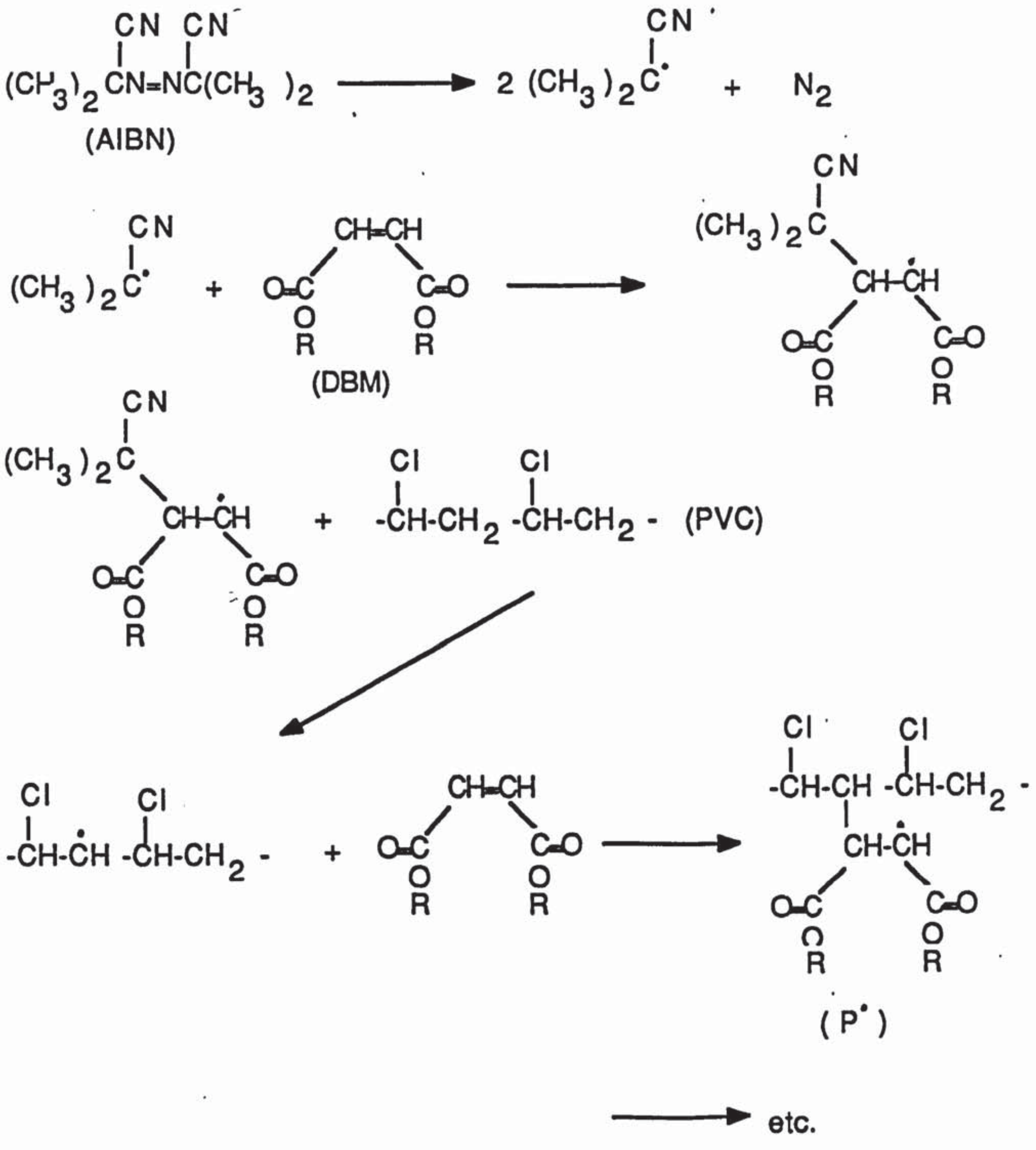
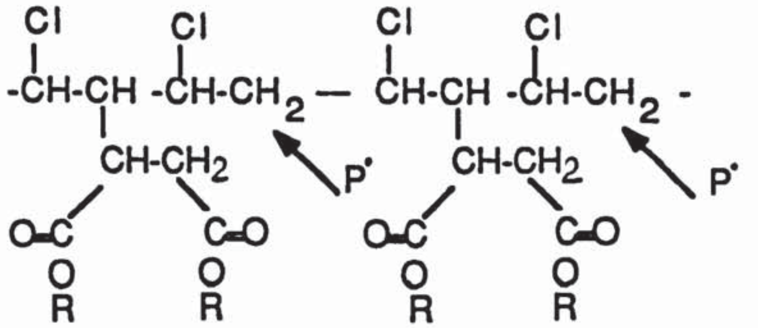


Fig.3.12 Temperature influence on the reaction between PVC and DBM initiated by AIBN (reaction temperature displayed on the curve)



Scheme3.3

group effect during the modification reaction, which is a common phenomenon in the polymer modification. This effect is usually considered to be caused by steric interference which limits the extent of polymer modification reaction (see section1.2). This possible influence is shown in Scheme3.4:



Scheme3.4

However, even when this factor is taken into account, the expected extent of the reaction should be 185%, which is much much higher than 30%. There must be some other reasons for the low values actually obtained.

Fig3.13 shows the effect of solvent extraction. The chemically modified PVC and PVC plasticised by DBM (the amount of plasticiser

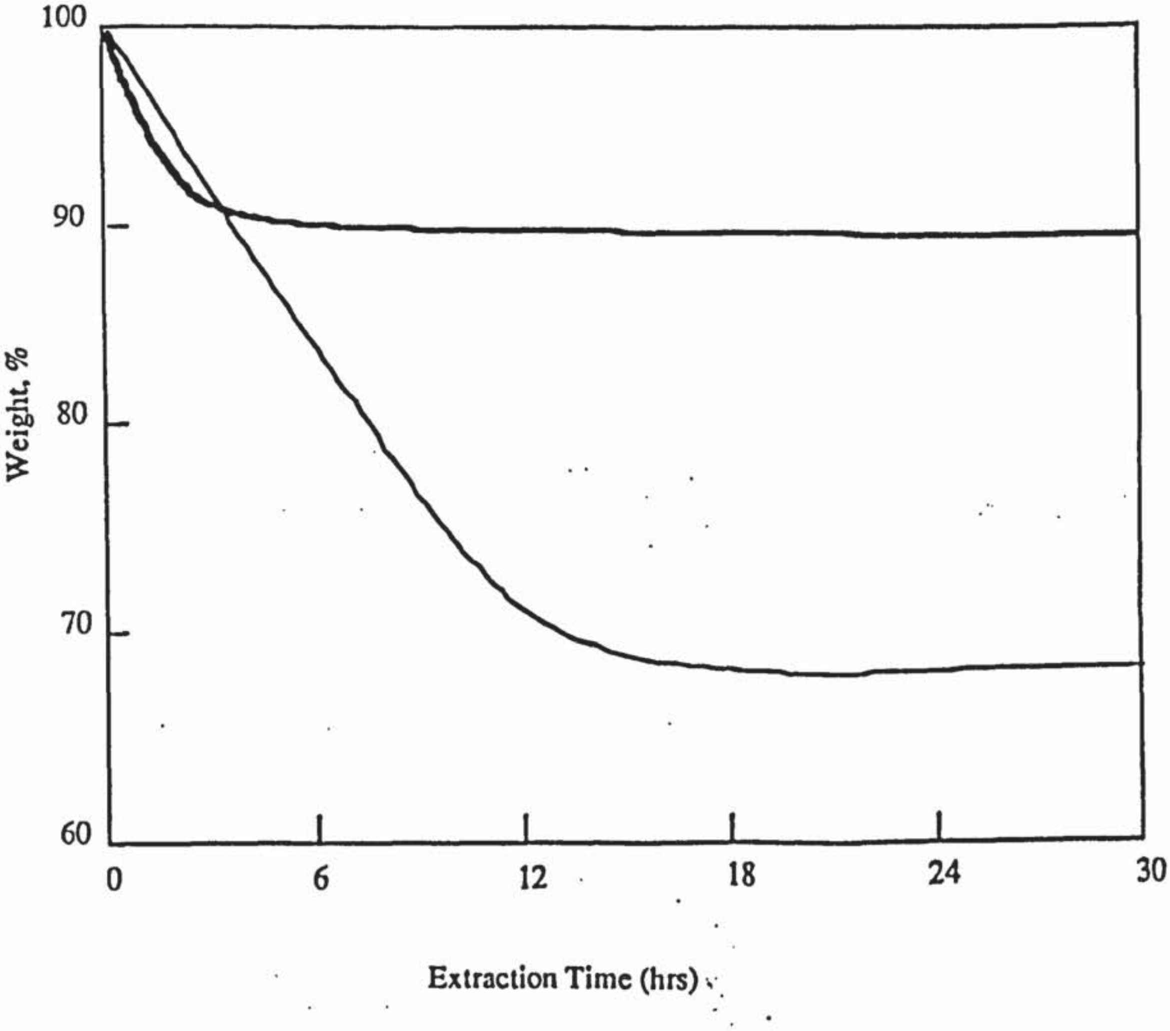


Fig. 3.13 Extraction results of modified PVC and plasticised PVC with the same initial content of plasticisers (30% of PVC by weight)

MPVC

Plasticised PVC

corresponds to the amount of DBM in modified PVC) were extracted by hexane for various times. The initial amount of plasticiser in both samples was 30.0% of PVC by weight. It can be seen that, the plasticised sample lost almost all plasticiser after 20 hours extraction, which is in accordance with other related research[128], wherase the sample of modified PVC still retained a substantial amount of plasticiser in the sample. This is because the plasticizer in the plasticized sample is only a physical solution in the polymer and when it is subjected to exhaustive extraction, the plasticizer is readily removed. In fact, even without extraction, under normal condition, the plasticised PVC may lose some of the plasticisers as a result of diffusion. However, the plasticizers (or more precisely the plasticizing functional groups) in the modified PVC are chemically bound to the polymer chain, making it impossible to remove the plasticizer by extraction. Unless, the chemical bond is broken, or the PVC is depolymerised, the plasticizers will remain in the polymer. Actually, the modified PVC is a polymeric plasticizer on its own right. The difference between plasticised PVC and plasticiser modified PVC with regard to the action of extraction is shown diagramatically in Fig3.14.

Fig3.15 shows the difference in the mechanical test of modified PVC and raw PVC. It can be seen that the modified PVC shows the characteristics of plasticised PVC[129], which reflects the fact that modified PVC does contain plasticisers.

Fig3.16 is the result of creep test which is often used to measure

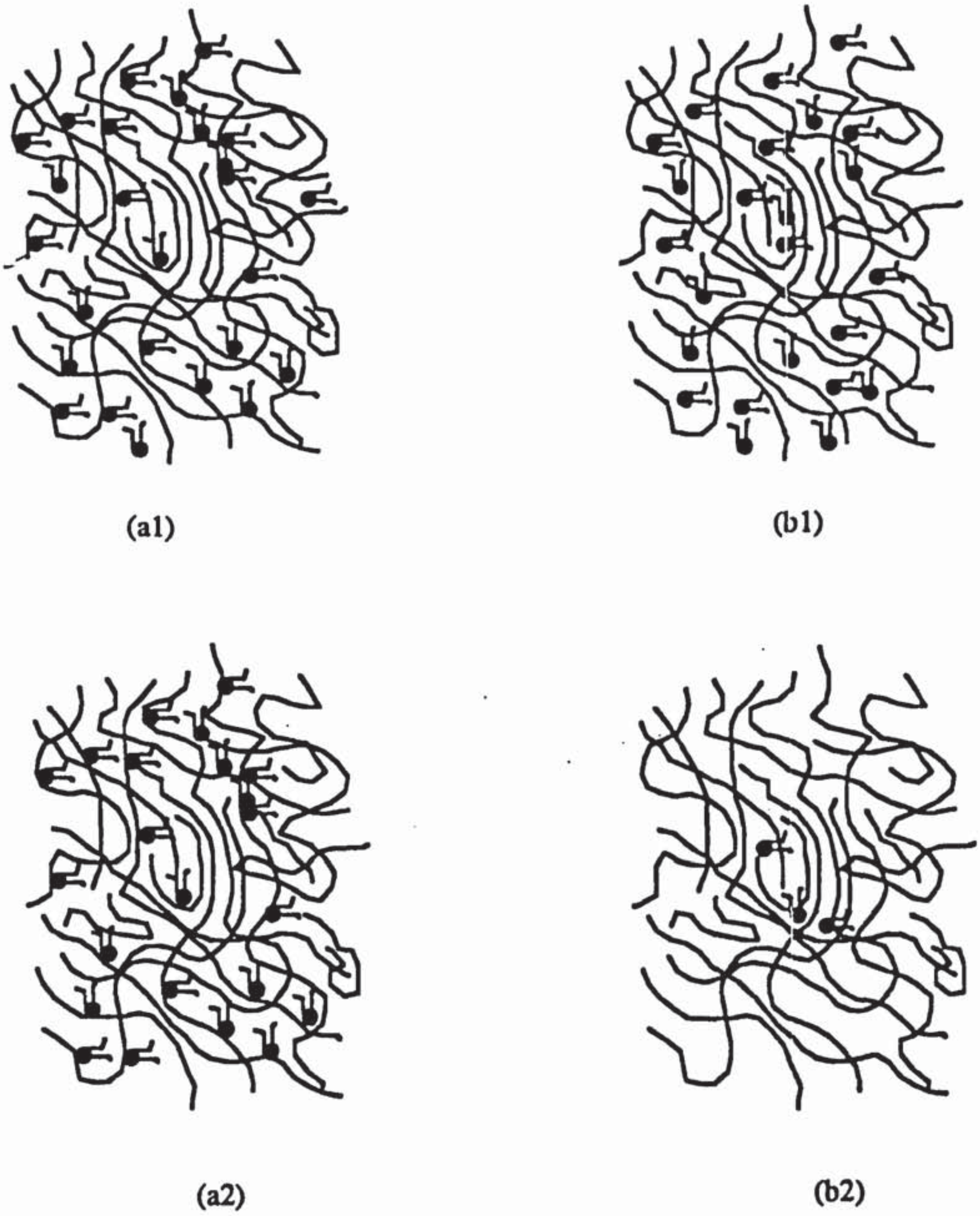


Fig.3.14 Schematic representation of MPVC and plasticised PVC before and after extraction.
(a1) modified PVC before extraction; (a2) after extraction;
(b1) plasticised PVC before extraction; (b2) after extraction

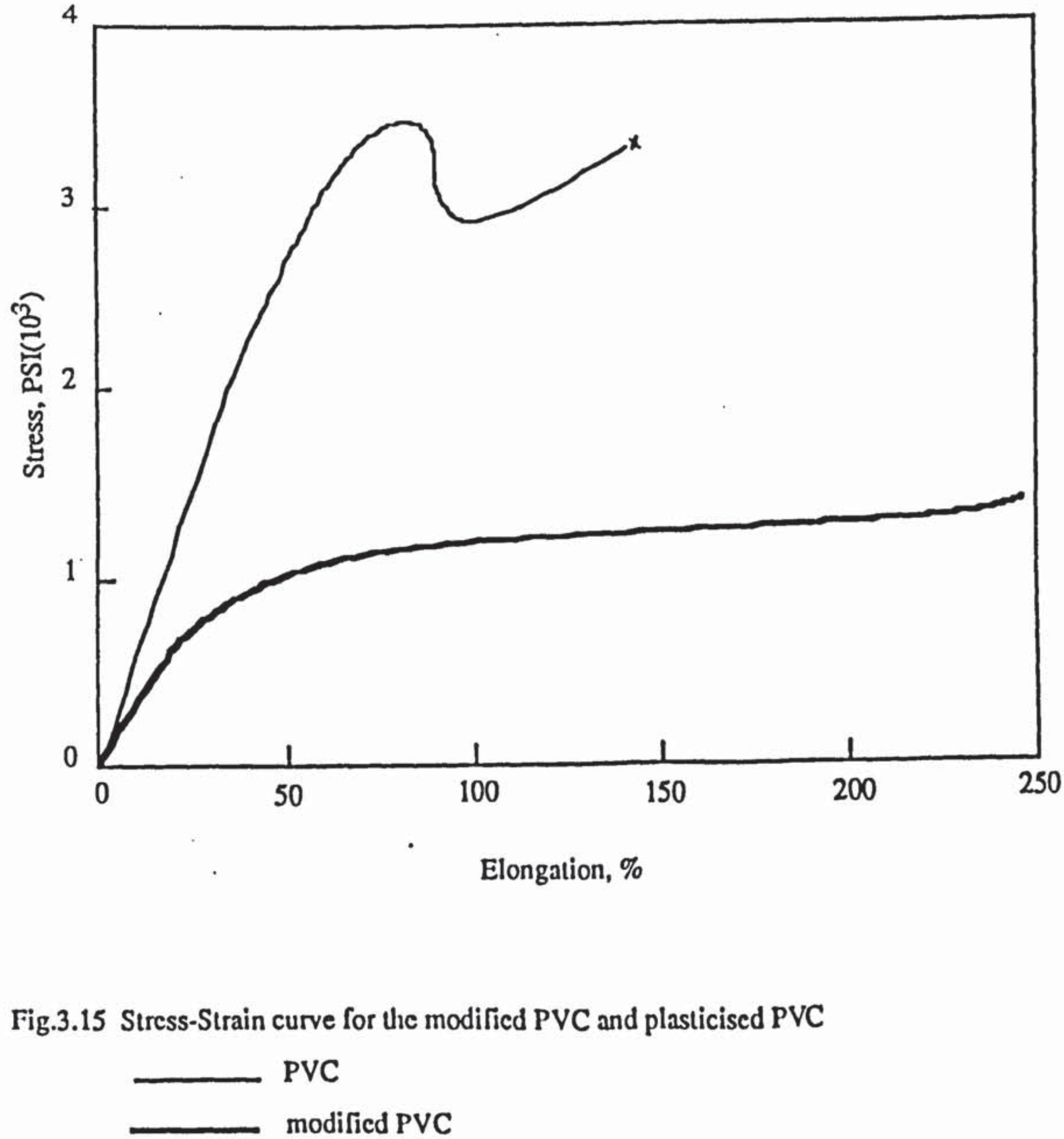


Fig.3.15 Stress-Strain curve for the modified PVC and plasticised PVC

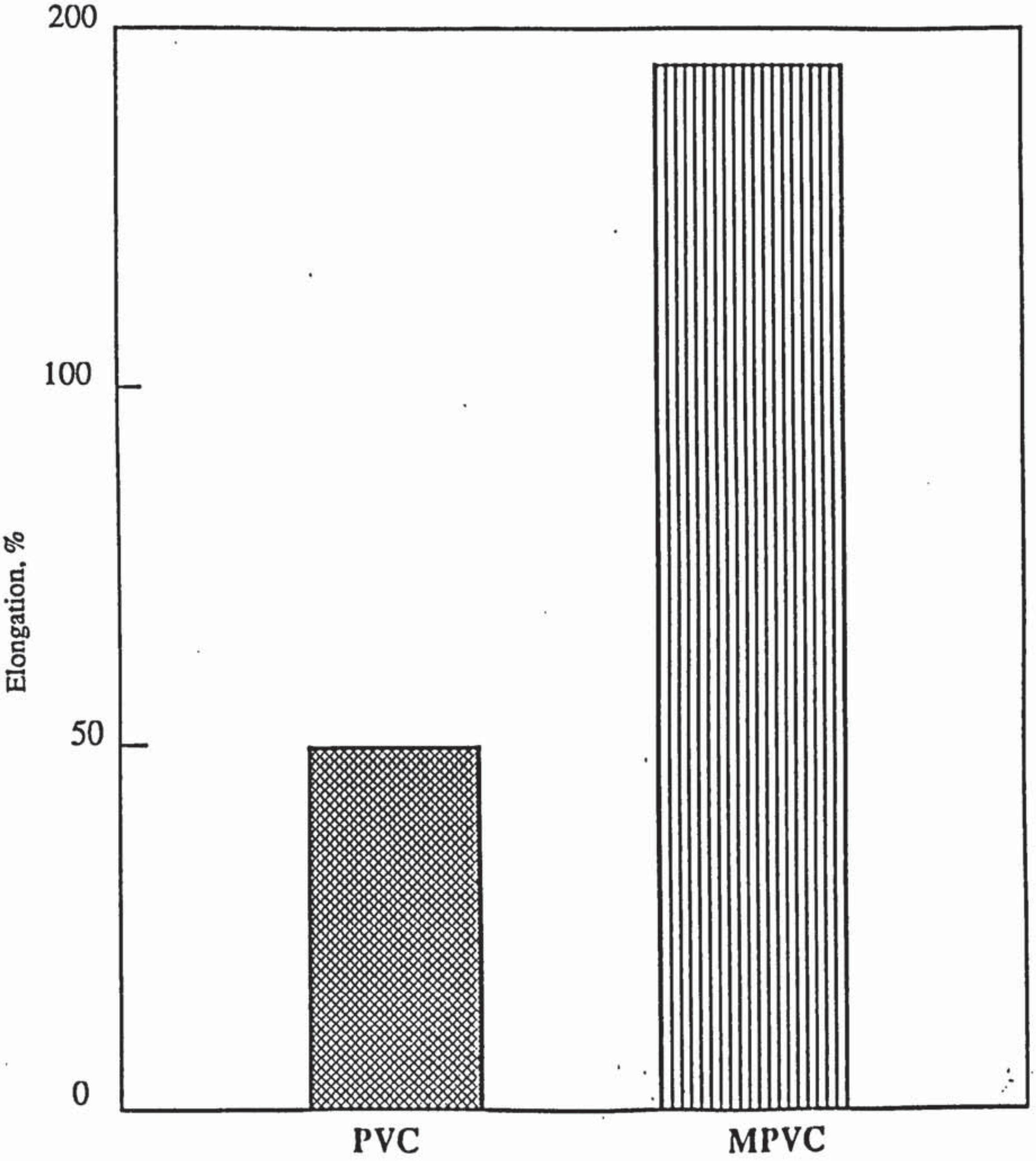
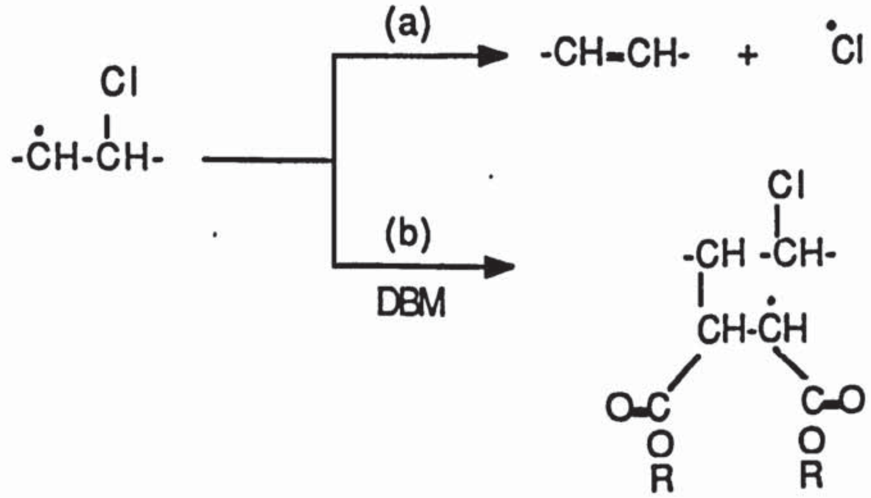


Fig.3.16 Creep test of PVC and modified PVC under 10kg/cm for 48hrs

the plasticity of a material. It can be seen that modified PVC shows a great deal more plasticity, wherease the raw PVC is relatively rigid.

There was no colour formation in plasticiser modified PVC, no HCl could be detected as a product of tile reaction. Even when the reaction was carried out at 90^{0}C for 10 hours, there is still no discolouration. The polymer remained as powder after purification and drying. Fig3.17 is the infrared spectra of PVC and modified PVC. The film made from this modified PVC is completely clean, transparent and soft.

It is known that maleate derivatives are stabilisers[165,166]. The mechanism for the stabilisation has been considered to be the Diels-Alder reaction between the maleate and conjugated double bonds in PVC, which inhibits the further development of polyene. However, the result in this work seems to suggest that there is a possibility that the maleate interacts with PVC through a radical mechanism:



Scheme3.5

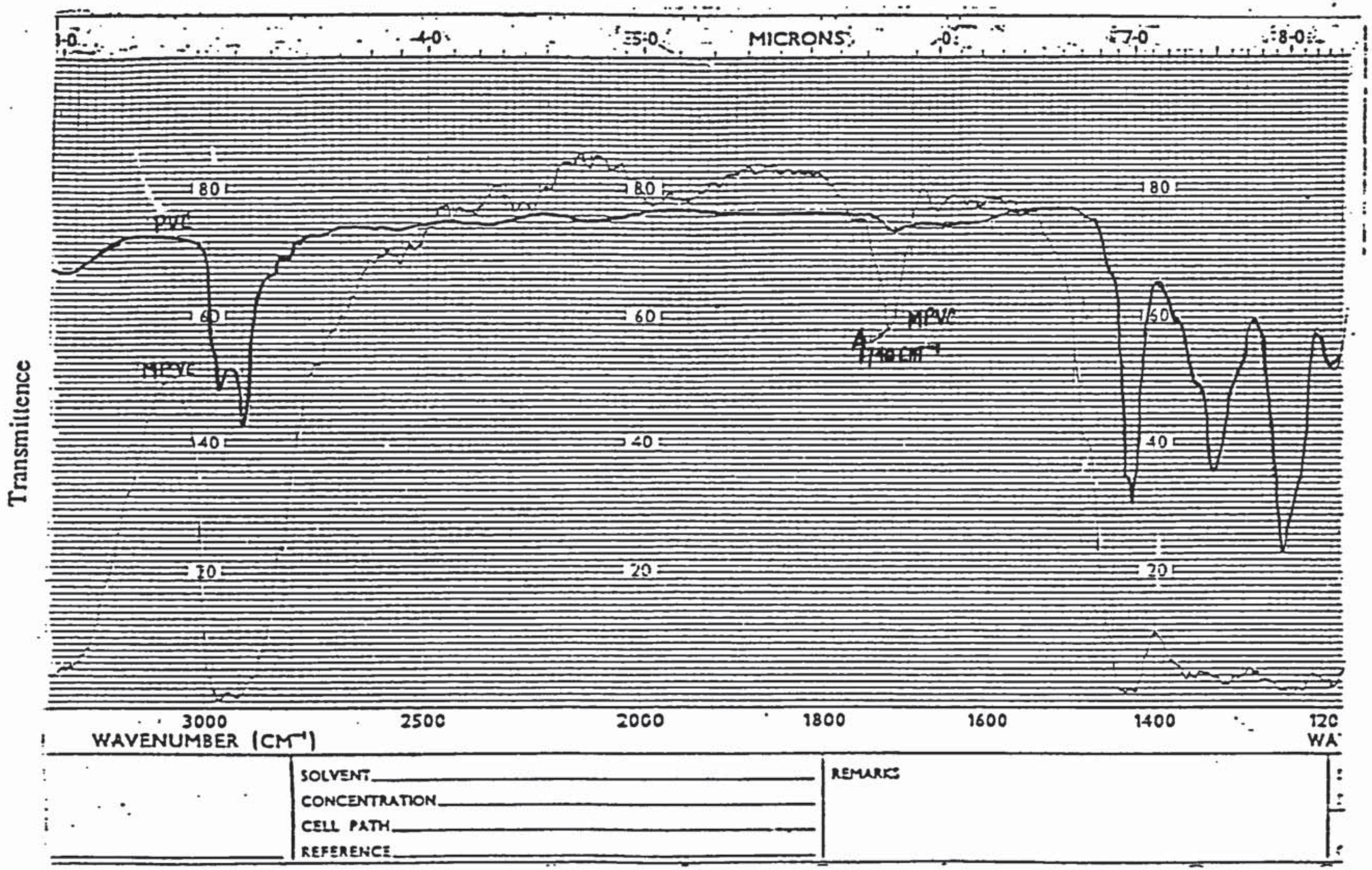


Fig3.17 Infrared spectrum of modified PVC, A: absorption of carbonyl 1740 cm^{-1}.

The competition between process (a) and (b) determines whether PVC degrades to develop unsaturation or reacts with maleic reagents. The results presented here suggest that process (b) is easier than (a), this is the reason why DBM reacted to PVC and no HCl was detected during modification reaction. This is important for understanding the stabilisation mechanism of maleic compounds.

Chapter4
Conclusions and Suggestions for Further Work

4.1 Conclusions

From the work done so far, the following points can be made:

1. Thermal degradation of PVC gives rise to the formation of polyene sequences with different length. The development of unsaturation is random, and the relative amount of polyenes of different length is constant.

2. Through the Diels-Alder reaction mechanism, it is possible to introduce maleic derivatives into degraded PVC, the discolouration of degraded PVC can be reduced substantially as a result of the removal of conjugated unsaturation.

3. By using a radical initiator, is is possible to introduce maleic derivatives (typically dibutyl maleate) along the PVC chain. PVC modified by DBM has the properties of plasticised PVC, but, because the plasticising functional groups in this modified PVC are chemically bound to the polymer chain, when they are subject to exhaustive extraction, they cannot be removed by solvent extraction. This feature may make this kind of material useful where non-migrating additives are needed (e.g. food packaging).

4. Besides the Diels-Alder reaction mechanism of maleic derivatives in the PVC stabilisation, this work also suggests that the maleate may also act by a radical mechanism.

4.2 Suggestions for Further Work

The present work has concentrated on the possibility of using maleic derivative as reagents to introduce functional groups into the PVC molecule. As the physical loss of additives from polymers has attracted the attention of polymer scientists and technologists, it is logical to extend the techniques used in this work to introduce other functional groups through maleic compound into polymer. Such functional groups may be, for example, UV absorbers, antioxidants, thermal stabilisers, and the combinations.

Another line of work which is of importance is to use some other radical initiators to initiate the reactions between PVC and maleic derivatives. Para-mathane, hydroperoxide, benzyl peroxide and cumyl peroxide should be examined.

Part II

The Study of Oscillation in Nitroxyl- and Nitroxyl Precursor-Polypropylene Reaction Systems

Chapter5

The Study of Oscillation in Nitroxyl- and Nitroxyl Precursor-Polypropylene Reaction System

5.1 Background

During the last decade, oscillations in chemical reactions have undergone a process of being rejected as a fundamental concept to being widely recognised as experimental phenomena by chemists [103,131]. Oscillation phenomena in homogeneous chemical system can be dated back to 1921, but the recognition of their importance has been slow, a situation very common in the development of scientific understanding of nature. As philosopher of science, Thomas Kuhn [132] has pointed out, the development of scientific principles involves the accumulation stage of normal science and the emergence of new paradigms. This is also true in the case of chemical oscillation. The rejection by chemists of chemical oscillation was largely due to the fact that at early times chemists tended to interpret the nonequilibrium phenomena in terms of equilibrium thermodynamics, which inevitably would mean that chemical oscillation is contrathermodynamic. Therefore, indisputable experimental evidence and a theory to understand it were urgently needed.

It is natural to think that this kind of theory would be about nonequilibrium situation and because of the work by Onsager and Prigogine, the theory of nonequilibrium thermodynamics was established[133~135]. This theory says that, although chemical oscillation about equilibrium is indeed unlikely, oscillation far from equilibrium is perfectly within the scientific paradigm and inherently consistent with physical laws.

As theoretical understanding of oscillatory reactions were developed, further experimental work continued to emerge. The best-known chemical oscillation is the Belousov-Zhabotinski(BZ) reaction[136,137,138], in which under appropriate conditions an initially homogeneous layer of BZ reagents unstirred develop an elaborate pattern of concentric rings whose colour and composition differ from that of the bulk of the medium[139]. This reaction can actually oscillate both in time and space. The detailed study of this reaction system has stimulated both experimental and theoretical study into chemical oscillations.

Our view of nature changes with time. Chemical oscillation is no longer strange to most chemists[140], and in fact many investigations have been made to discover more oscillations, especially chemical oscillations based on organic systems. It is believed[130] that the oscillation phenomena still to be discovered are far more varied than originally anticipated, and the possible applications are far wider than initially perceived.

It was against such a background then we began to investigate some of the findings which were made in the study of polymer stabilisation and antioxidation action. There is a class of antioxidants[141,142] whose action depends on the generation of their complimentary reduced or oxidised forms, and they form pairs of components which are regenerated at the expense of each other during the reaction process. The experimental results suggest that the mechanisms involved in these regeneration processes are oscillatory. As the chemical nature of antioxidants is important to the understanding of their antioxidation action, the same is true of their dynamical behaviours, that is, it is of practical importance to study the kinetics of anioxidants or stabilisers within the polymer during processing or in later applications.

This part of my work has been to collect experimental results from previous researches on the nitroxyl-polypropylene reaction system which show the oscillation pattern, and try to study representative examples mathematically, using available chemical oscillation methods and computational techniques to reach a reasonable explanation of the experimental results.

5.2 Brief Experimental Description[142]

5.2.1 Experimental procedures

The nitroxyl radical ($>NO^{\bullet}$, which will be denoted as $NO^{\bullet}$ for simplicity) as additive was mixed with unstabilised polypropylene (PP) at concentration of 1.16×10^{-5} mol/g, and processed at $180^{0}C$ in a torque rheometer which was used as a reactor simulator. The experimental procedures were carried out in two ways:

(1) The mixer is closed (CM). It has been shown that although the whole reactor is closed, there is evidence that slow diffusion of air into the system does occur during reaction;

(2) The mixer is left open (OM), so as to investigate the reaction system under the condition of full oxidation.

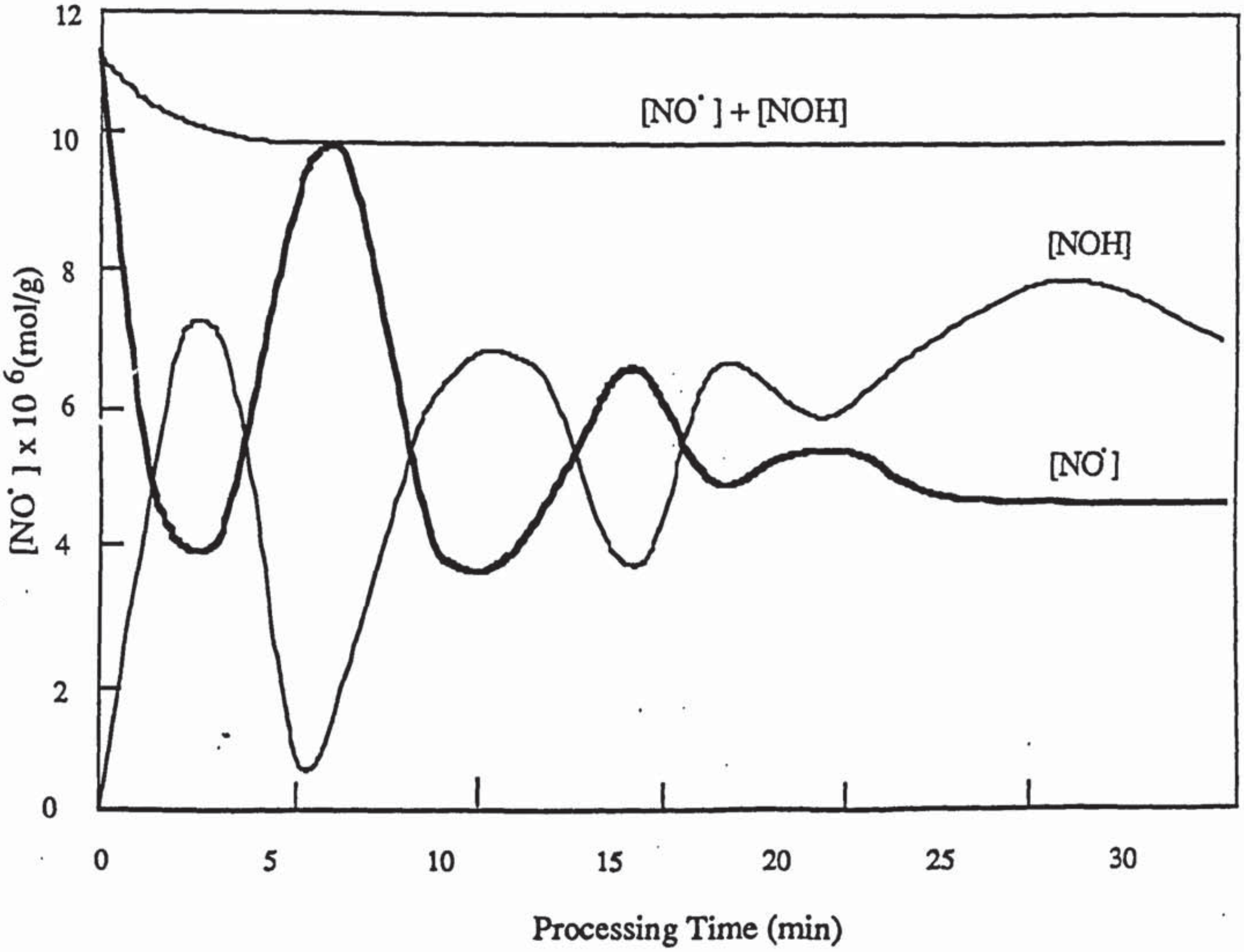
On completion of mixing, the processed polymer samples which now contained fully mixed additive were rapidly removed from mixer and quenched in cold water to prevent further oxidation or other reactions. The polymer was then compression moulded at 180 C for certain time into thin films. Exhaustive extraction of polymer films were carried out in Soxhlet extractor by using dichloromethane as solvent for 12 hours with the protection of nitrogen. The extracts were examined by electron spin resonance spectroscopy (ESR) to measure the concentration of nitroxyl radicals.

Measurement of the concentration of free hydroxylamine (NOH, which will be denoted as NOH) which was generated by the reaction between nitroxyl radical NO and some polymeric species, was achieved by quantitatively converting NOH to the corresponding NO and measuring the total concentration of NO. Because the concentration of NO was already known, the concentration of NOH was obtained by subtracting the independently measured concentration of NO from the total concentration of NO in the sample.

5.2.2 Experimental Observation

It was observed that when the reaction was carried out under the condition of closed mixing, which implies that the presence of oxygen is limited, the concentrations of NO and NOH were found to exhibit an oscillatory pattern (see Fig.5.1). Under condition of open mixing, the oscillation decayed rapidly, and the concentrations of NO and NOH vanished after a certain time (see Fig.5.2).

This phenomenon has practical meaning in polymer stabilisation, that is, in the nitroxyl-polypropylene reaction system, both NO and NOH are actually involved in a regenerative cycle and it is this pair of chemical components that are responsible for the stability of the polymer irrespective of whether the initial additive is NO or NOH. On the other hand, this oscillatory behaviours also requires kinetic attention and, under the extreme complicated condition of polymeric



.Fig5.1 Changes in [NO˙], [NOH] and [NO˙] + [N˙OH] (CM) starting with [NO˙]

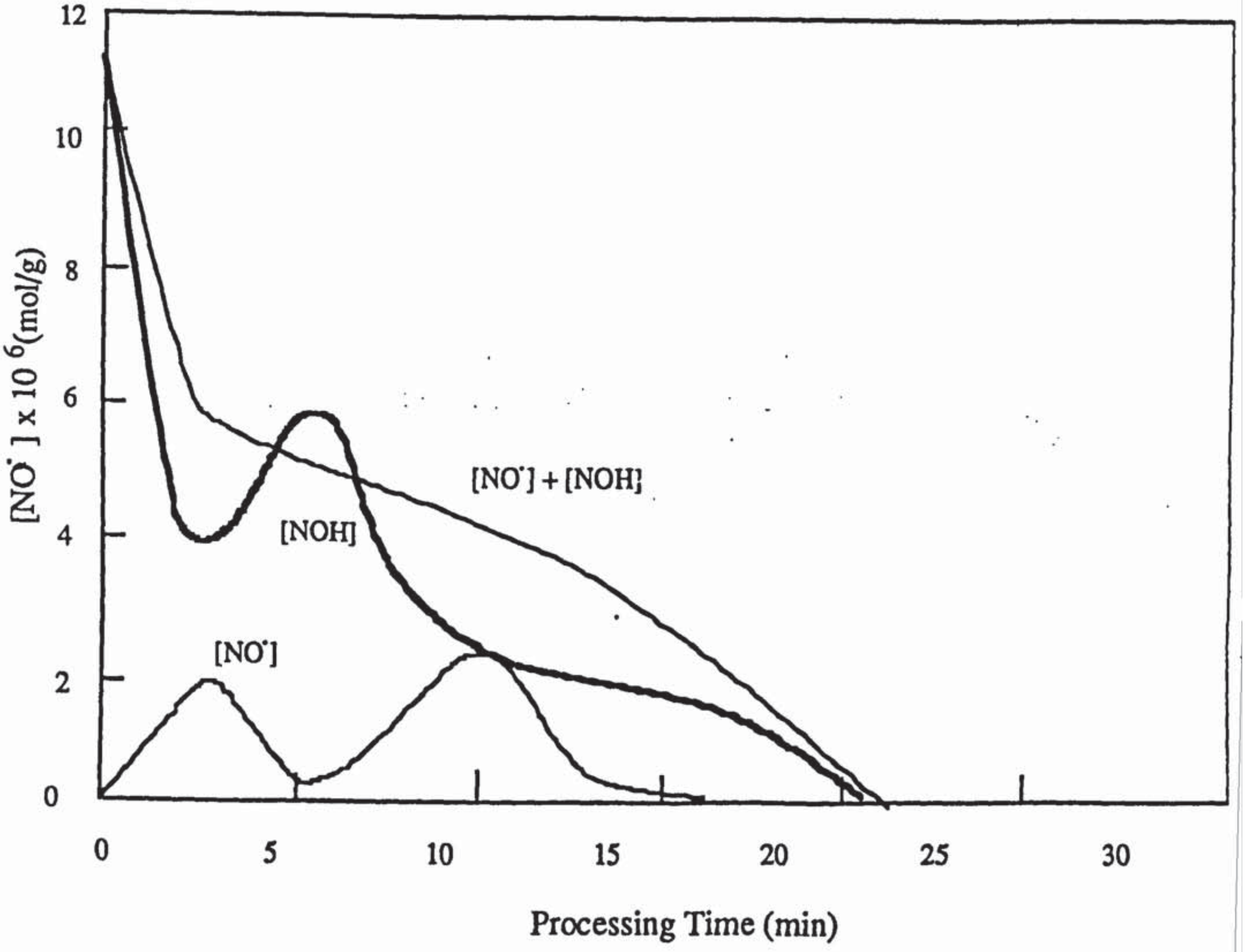


Fig5.2 Changes in [NO˙] , [NOH] and [NO˙] + [NOH] concentrations (OM), starting with [NO˙]

mixing environment, the existence of oscillatory reactions suggests that chemical oscillation is general chemical process which can occur in other ploymer system.

Another important experimental observation is that hydroperoxide POOH was not detected experimentally in the closed mixing. This is important for the later formulation of the reaction mechanism.

5.3 Oscillation Models

In order to form a reasonable model for the oscillation observed in our experimental research, the basic chemical oscillation models will be discussed briefly in this section.

5.3.1 Thermodynamic Aspects of Chemical Oscillation

Among the various states of a system, some can be designated as oscillatory in the sense that repeated maxima and minima in the value of some properties can occur in time and space. These ordered states have been named as dissipative structures[143], because the ordering exists by virtue of the dissipation energy in the irreversible processes.

One of the important thermodynamic parameters of a system is entropy, which is given by

$$dS=deS + diS \qquad (5.1)$$

where deS is the entropy flow due to the exchange of matter and energy between the system and its enviroment, and diS is the entropy production due to the irreversible processes taking place within the system itself. There can be many kinds of irreversible processes, but in the case of chemical oscillations the main concerns are chemical reactions and diffusion of reagents.

The second law of thermodynamics requires that for any process diS > 0. If the system is isolated, deS=0 and equilibrium is the only process possible. In the thermodynamic steady state, dS=diS=0. If the system is closed to exchange of matter with its sorroundings but is maintained at constant temperature and pressure, the Gibbs free energy is the parameter of significance. At equilibrium, the Gibbs energy (G) is at a minimum with respect to any hypothetical change of state and the system will return monotonicallly to equilibrium after any perturbation. So, in a closed system, oscillations about the equilibrium position are impossible. (This was the reason why chemists rejected chemical oscillation in the early studies. see section 5.1).

If the system is open to exchange of matter and entropy with its environment, there may be any number of steady-state processes during which the entropy of the system itself remains constant, i.e., dS=0. For any such steady state:

$$deS=-diS < 0 \qquad (5.2)$$

Because entropy is flowing to the enviroment, the system may maintain a steady state with considerable internal order. Some of these steady states may be oscillatory, but stable undamped oscillations can only occur in open system. This statement is analogous to the requirement that undamped mechanical and electrical oscillators must receive energy from their surroundings.

There are three types of dissipative structures[144] that could develop in chemical systems. These are:

(1) Temporal oscillations of concentration of intermediates about

their steady state values.

(2) Symmetry breaking instabilities such that concentrations of intermediates undergo spontaneous organisation in a previously homogeneous system;

(3) multiple kinetic steady states leading to hysteresis effect.

Until now, only the first two have been observed with certainty in actual chemical systems (Belousov-Zhabotinski reaction and the concentric rings formed by it).

Therefore, irreversible (nonequilibrium) thermodynamics require that sustained chemical oscillations occur only in systems that are open to exchange of matter and energy with their enviroment; they are far from equilibrium so that thermodynamic forces and fluxes are not linearly related, and they obey kinetic laws such that at least one term arises from a reaction which is subject to feedback mechanism. Rapidly damped oscillations may also occur for limited period of time in closed system far enough removed from equilibrium.

5.3.2 Oscillatory Models[145]

There are two kinds of models of chemical oscillation. One is a purly mathematical representation which is derived from a postulated reaction mechanism and displays oscillatory behaviours. A second is based on actual chemical reaction mechanisms whose kinetic constants are used to explain the experimental oscillation.

5.3.2.1 Pure Mathematical Models

5.3.2.1.1 Lotka-Volterra Model[146]

This is the first mathematical model based on an assumed chemical reaction mechanism to give rise the sustained oscillation.

The reaction mechanism of the Lotka-Volterra oscillator is:

$$\begin{aligned} A+X &\xrightarrow{k1} 2X \\ X+Y &\xrightarrow{k2} 2Y \\ Y &\xrightarrow{k3} E \end{aligned} \qquad \text{(R1)}$$

In this model, the concentrations of components A and E remain constant in time, only X and Y change. By applying the law of mass action, the ordinary differential equations (ODEs) describing this system are:

$$\begin{aligned} \dot{X}&=k_1AX - k_2XY \\ \dot{Y}&=k_2XY-k_3Y \end{aligned} \qquad (5.3)$$

The general relation between X and Y is

$$Ye^{(K-Y)}=e^{X}/X \qquad (5.4)$$

where K is a constant determined by the initial conditions. This equation defines the trajectories of X and Y in their phase plane. The equation actually denotes a family of closed curves, each corresponding to a given value of K (see Fig.5.3). In other words, a small perturbation in the initial condition of the system is sufficient to change the behaviour of it to a new oscillatory pattern corresponding to a different frequency. Another interesting feature of this system is that the average concentration of X and Y over an arbitrary cycle are equal to their steady state values. It is very hard to experimentally reproduce the characteristics exhibited by this model. Over the years, the Lotka-Volterra model has been acting as

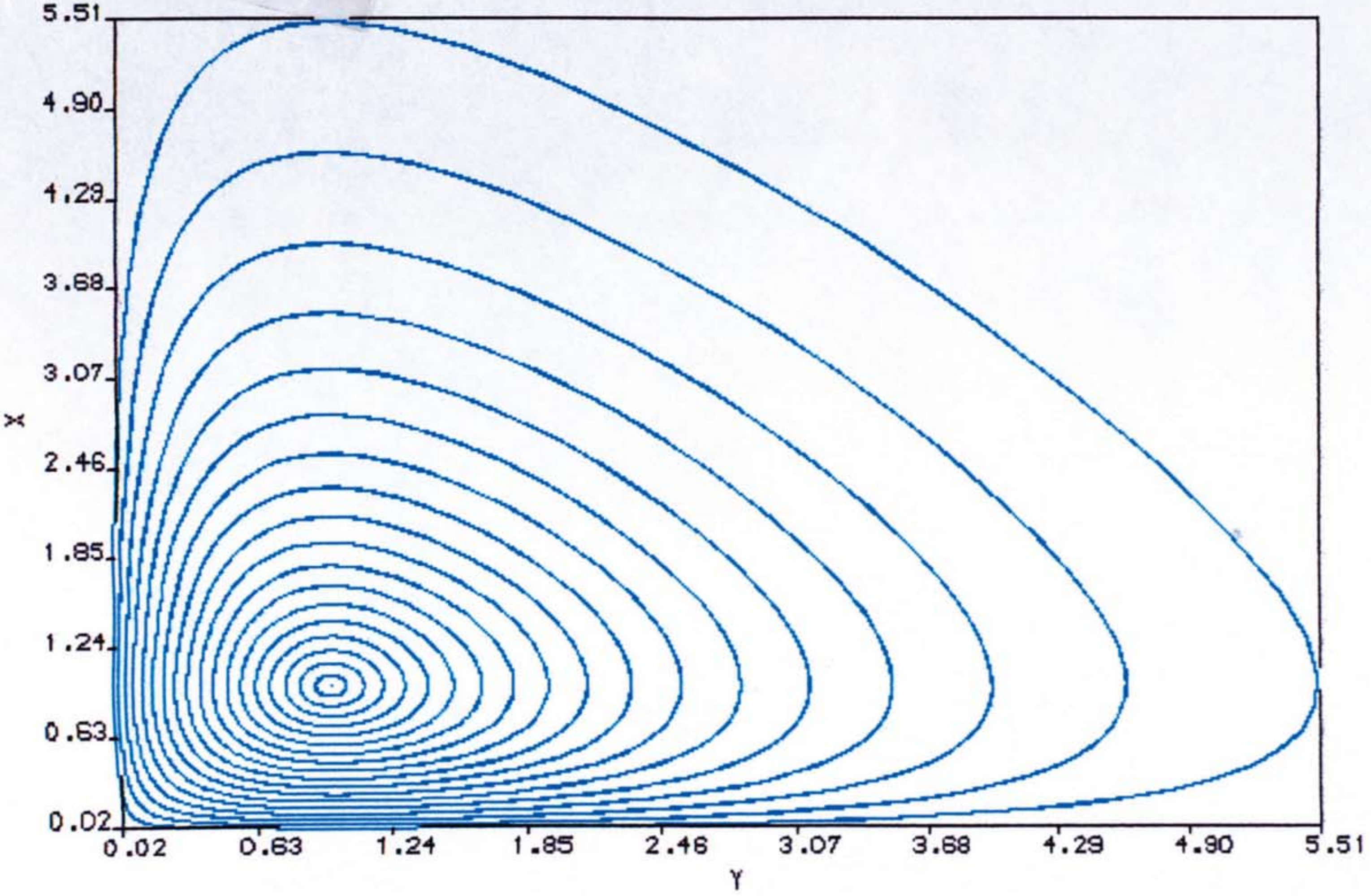


Fig5.3 Phase portrait of Lotka-Volterra oscillator

inspiration for the study of chemical oscillation and nonlinear dynamics[147].

5.3.2.1.2 The Brusselator

The Brusselator is another important model formed to study the chemical oscillation. It was formulated by Prigogine and coworkers[148] on the basis of a chemical reaction mechanism postulated by a computer scientist Turing[149] to explain some striking spatial organisations.

The reaction mechanism for the Brusselator is as follows:

$$\begin{aligned} A &\xrightarrow{k1} X \\ B+X &\xrightarrow{k2} Y+D \\ 2X+Y &\xrightarrow{k3} 3X \\ X &\xrightarrow{k4} E \end{aligned} \qquad (R2)$$

In this system, A and B are the starting chemicals which are maintained constant; D and E are the products also maintained constant, and X and Y are the intermediates. The ODEs for this system are:

$$\begin{aligned} \dot{X} &= k_1A+k_3X^2Y-k_2BX-k_4X \\ \dot{Y} &= k_2BX-k_3X^2Y \end{aligned} \qquad (5.5)$$

Usually, the Brusselator is expressed in the following way:

$$\begin{aligned} \dot{X} &= A-BX+X^2Y-X \\ \dot{Y} &= BX-X^2Y \end{aligned} \qquad (5.6)$$

which is the result of letting all Ks in (5.5) be unity for the purpose of simplicity. It is easy to find that this system has the steady state

$X_0=A, Y_0=B/A$. The normal mode analysis of equation(5.6) shows that when:

$$B > 1+A^2 \qquad (5.7)$$

The steady state becomes unstable and oscillatory solutions might occur(see Fig.5.4). Another important feature of the Brusselator is the limit cycle behaviour which means that under such condition any initial point in the phase plane of X and Y approaches the same periodic trajectory with time (see Fig.5.5).

As chaos is observed in almost every physical science[150,151], many attempts have been made to study the chaotic phenomena observed experimentally. The Brusselator has also been modified by adding a forcing factor into equation(5.6) so that it displays chaos. The forced Brusselator is as follows[152]:

$$\begin{aligned}&\dot{X}=A-BX+X^2Y-X+a\cos(ft)\\&\dot{Y}=BX-X^2Y\end{aligned} \qquad (5.8)$$

Fig.5.6 showes the chaotic trajectory of this system.

5.3.2.2 The Oregonator

The real academic interest into chemical oscillation may be said to begin with the experimental discovery of Belousov and Zhabotinskii[136,137]. The classical BZ reaction consists of the oxidation by bromate ion in an acidic medium of easily brominated organic material, catalyzed by weak one-electron oxidants. There may be many variations of the BZ reaction, depending on the metal-ion catalyst and the organic substrate. However, the requirement for bromate ion in the system seemes to be essential.

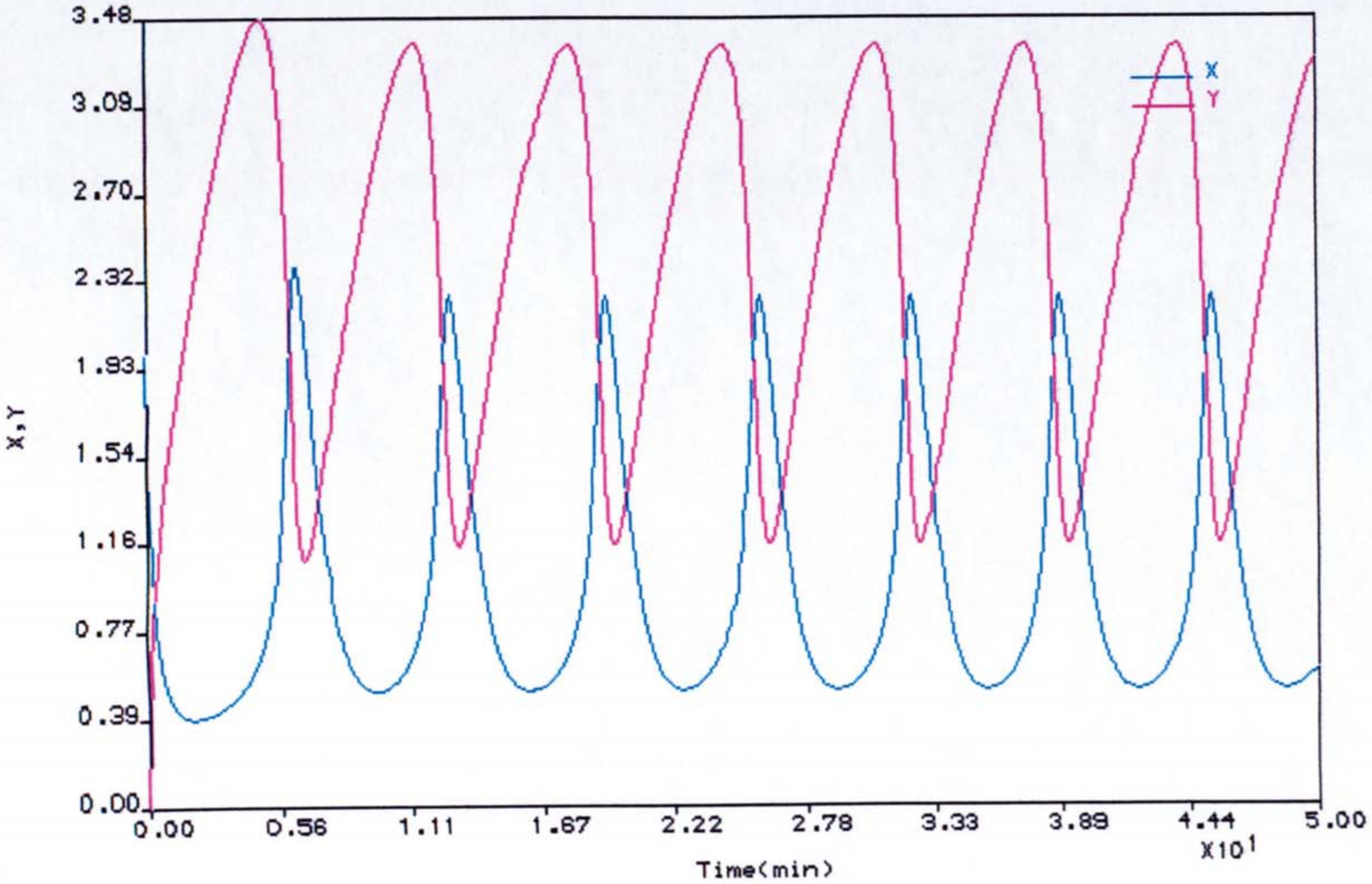


Fig5.4 Brusselator oscillation with A=1.0,B=2.4.

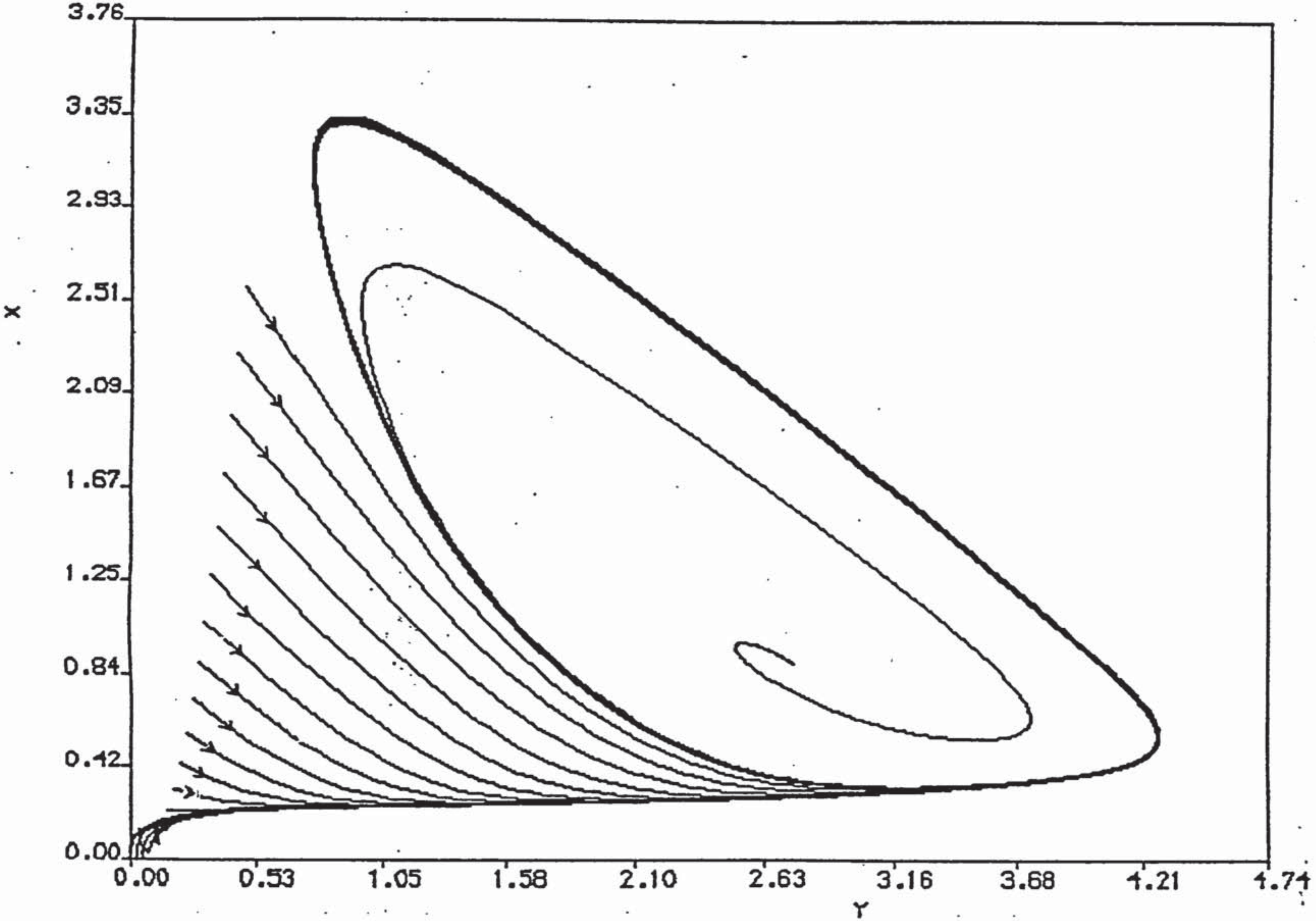


Fig5.5 Limit cycle behaviour of the Brusselator, withA=1.0, B=3.0.

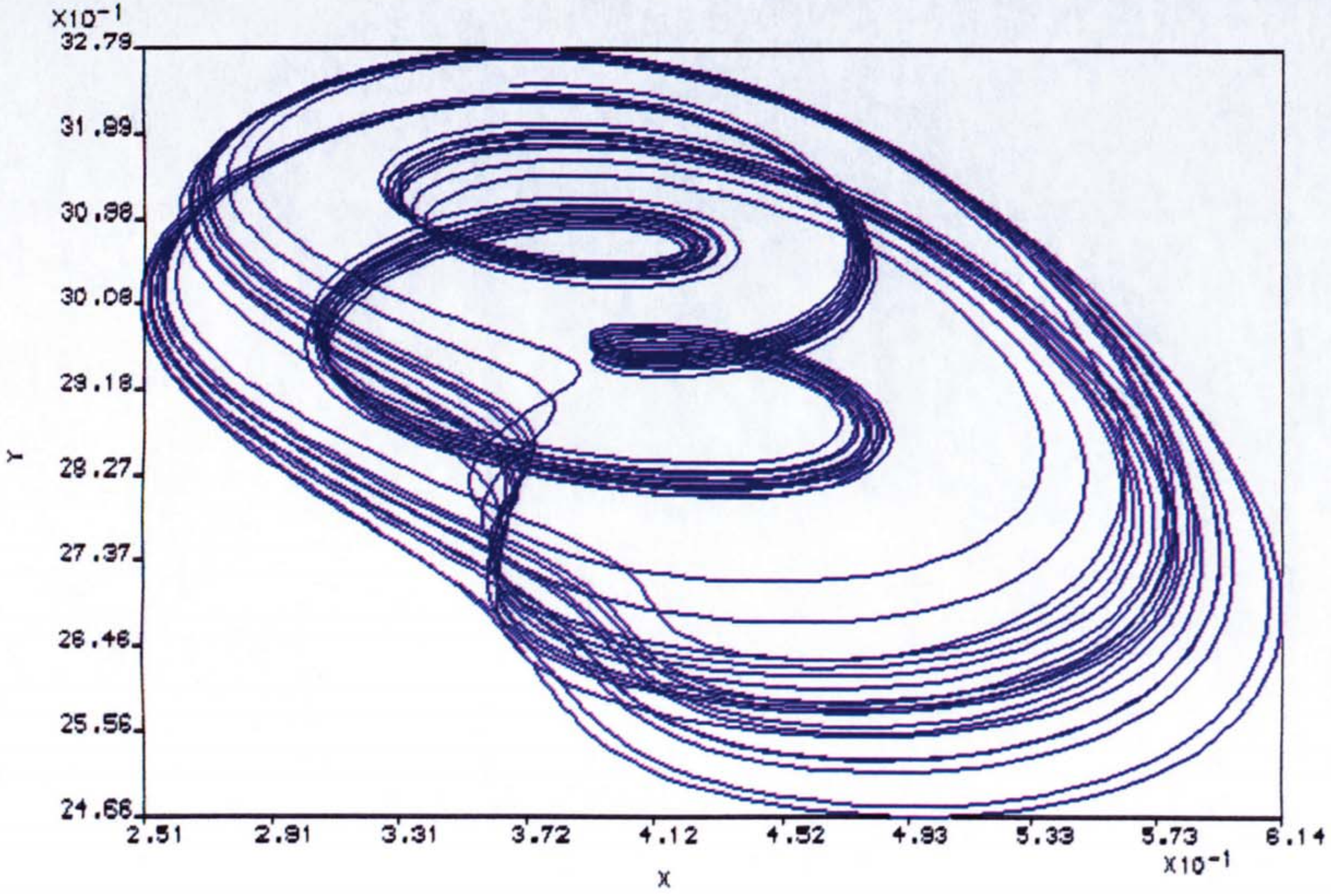


Fig5.6 Chaotic trajectory of Forced Brusselator(A=0.4,B=1.22,a=0.05,F=0.9)

Because the complexity of the system (it is generally believed that there are 18 elementary reactions involved), a model has to be formed to study the oscillatory behaviour observed experimentally. The model derived[153,154] to describe BZ reaction is called the Oregonator which is the only oscillation model which can be derived from the experimental observation and it has been studied in great detail. It contains the following reaction mechanism:

$$\begin{aligned} A+X &\xrightarrow{k1} X \\ X+Y &\xrightarrow{k2} P \\ B+X &\xrightarrow{k3} 2X+Z \\ 2X &\xrightarrow{k4} Q \\ Z &\xrightarrow{k5} fY \end{aligned} \qquad \text{(R3)}$$

where A and B are reactants, P and Q are products, X,Y,Z are the intermediates of interest. k_1 - k_5 are the rate constants for each reaction. f is the stoichiometric coefficient.

The ordinary differential equations to describe this scheme are:

$$\begin{aligned} &\dot{X}=k_1AY-k_2XY+k_3BX-2k_4X2 \\ &\dot{Y}=-k_1AY-k_2XY+fk_5Z \qquad (5.9) \\ &\dot{Z}=k_3BX-k_5Z \end{aligned}$$

The detailed analysis of this system is rather hard mathematically[155,156] and a detailed description is beyond the scope of this work.

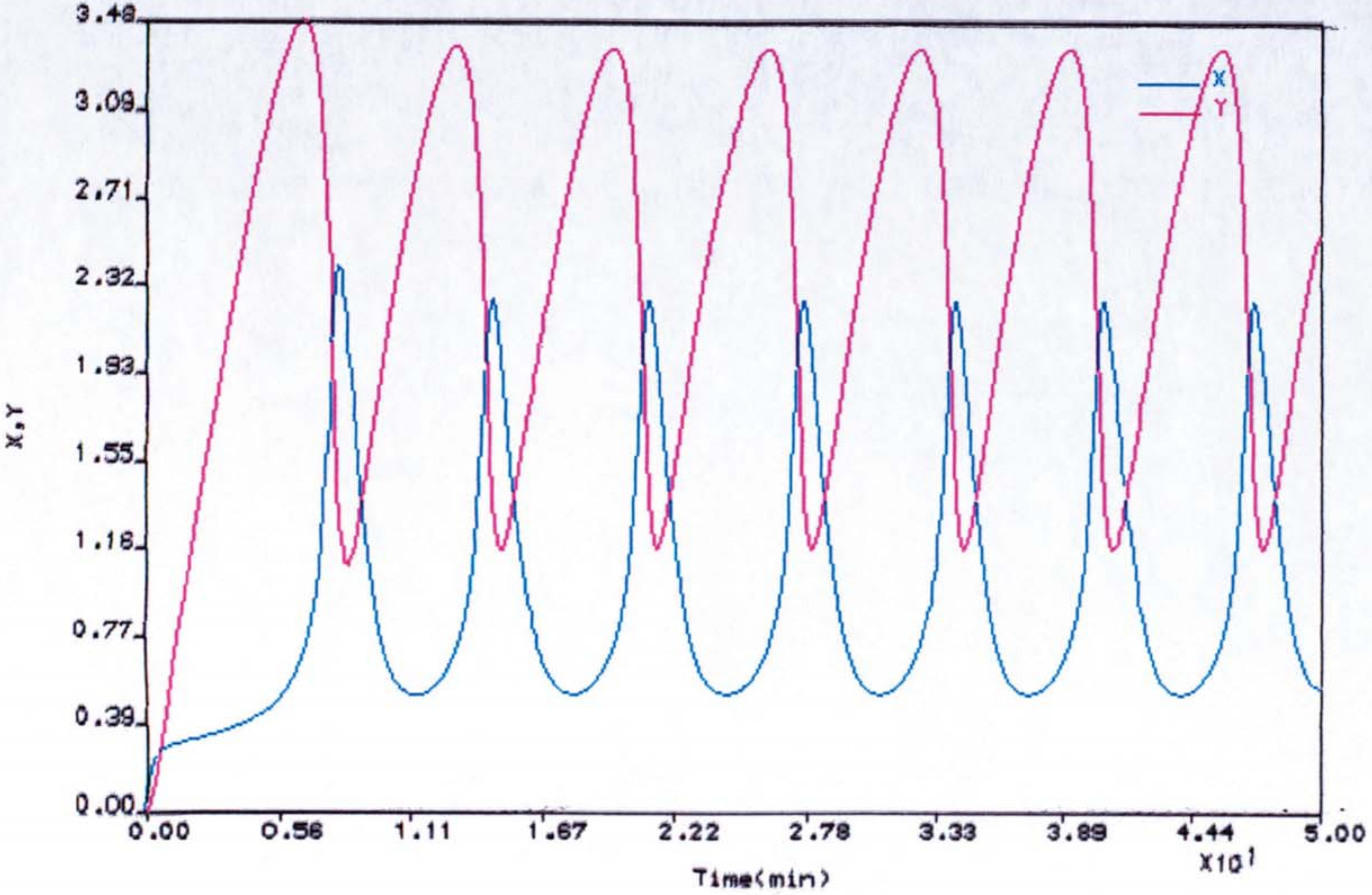


Fig5.7 Brusselator oscillation with A=1.0,B=2.4,x(0)=y(0)=0.0.

5.4 The Study of Oscillation in a Nitroxyl Polymer System

As it has been shown that it is necessary to formulate a reaction mechanism in order to obtain the ODEs which will describe the kinetic behaviour of the reaction system. Although in the system we are going to study there are only two starting reagents, the actual reactions taking place are extremely complex because the characteristics of polymer. It is thought[157] that reactions in (R4) might occur in nitroxyl polypropylene (NO - P) system. However, not all of these reactions to the same extent influence on the kinetic behaviour of the reaction system. Our main interest is to form a model which will exhibit the kinetic behaviour of NO and NOH under two reaction condition (open mixing and closed mixing).

5.4.1 Modified Brusselator

From the models discussed above (see section 5.3), it is natural to think that the Brusselator is the suitable model to start with in the present study. But because the two variables in the Brusselator are the true intermediates, which means that even when their initial concentrations are zero it is still possible to observe oscillation (see Fig.5.7), and the oscillating components in the nitroxyl polypropylene are not all intermediates, at least one of them is a necessary starting material in the experiment(see section 5.2), it is then important to modify the Brusselator for our particular need.

The experimental observation shows that the oscillation is between NO and NOH. Therefore, the proposed model should contain a regeneration step between NO and NOH. It is also known that the crucial step in the Brusselator is the autocatalytic step (see R2) which forms the feedback mechanism causing oscillation. By taking into

$$P \longrightarrow P^{\bullet}$$

$$P^{\bullet} + O_2 \longrightarrow POO^{\bullet}$$

$$POO^{\bullet} + P \longrightarrow POOH + P^{\bullet}$$

$$POOH \longrightarrow PO^{\bullet}, OH, POO^{\bullet}$$

$$P^{\bullet} + P^{\bullet} \longrightarrow \text{molecular product}$$

$$P^{\bullet} + POO^{\bullet} \longrightarrow \text{molecular product}$$

$$POO^{\bullet} + POO^{\bullet} \longrightarrow \text{molecular product}$$

$$P^{\bullet} + NO^{\bullet} \longrightarrow NOP$$

$$POO^{\bullet} + NOP \longrightarrow POOP + NO^{\bullet}$$

$$POO^{\bullet} + NOP \longrightarrow \text{molecular product}$$

$$P^{\bullet} + NO^{\bullet} \longrightarrow NOH + C{=}C \qquad \text{(R4)}$$

$$P + NO^{\bullet} \longrightarrow NOH + P^{\bullet}$$

$$POO^{\bullet} + NOH \longrightarrow NO^{\bullet} + POOH$$

$$POOH + NOH \longrightarrow NO^{\bullet} + PO^{\bullet} + H_2O$$

$$PO^{\bullet} + NOH \longrightarrow NO^{\bullet} + POH$$

$$NO^{\bullet} \xrightarrow[P/O_2]{} \text{molecular product}$$

account of these points and the reactions listed in (R4), we devised the following mechanism for the nitroxyl polypropylene system.

1. Reaction for $NO^{\bullet}$ to produce NOH

$$\mathrm{N\dot{O} + \dot{P}} \xrightarrow{k1} \mathrm{NOH + C{=}C} \qquad (5.10)$$

2. Reaction for NOH to produce $NO^{\bullet}$

$$\mathrm{NOH + PO\dot{O}} \xrightarrow{k2} \mathrm{N\dot{O} + POOH} \qquad (5.11)$$

3. Autocatalytic step between $NO^{\bullet}$ and NOH

$$\mathrm{2N\dot{O} + NOH} \xrightarrow{k3} \mathrm{3N\dot{O} + H^{\bullet}} \qquad (5.12)$$

4.Conversion of $NO^{\bullet}$ into stable molecular products

$$\mathrm{N\dot{O} + B} \xrightarrow{k4} \mathrm{E} \qquad (5.13)$$

The ODEs for this scheme are:

$d[NO]/dt=k_2[POO][NOH]+ k_3[NO]^2[NOH] - k_1[P][NO] - k_4[B][NO]$ (5.14)

$d[NOH]/dt= k_1[P][NO] - k_3[NO]^2[NOH]$ (5.15)

where B denotes a reagent which transformes $NO^{\bullet}$ into a stable product. [POO], [P] and [B] are constant.

By introducing following notations:

$$a=k_1[P], b=k_4[B], c=k_2[POO\ ], X=[NO], Y=[NOH]$$

Then we have

$$\begin{aligned} \dot{X}&=cY + k_3X^2Y-(a+b)X \\ \dot{Y}&=aX-k_3X^2Y \end{aligned} \qquad (5.16)$$

which admits the steady states:

$$X_0=0.0, X_0=0.0 \text{ and}$$

$$X_0=\sqrt{ac/k_3b} \quad Y_0=X_0(b/c)$$

In order to see under what condition the steady states become unstable, we consider infinitesimal perturbation in the form

$$X=X_0 + xe^{\varpi t}$$
$$X=Y_0 + ye^{\varpi t} \qquad (5.17)$$

where

$$\left|\frac{x}{X}\right| \ll 1 , \qquad \left|\frac{y}{Y}\right| \ll 1$$

By introducing equation(5.17) with X_0 and Y_0 being non-zero values into equation(5.16), and neglecting higher-order terms in X and Y, we then obtain the dispersion equation for ϖ in equation(5.17):

$$b(a-b-\varpi)x + (a+b)cy=0$$
$$abx + (ac+b\varpi)y=0 \qquad (5.18)$$

In these algebraic equations, nonzero values of x and y are possible only if

$$\begin{vmatrix} b(a-b-\varpi) & (a+b)c \\ ab & (ac+b\varpi) \end{vmatrix} =0$$

or

$$\varpi^2 + \varpi(b^2+ac-ab)/b + 2ac = 0 \qquad (5.19)$$

This characteristic or secular equation in general has two eigenvalue solutions[158]: ϖ_1 and ϖ_2. If the real parts of ϖ_1 and ϖ_2 are both negative, an infinitesimal perturbation of the steady state will decay to zero. If the real part of either eigenvalue is positive, the perturbation will grow and the system will leave the steady state. The detailed examination of these eigenvalues will disclose whether motion toward or away from the steady state is monotonic or oscillatory.

It is clear that the real part of one of the eigenvalues becomes positive whenever

$$(b^2+ac-ab)/b < 0$$

Or, if the rate constants and other parameters satisfy:

$$k_4[B]-k_1[P]+k_1[P]k_2[POO]/(k_4[B]) < 0 \qquad (5.20)$$

The steady state of the system will be unstable, and the oscillatory phenomenon might appear.

Let us change equation(5.20) into

$$k_1[P](k_2[POO]-k_4[B]) > (k_4[B])^2 \qquad (5.21)$$

and see what it actually means in the reaction model(5.10) to (5.13). From (5.21), it is obvious that

$$k_2[POO\ ] > k_4B$$

in order to maintain an unstable steady state. That is, the rate of the reaction(5.11) must be faster than that of the reaction(5.13) in the model so that the unstable steady state can be maintained to allow the onset of oscillations. From chemical point of view, this is more understandable. Because reaction(5.13) converts NO into inert product, if its rate is very high, the whole dynamic process could not be maintained due to depletion of NO which would soon decay, in other words, if the steady state is stable there is no chance for the appearance of oscillation in such a situation.

Knowing the relation between the parameters of this system, we can carry out some numerical experiment. The results are shown in Fig.5.8 to Fig.5.10. These are the basic patterns we may observe from the system. Fig.5.11 gives the similar oscillatory behaviour to the

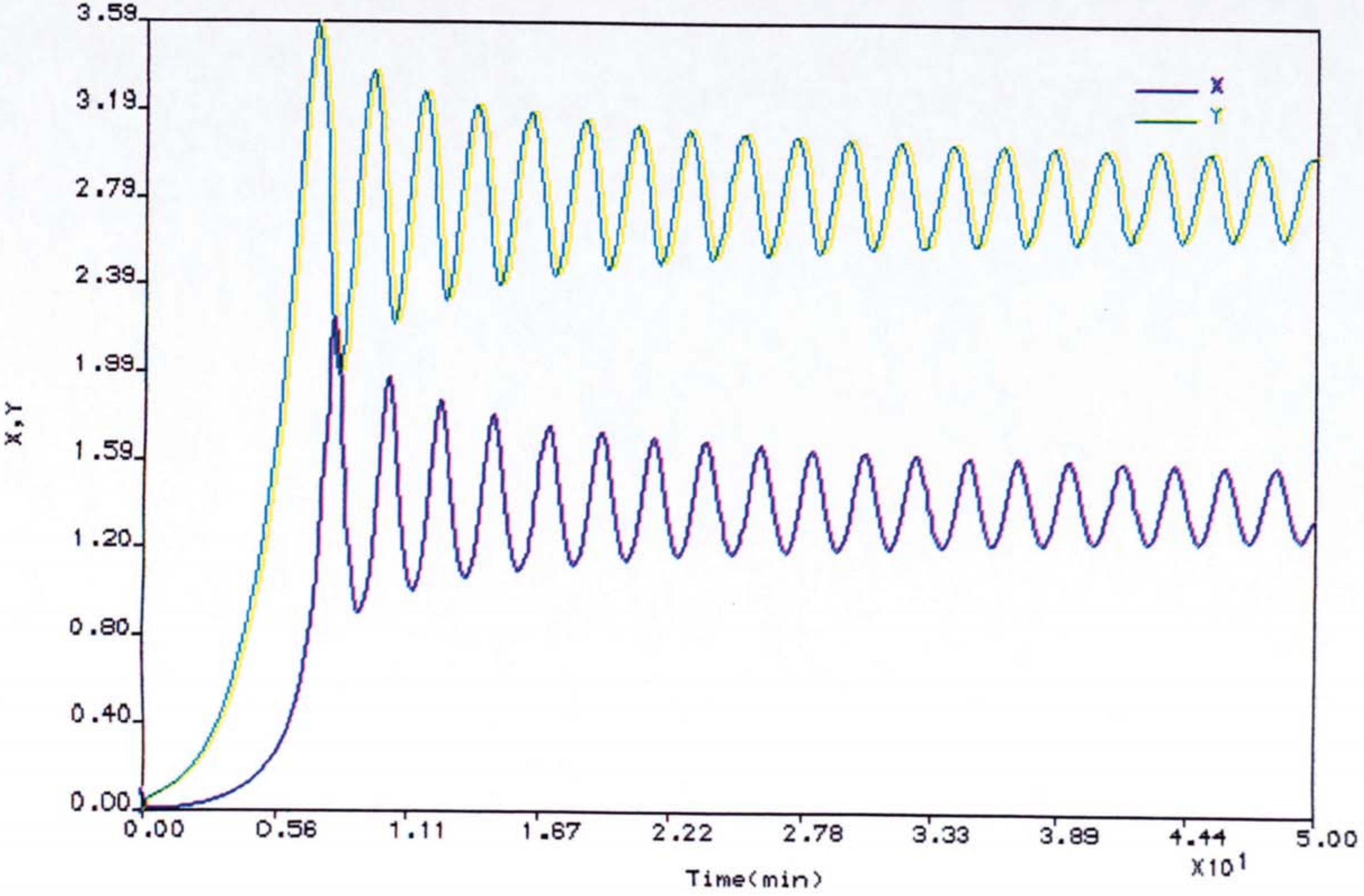


Fig5.8 Oscillaton of modified Brusselator(a=4.0,b=1.0,c=2.0).

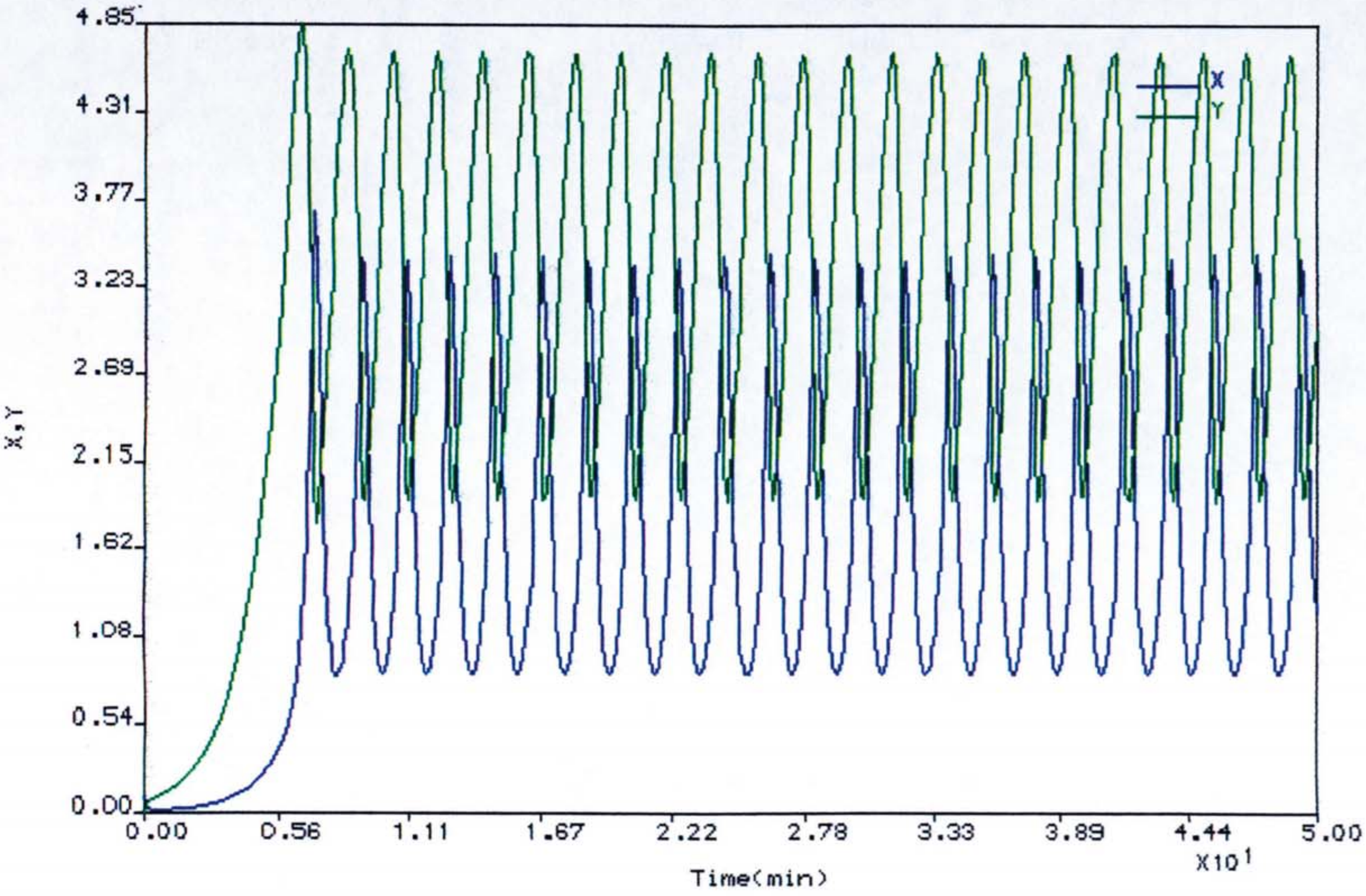


Fig5.9 Oscillation of modified Brusselator(a=6.0,b=1.0,c=2.0).

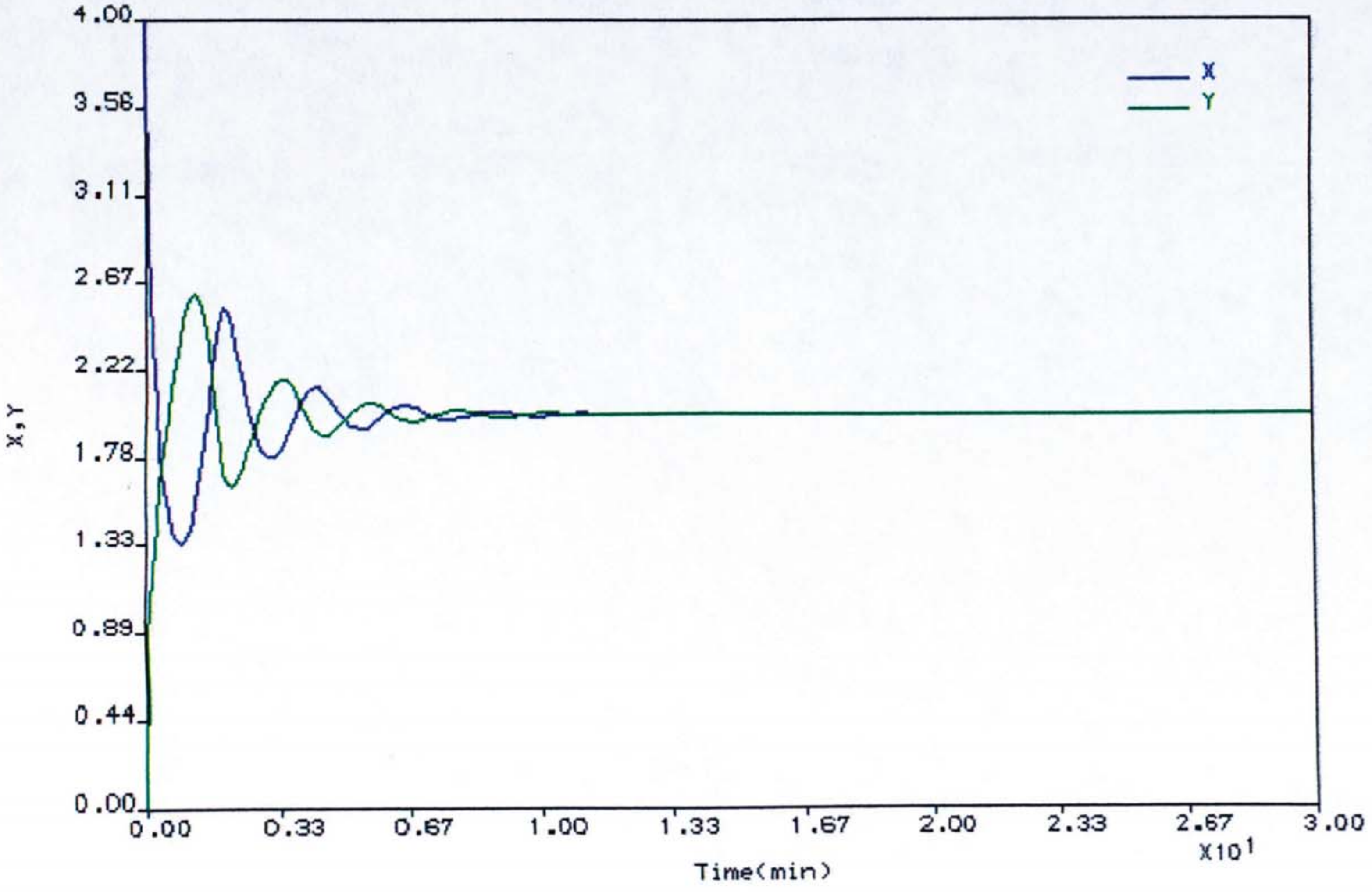


Fig5.10 Oscillation of modified Brusselator(a=4.0,b=1.0,c=1.0).

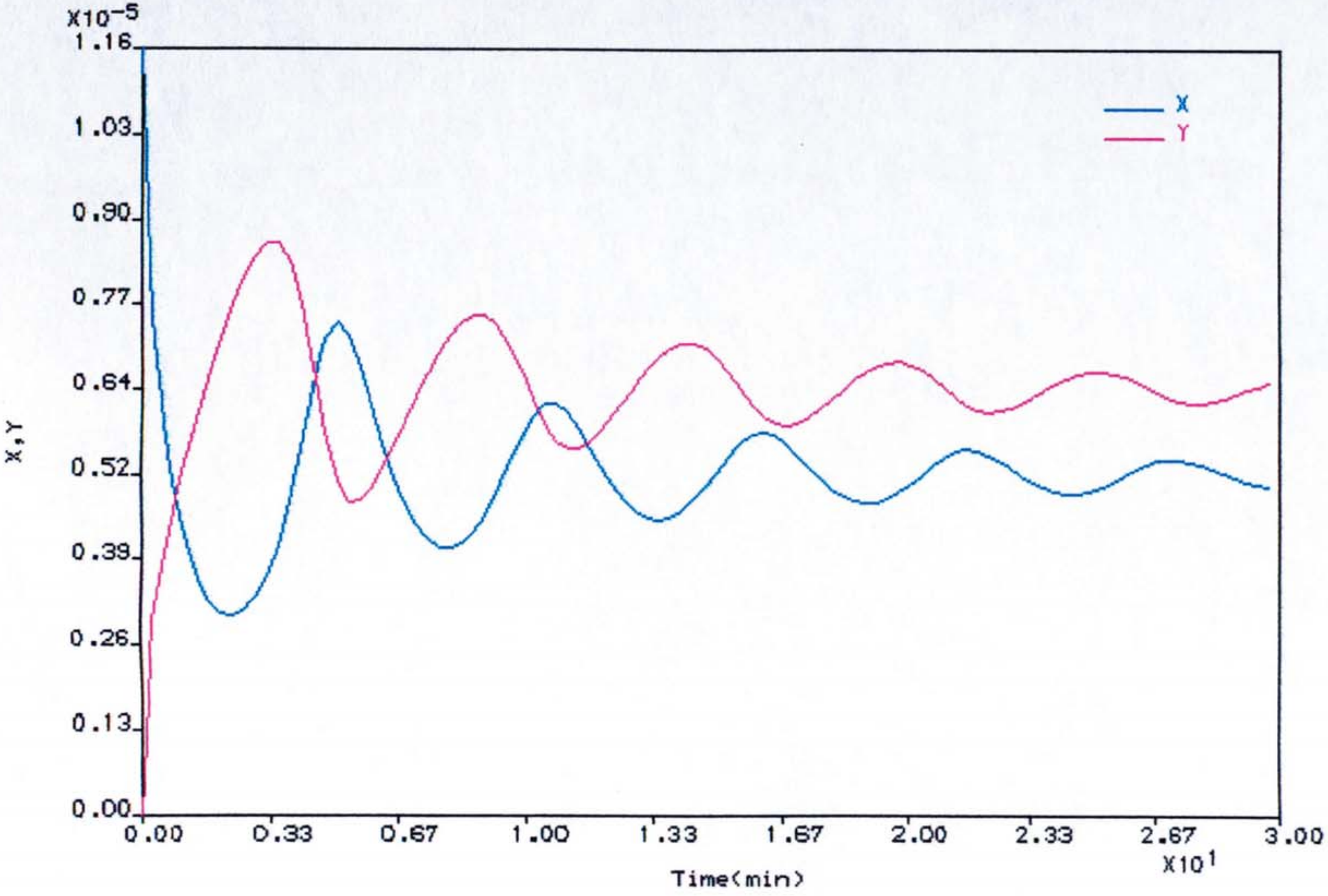


Fig5.11 Simulation result from modified Brusselator(x=[NO],y=[NOH]).

experimental observation(see Fig.5.1). The rate constants used for this are:

$$k_1=4.0 \times 10^5 \text{ litre mol}^{-1} \text{ s}^{-1}$$
$$k_2=5.0 \times 10^5 \text{ litre mol}^{-1} \text{ s}^{-1}$$
$$k_3 = 1.4 \times 10^9$$
$$k_4 = 1.0 \times 10^4 \text{ s}^{-1}$$

k_2 is from experimental value [159]. There are no experimental values for other rate contants. The chemical significance of reaction (5.12) is not clear. Its unusually high rate constant makes it unacceptable as a real reaction mechanism in view of the fact that it produces a hydrogen atom. However, as a modified version of the Brusselator, it has some characteristics worth mentioning. First, it has two steady states. Second, also the most important feature, is that in certain parameter range this model displayes chaotic behaviour(see Fig.5.12 and Fig5.13) which is not exhibited by the Brusselator. In fact, only the forced Brusselator displays this chaotic bahaviour. This feature may make this model useful in the study of chemical chaos and may be used in the study of some other chemical oscillation phenomenon. Like forced Brusselator, it is also possible to introduce sinusoidal forcing function into Eqs(5.16) to obtain the forced form of it:

$$\begin{aligned} &\dot{X}=cY + k_3X^2Y-(a+b)X+\alpha\cos(ft) \\ &\dot{Y}=aX-k_3X^2Y \end{aligned} \qquad (5.16a)$$

This oscillator displays very strong nonlinearity. One feature of it is that it can display the behaviours of forced Brusselator and other known oscillatiors. Some of the phase diagrams from this oscillator are shown in Figs5.14~5.16.

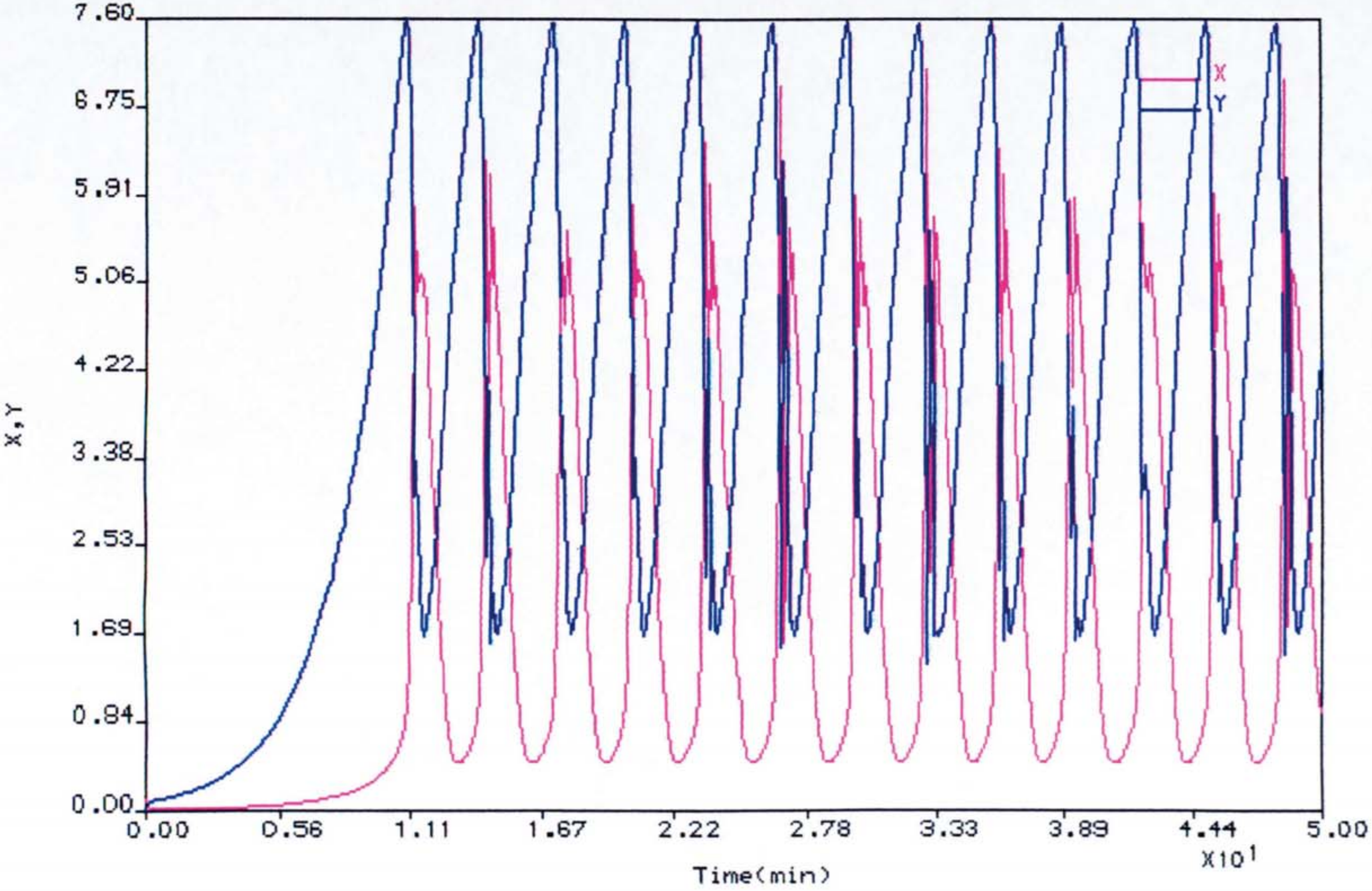


Fig5.12 Chaotic behaviour of modified Brusselator(a=8.0, b=0.5, c=1.0).

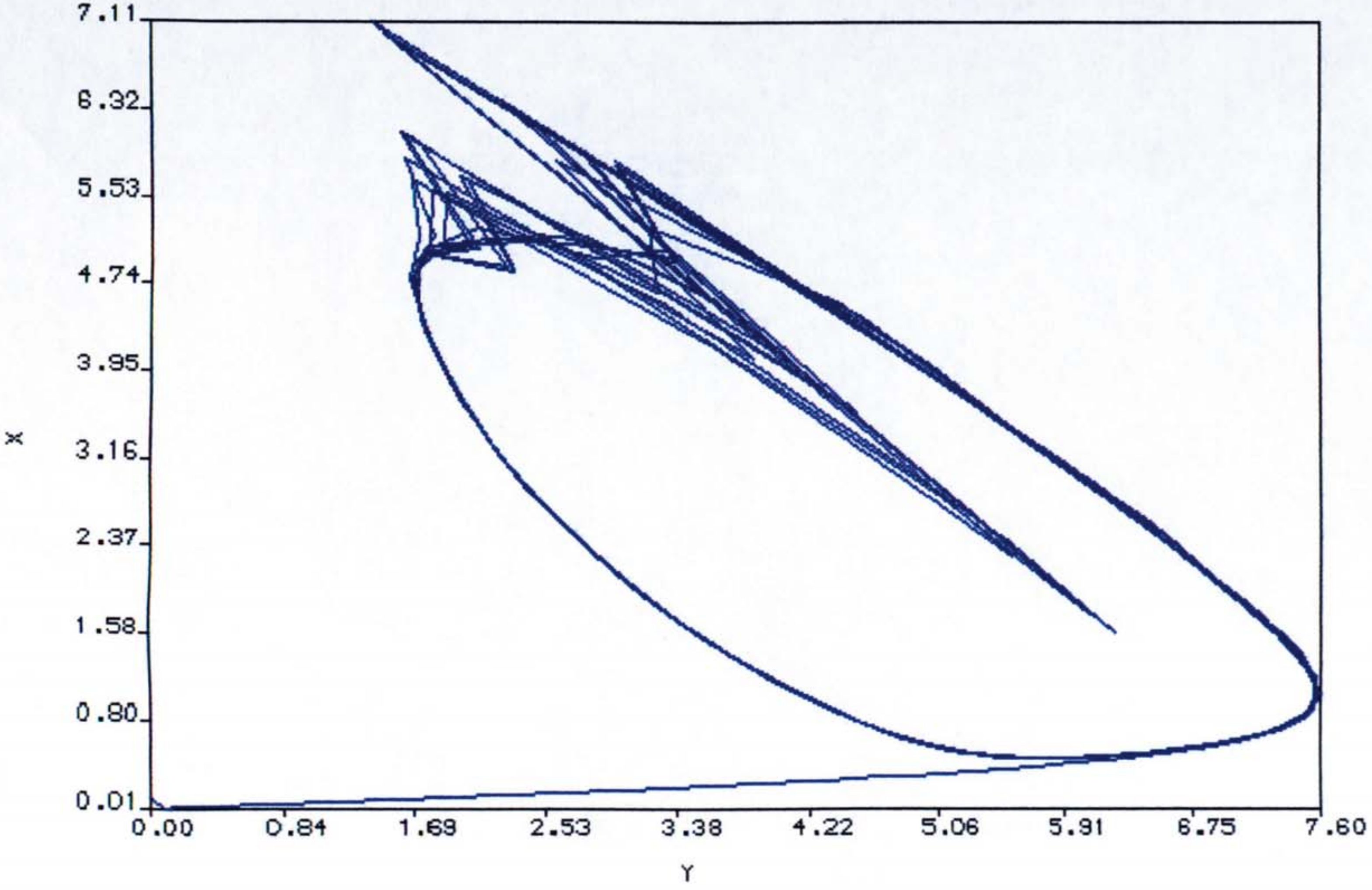


Fig5.13 Phase diagram from Fig5.12.

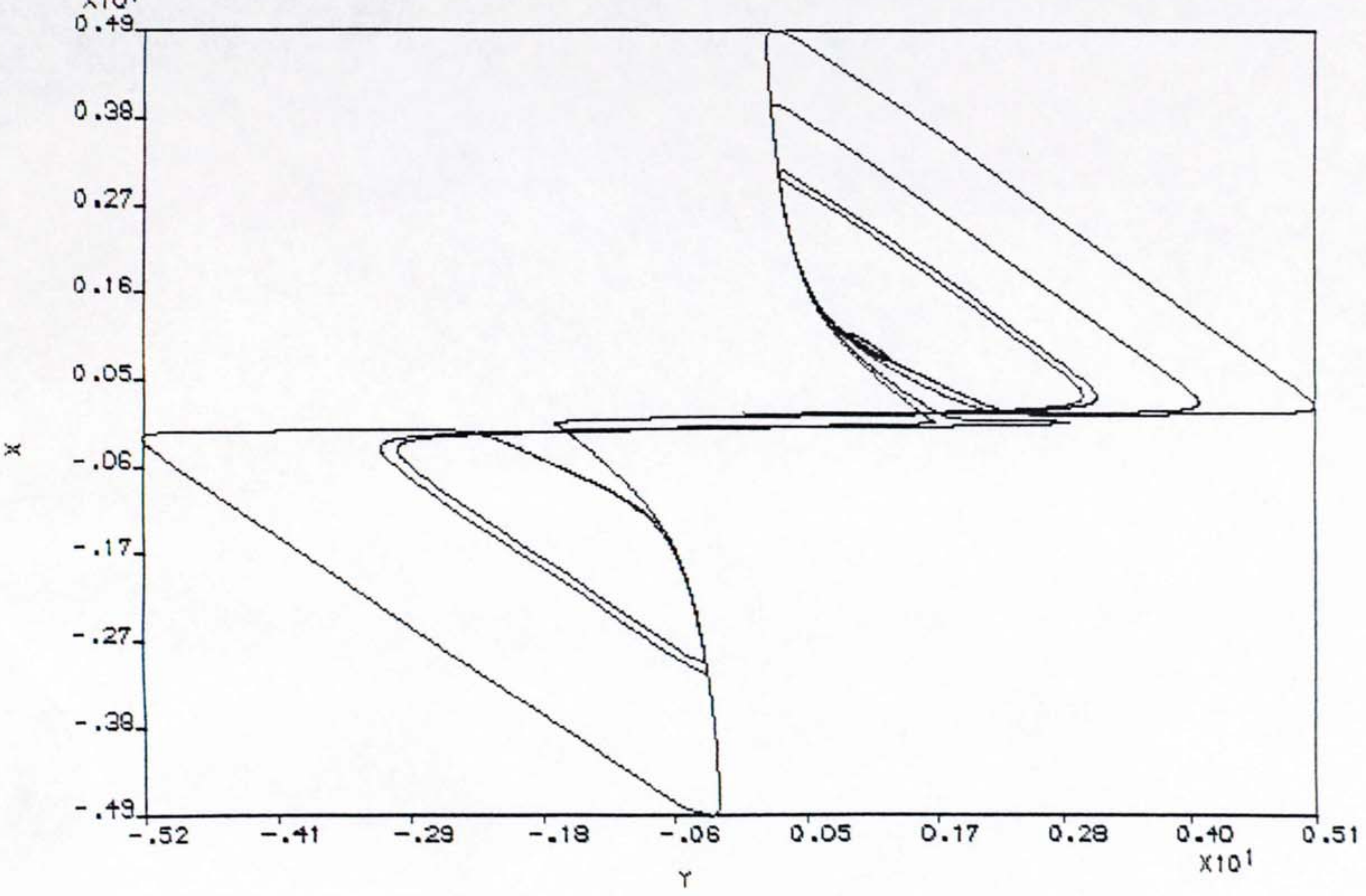


Fig5.14 Complex periodic trajectory with a=1.0,b=0.01,c=0.1.

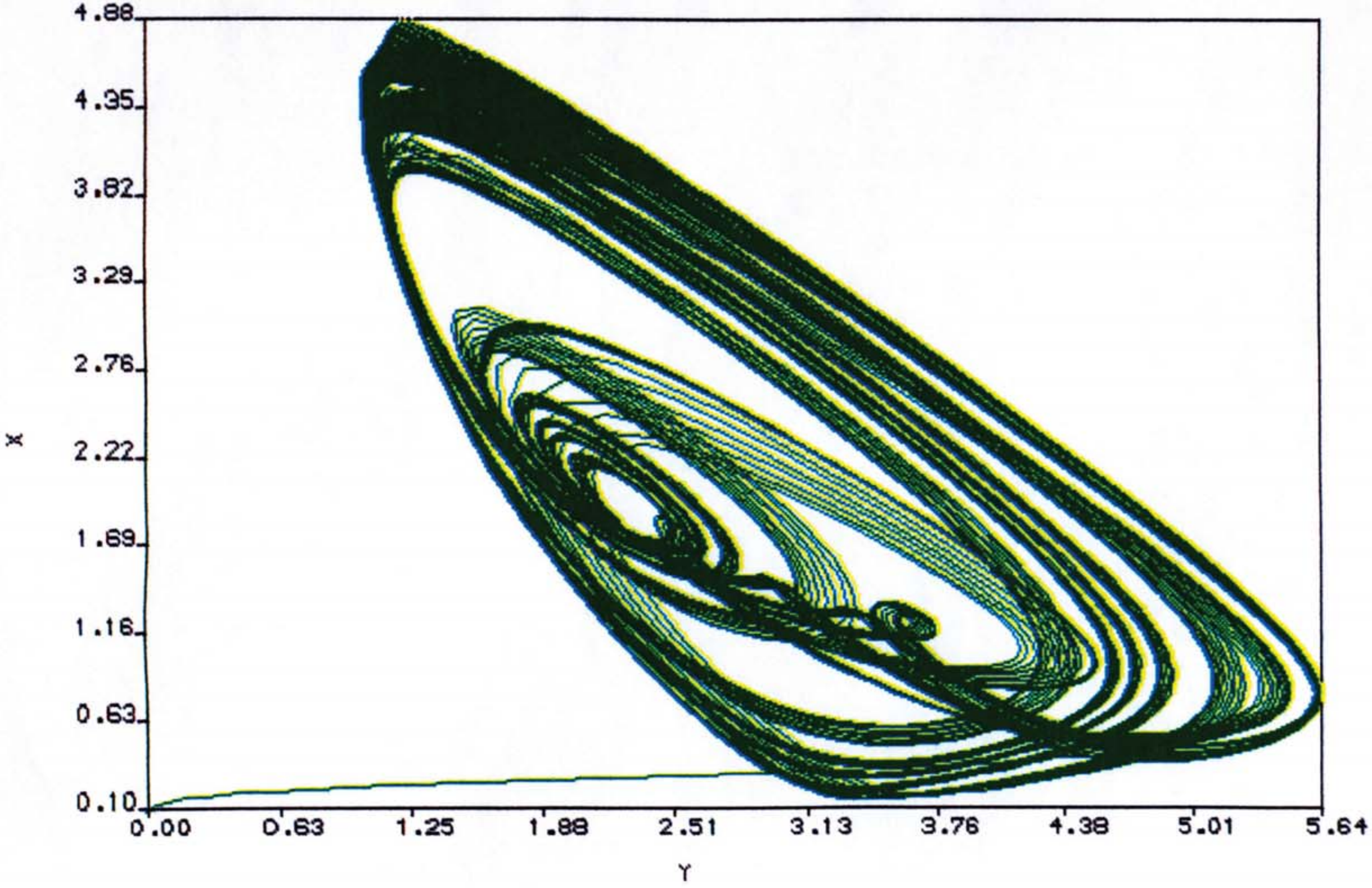


Fig5.15 Chaotic trajectory with a=4.5,b=0.5,c=1.0

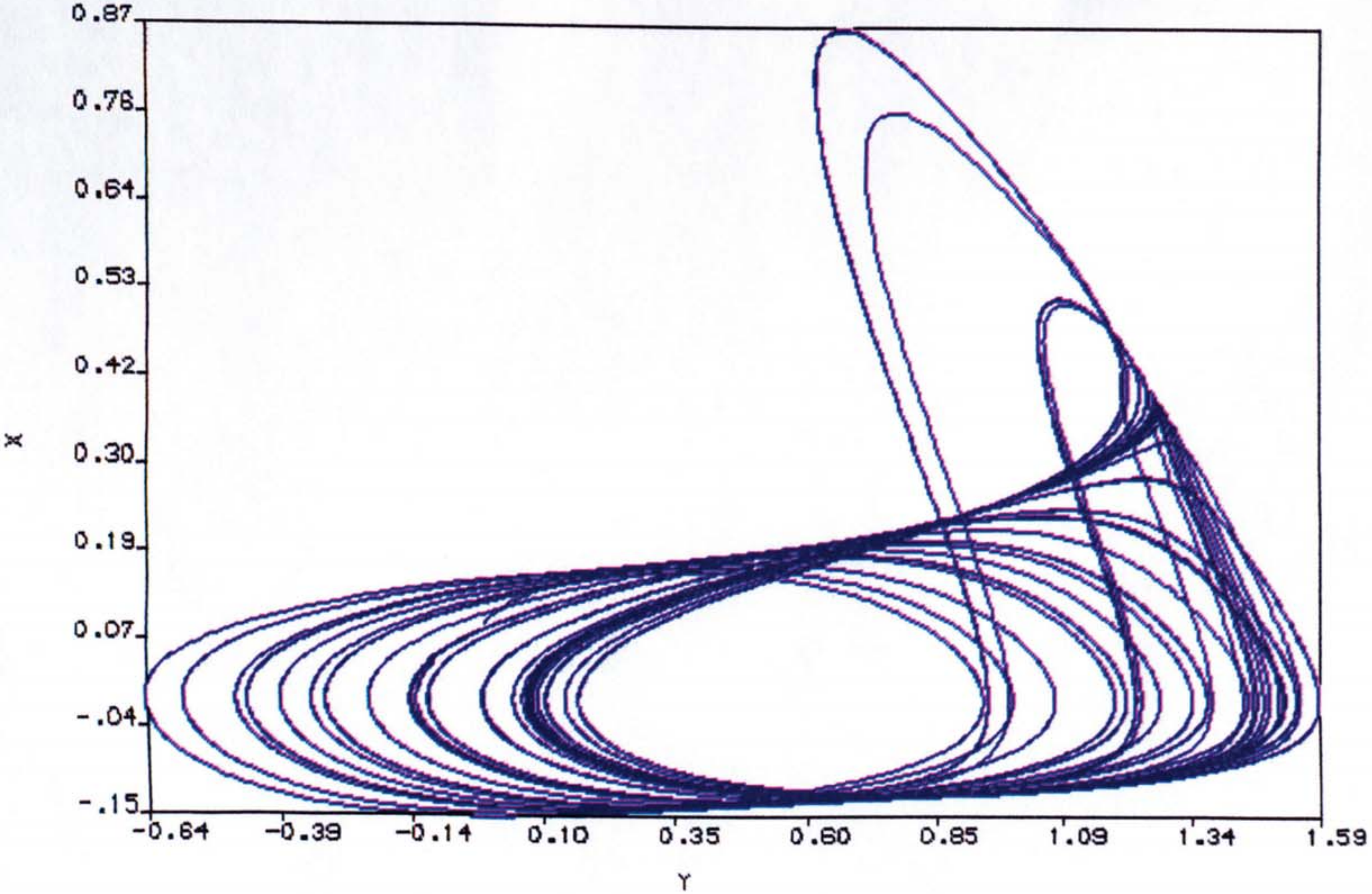


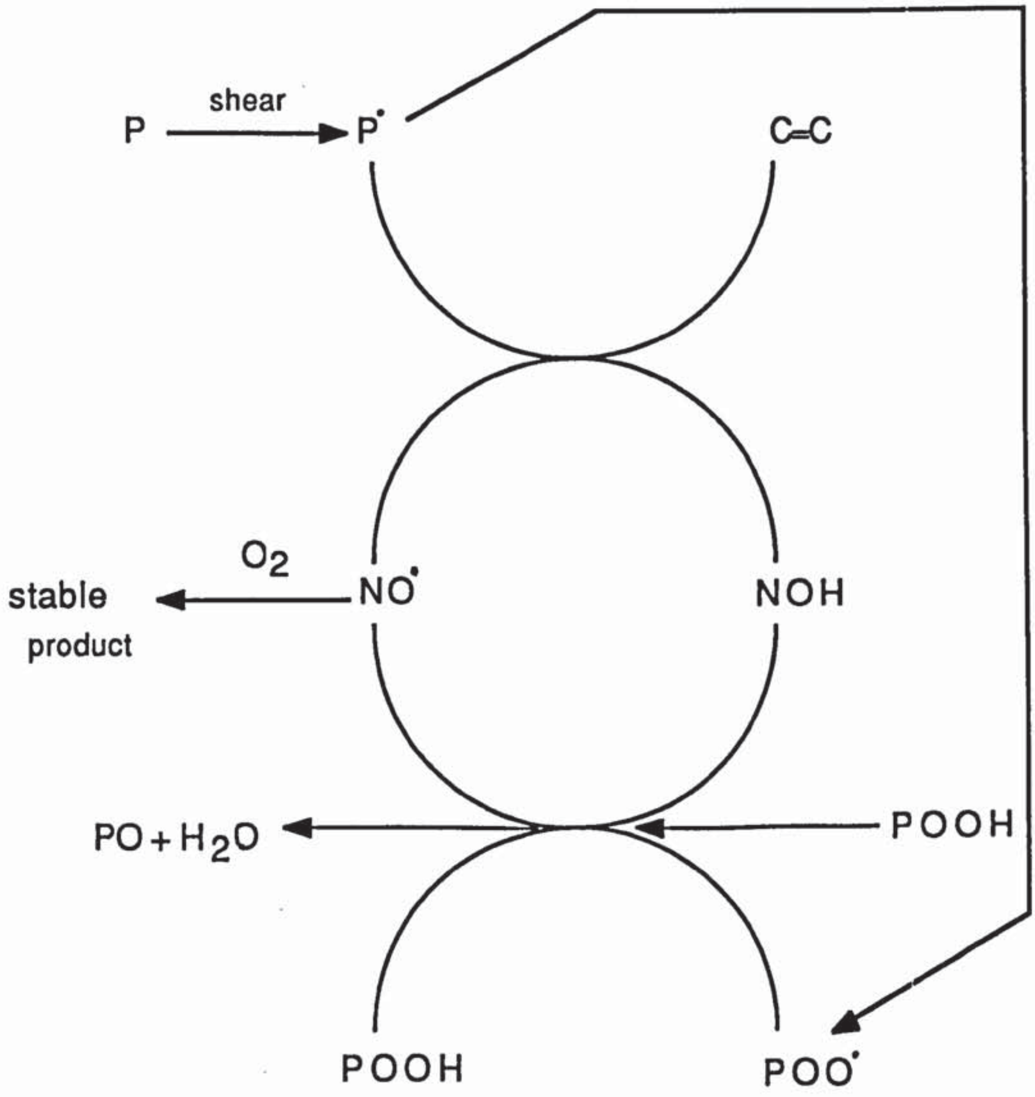
Fig5.16 Chaotic trajectory with a=0.5,b=0.01,c=0.1.

5.4.2 The Detailed Model of Nitroxyl Polypropylene System

The model devised in section5.4.1 displayes oscillation, but because one of the reaction mechanisms used in it is chemically unacceptable, we need to look at more details into the actual reaction mechanism in the nitroxyl polypropylene system. From experimental observation (see section 5.1), it is known that oxygen plays a crucial role in the regeneration process of NO and NOH. Actually, the closed mixing and open mixing procedures differ from each other in that in one case there is only limited amount of oxygen present in the reaction system, and in the other case the oxygen is in excess. Therefore, in a reasonable model for this reaction process, it is necessary to consider the influence of oxygen. The reactions used for the model should be based on experimental evidence.

5.4.2.1 Reaction Mechanism

It is known[142,160] that two phenomena are clear. One is that when the oscillation occures, the total concentrations of NO and NOH remain almost constant. The second is that the hydroperoxide did not appear to accumulate in large amount. However, the most probable reaction of NOH leading to regeneration of NO would also lead to the formation of hydroperoxide (see reaction (5.11)). This suggests to us that in the nitroxyl-polypropylene reaction system there must be a mechanism for the consuming of POOH formed by the reaction between NOH and POO. By comparing the available data, especially the reactions listed in the (R4), we can diagramatically express the regeneration process and other related reactions in Scheme5.1. From this scheme, it is clear that as long as the decomposition of NO into stable molecular product is very limited, the cycle between NO and NOH can be maintained, provided that certain amount of oxygen is present in the system. However, when the transformation of NO into stable end products is fast, there will not be enough NO to generate NOH which in turn would be able to form NO by reacting with peroxide and other species. Our modelling is then based on these considerations.



Scheme5.1 Reaction mechanism in nitroxyl-polypropylene (NO-P) reaction system

The corresponding reaction steps for this scheme are as followes:

$$P^{\bullet} + O_2 \xrightarrow{k_1} POO^{\bullet} \tag{5.22}$$

$$P^{\bullet} + NO^{\bullet} \xrightarrow{k_2} NOH + C{=}C \tag{5.23}$$

$$NOH + POO^{\bullet} \xrightarrow{k_3} NO^{\bullet} + POOH \tag{5.24}$$

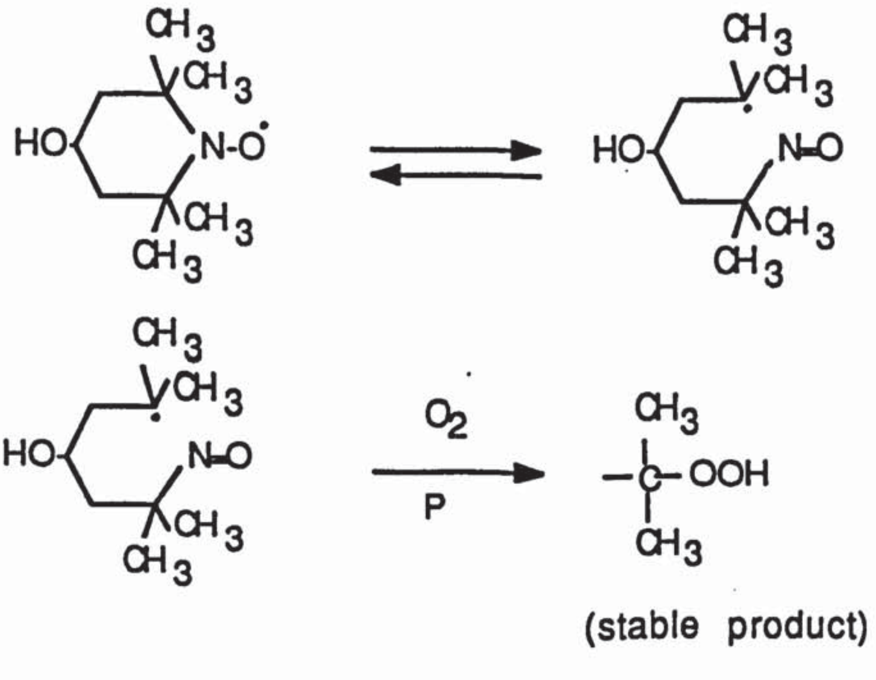
$$NOH + POOH \xrightarrow{k_4} NO^{\bullet} + PO^{\bullet} + H_2O \tag{5.25}$$

$$NOH + PO^{\bullet} \xrightarrow{k_5} NO^{\bullet} + POH \tag{5.26}$$

$$NO^{\bullet} + P \xrightarrow{k_6} NOH + P^{\bullet} \tag{5.27}$$

$$NO^{\bullet} + O_2 \xrightarrow{k_7} \text{stable product} \tag{5.28}$$

In this scheme, reaction (5.28) is actually consisted of following three steps:



For the reason of simplicity, only the overall process of NO being converted to stable product is represented in the reaction mechanism. It is thought that, under the condition of closed mixing, this conversion is very limited, which is denoted by smaller value of k_7, and under the condition of open mixing, the conversion is considerable, which is denoted by larger value of k_7.

By introducing the following notations:

$C_0=[O_2]_0$

$C_1=[P]$

$C_2=[P\ ]$

$C_3=[PO\ ]$

$X_1=[NO\ ]$

$X_2=[NOH]$

$X_3=[O_2]$

$X_4=[POO\ ]$

$X_5=[POOH]$

The ODEs for this reaction model are:

$$\dot{X}_1=k_3X_2X_4+k_4X_2X_5+k_5X_2C_3-k_2X_1C_2-k_6X_1C_1-k_7X_1X_3 \tag{5.29}$$

$$\dot{X}_2=k_2X_1C_2+k_6X_1C_1-k_3X_2X_4-k_4X_2X_5-k_5X_2C_3 \tag{5.30}$$

$$\dot{X}_4=k_1C_2X_3-k_3X_2X \tag{5.31}$$

$$\dot{X}_5=k_3X_2X_4-k_4X_2X_5 \tag{5.32}$$

5.4.2.2 Model for Oxygen Concentration

From the reaction mechanism, it is clear the concentration of O_2 play a very important role, because the chemical components involved in the regeneration process of NO and NOH are also related to oxygen. Without oxygen, there is no formation of peroxide ($POO^{\bullet}$), and the subsequent reactions of NO and $POO^{\bullet}$, POOH and NOH would not be possible. It is not known exactly how the concentration of oxygen changes during the reaction due to the experimental difficulties in measuring it. However, from the analysis of the experimental conditions, it can be said that the diffusion of oxygen into the system is very limited, and because the varying rate of reaction of O_2 in the reaction process, there may be a periodic fluctuation of the amount of oxygen existed in the reaction system. Since in the reaction mechanism proposed above O_2 is consumed, the total amount of oxygen tends to decay in exponential term if the operation involved is closed mixing. Therefore, the overall changing pattern of oxygen changes may be assumed to be exponential reduction with a periodic fluctuation. The following function may serve this purpose:

$$[O_2]=C_0(1.0 + A\cos(\omega t))e^{(D_f-K_1C_2)t} \qquad (5.33)$$

where

D_f ______ diffusion rate of O_2

A ______ amplitude of O_2 oscillation

ω ______ oscillation frequency of O_2

t ______ reaction time

The others are the same as in the Eq(3.22) to Eq(3.28) . The change pattern is shown in Fig.5.17. Therefore, the full model for this reaction system is:

$$\dot{X}_1=k_3X_2X_4 + k_4X_2X_5 + k_5X_2C_3-k_2X_1C_2 -k_6X_1C_1 -k_7X_1X_3 \qquad (5.34)$$

$$\dot{X}_2=k_2X_1C_2 + k_6X_1C_1 - k_3X_2X_4- k_4X_2X_5- k_5X_2C_3 \qquad (5.35)$$

$$\dot{X}_3=C_0\{(Df-k_1C_2)[1.0+A\cos(\omega t)]-A\omega\sin(\omega t)\}e^{(Df-K_1C_2)t} \qquad (5.36)$$

$$\dot{X}_4=k_1C_2X_3 - k_3X_2X \qquad (5.37)$$

$$\dot{X}_5=k_3X_2X_4 - k_4X_2X_5 \qquad (5.38)$$

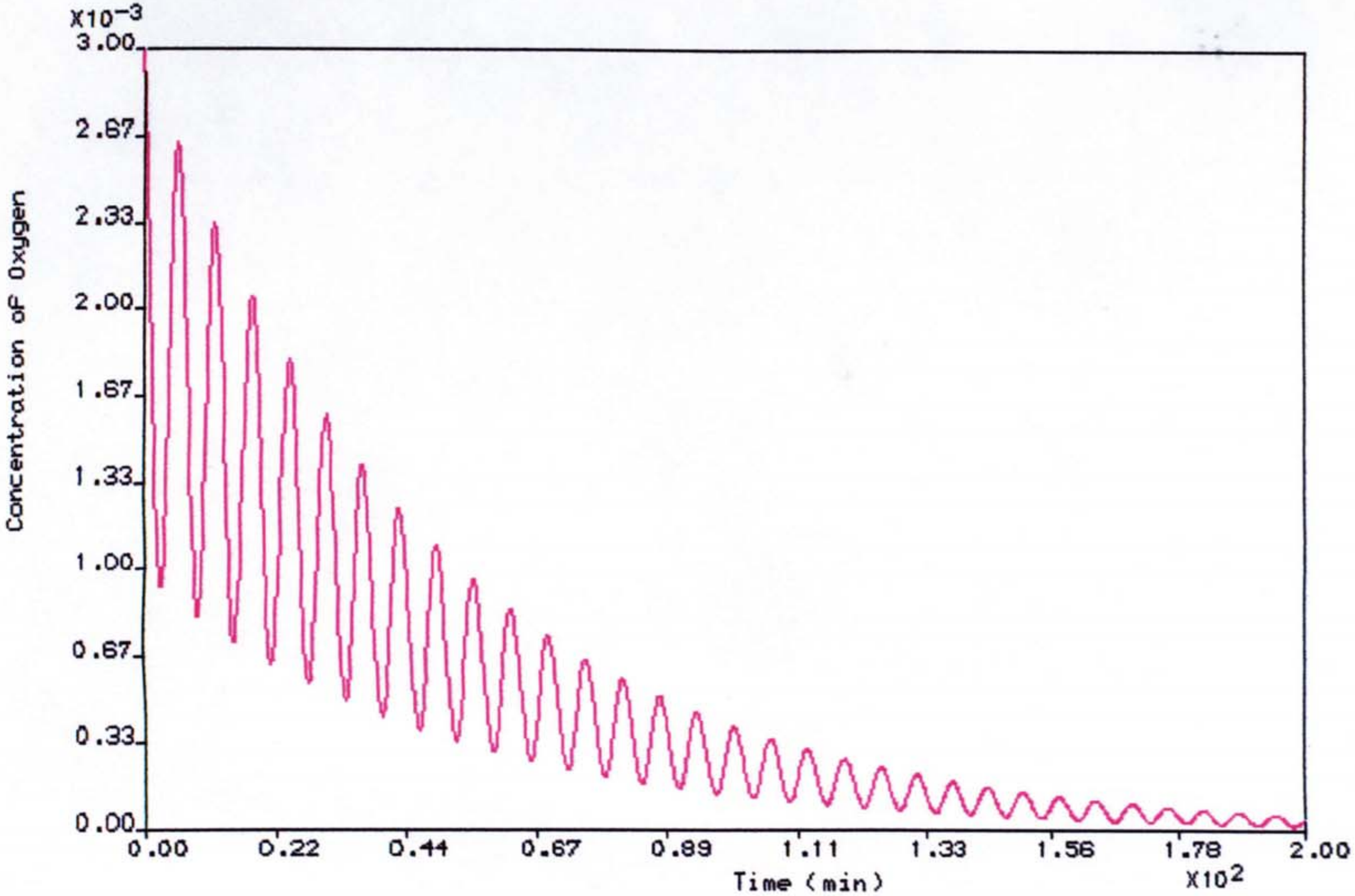


Fig5.17 Change pattern of oxygen concentration in the model.

Obviously, this ODE system can only be solved by numerical method. The numerical method used is the fourth order Runge-Kutta method. The implementation of this method for our particular need is listed in Appendix B.

5.3.3 Simulation Results

The known parameters in the above model are:

$C_0 = 2.0 \times 10^{-3}$ mol g^{-1} (ref. 142)

$C_1 = 2.0 \times 10^{-5}$ mol g^{-1}

$k_1 = \sim 10^6$ l^{-1} mol s^{-1} ($130\ {}^0C$)[159]

$k_3 = 5.0 \times 10^5$ l^{-1} mol s^{-1} ($130\ {}^0C$)[161]

$k_6 = \sim 10^4$ l^{-1} mol s^{-1} ($130\ {}^0C$)[162]

C_1 was calculated on the basis of assuming the average molecular weight of polypropylene used in the reaction to be 500,000. Because the values of rate constants listed above are the values obtained at reaction temperature of 130 ^0C, our experiments were carried out at 180 ^0C, it is necessary to adjust the values of the rate constants. The adjustment of these values was according to the rule of thumb, i.e., the K values will be increased by one order of magnitude. Therefore,

$k_1 = \sim 10^7$ l^{-1} mol s^{-1} ($180\ {}^0C$)

$k_3 = 5.0 \times 10^6$ l^{-1} mol s^{-1} ($180\ {}^0C$)

$k_6 = \sim 10^5$ l^{-1} mol s^{-1} ($180\ {}^0C$)

Other parameters were chosen basically by trail and error. The known parameters and the estimated values of other parameters were introduced into the Eqs(5.34) -(5.38), and the solutions were compared with the experimental parameters to decide the choice of parameter. In this way, values for other parameters were selected as follows:

$C_2=1.0 \times 10^{-6}$ mol g^{-1}

$C_3=1.0 \times 10^{-5}$ mol g^{-1}

$k_1=5.01 \times 10^{7}$ l^{-1} mol s^{-1}

$k_2=1.0 \times 10^{6}$ l^{-1} mol s^{-1}

$k_3=5.0 \times 10^{6}$ l^{-1} mol s^{-1}

$k_4=1.0 \times 10^{5}$ l^{-1} mol s^{-1}

$k_5= 1.0 \times 10^{5}$ l^{-1} mol s^{-1}

$k_6= 1.0 \times 10^{5}$ l^{-1} mol s^{-1}

These values were used throughout the numerical experiments. Other parameters depend on whether the operation were closed mixing or open mixing. The numerical results are listed below.

5.4.3.1 Results under restricted oxygen access

In this case, the oxygen concentration were assumed to change as described in section 5.4.3. The parameters for oxygen changes were:

$D_f=5.0$ s^{-1}

$\omega=1.0$ s^{-1}

$A=0.5$

Reaction(5.28) is formulated to take into account of the decomposition of $N\mathring{O}$. In the case of closed mixing, its rate is very small comparing with other reactions, $k_7=1.0 \times 10^{2}$ l^{-1} mol s^{-1}. From Eqs(5.34) - (5.35), it is known that the rate of total concentration ([NO] + [NOH]) change is:

$$\mathring{X}=\mathring{X}_1+\dot{X}_2=-k_7X_1X_3 \qquad (5.39)$$

because X_1 is about 10^{-5} and X3 10^{-3}, this makes $\dot{X}$ to be 10^{-6}. This means that the rate of total concentration changes is negligible as observed exprimentally in the case of closed mixing (see section5.2

and Fig5.1).

(1) When $NO^{\bullet}$ is the reagent

The initial conditions are:

$X_1(0)=1.16 \times 10^{-5}$ mol g^{-1}

$X_2(0)=0.0$ mol g^{-1}

$X_3(0)=C_0$ mol g^{-1}

$X_4(0)=0.0$ mol g^{-1}

$X_5(0)=0.0$ mol g^{-1}

The result is shown in Fig.5.18, which corresponds to experimental result shown in Fig.5.19.

(2) When NOH is the reagent

The initial conditions are:

$X1(0)=0.0$ mol g^{-1}

$X_2(0)=1.16 \times 10^{-5}$ mol g^{-1}

$X_3(0)=C_0$ mol g^{-1}

$X_4(0)=0.0$ mol g^{-1}

$X_5(0)=0.0$ mol g^{-1}

The result is shown in Fig.5.20, which corresponds to the experimental result shown in Fig.5.21.

5.4.3.2 Results under excess oxygen concentration

In the case of open mixing, two characteristics must be taken into account: (a). decomposition of $NO^{\bullet}$ into stable molecular product, which means the rate of reaction(5.28) must be higher than that in the case of closed mixing, since the oxygen does not decay in this case, thus k_7 was chosen to be 2.5×10^4; (b). oxygen concentration chnages differently from the case of closed mixing. It does not decay

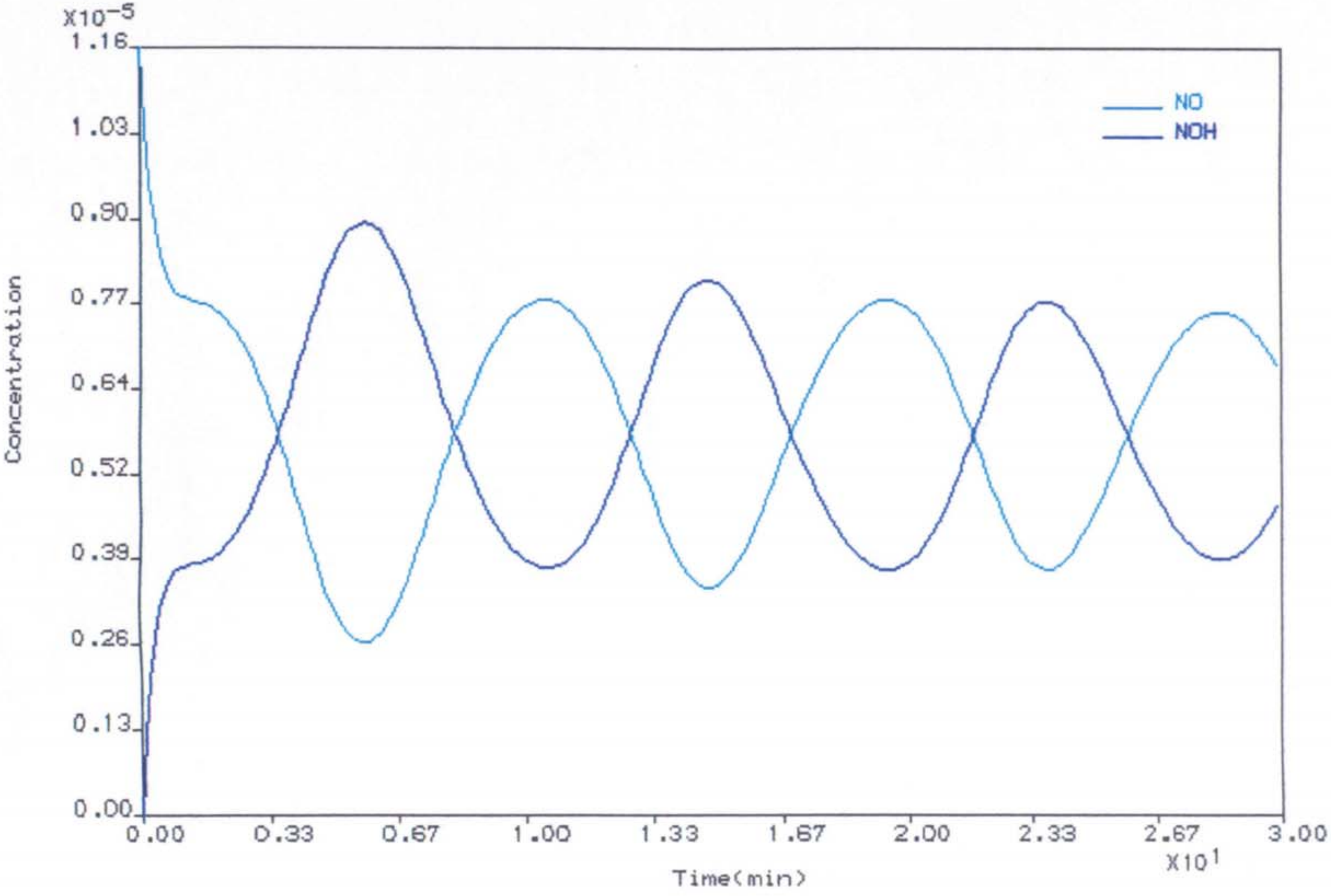


Fig5.18 Simulation result(NO is starting reagent).

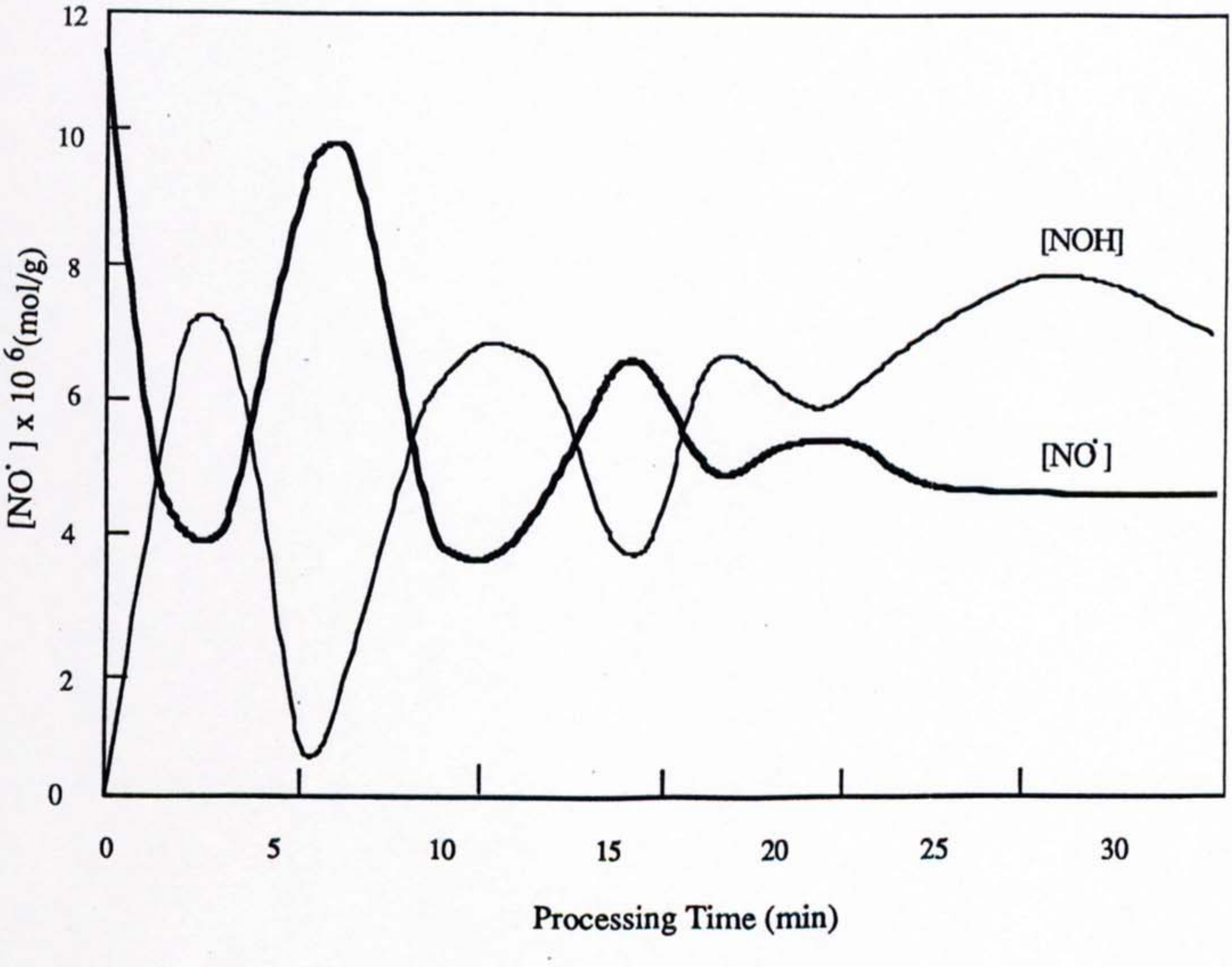


Fig5.19 Changes in [NO˙] , [NOH] concentrations (CM), starting with [NO˙]

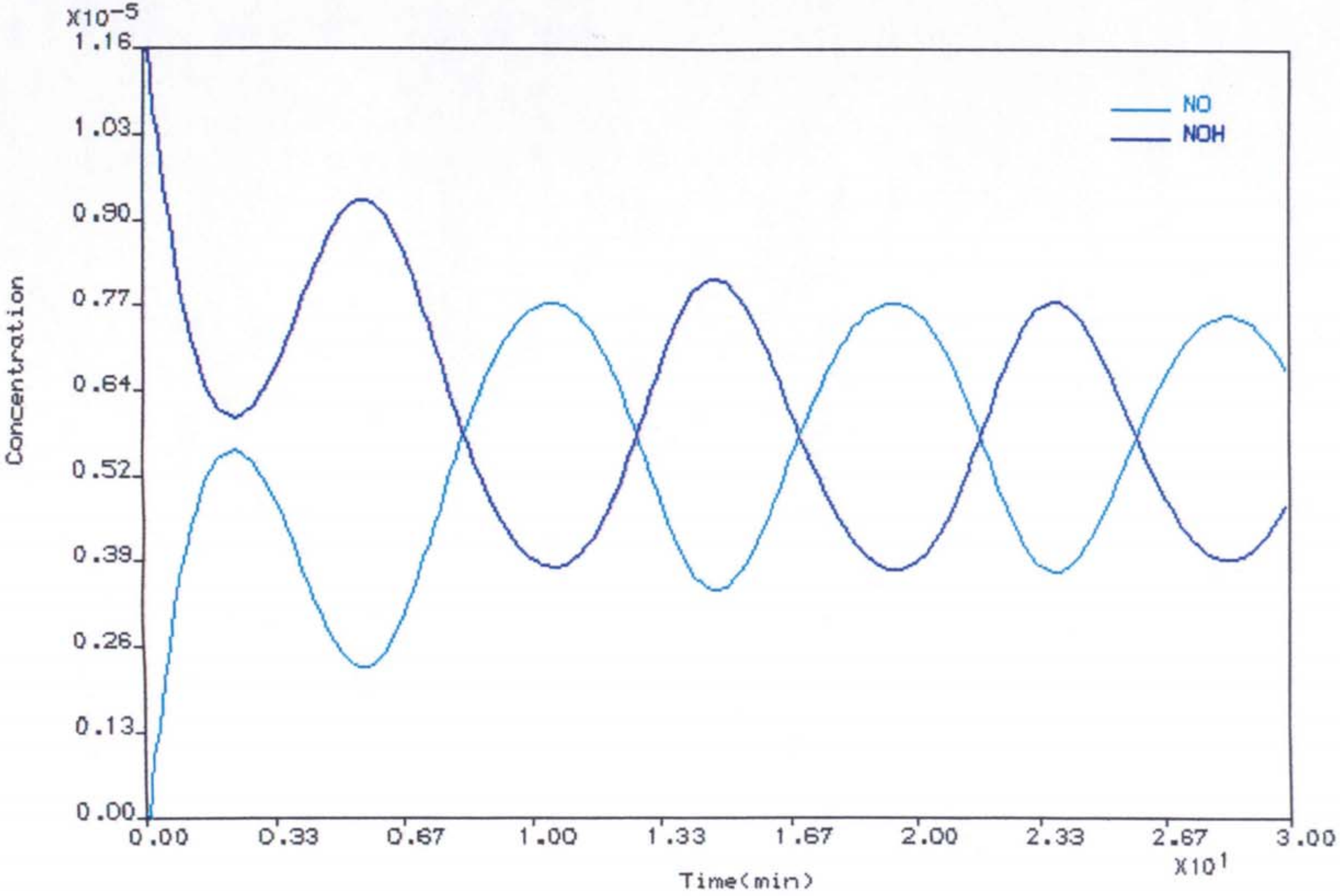


Fig5.20 Simulation result(NOH is starting reagent).

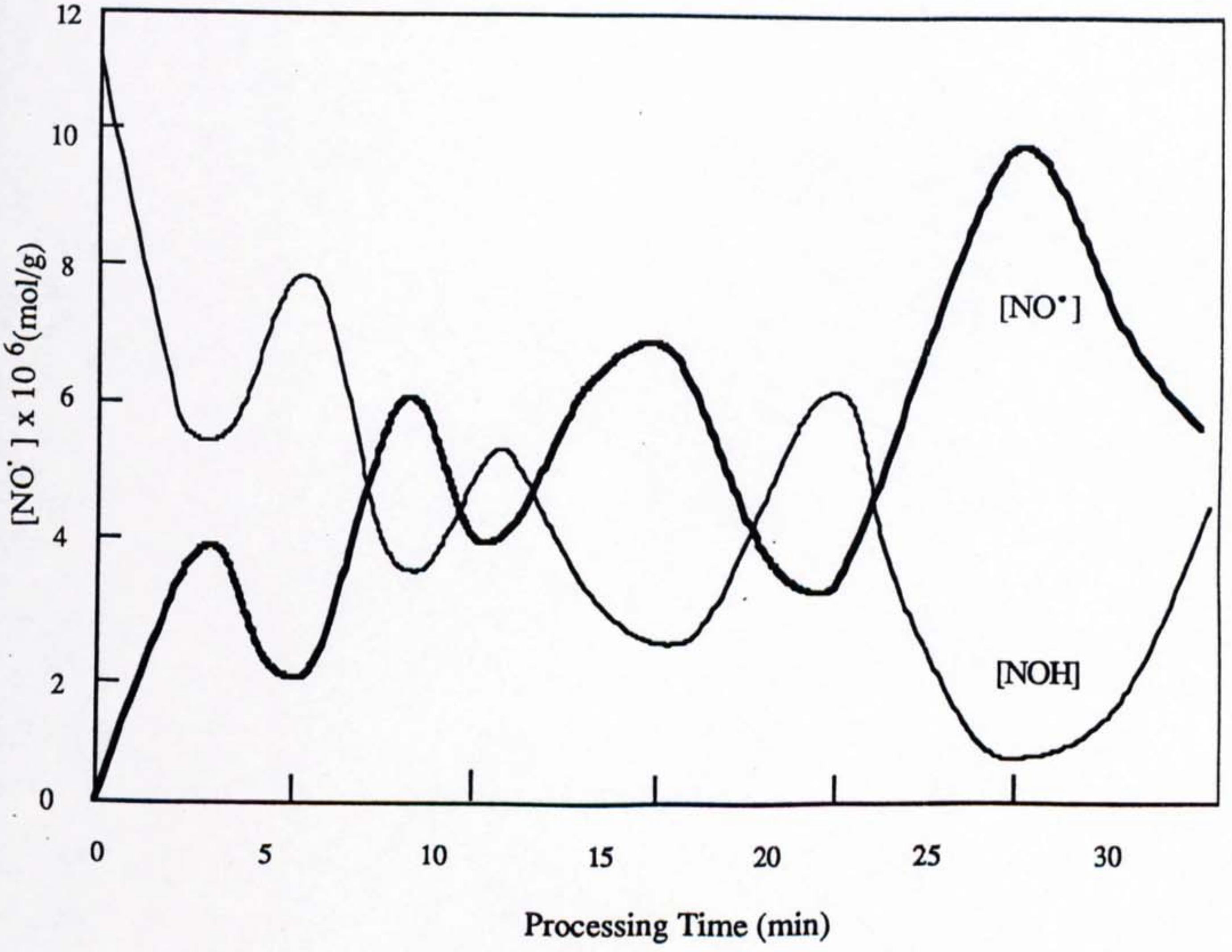


Fig5.21 Changes in [NO˙] and [NOH] (CM), starting with [NOH]

as it does in the closed mixing, which requires the exponential term in(5.36) should remain almost 1, and the oscillation term should also nearly vanish, i.e., A~0, ω~0. Therefore, for open mixing:

K_7=2.5 x 10^4 l^{-1} mol s^{-1}

D_f=5.1 s^{-1}

ω = 0.05

A=0.01

(1) When $NO^\bullet$ is the starting reagent

The initial conditions are:

X1(0)=1.16 x 10^{-5} mol g^{-1}

X_2(0)=0.0 mol g^{-1}

X_3(0)=C_0 mol g^{-1}

X_4(0)=0.0 mol g^{-1}

X_5(0)=0.0 mol g^{-1}

The result is shown in Fig.5.22, which corresponds to the experimental result shown in Fig.5.23..

(2) When NOH is the starting reagent

The initial conditions are:

X1(0)=0.0 mol g^{-1}.

X_2(0)=1.16 x 10^{-5} mol g^{-1}

X_3(0)=C_0 mol g^{-1}

X_4(0)=0.0 mol g^{-1}

X_5(0)=0.0 mol g^{-1}

The result is shown in Fig.5.24, which corresponds to the experimental result shown in Fig.5.25 .

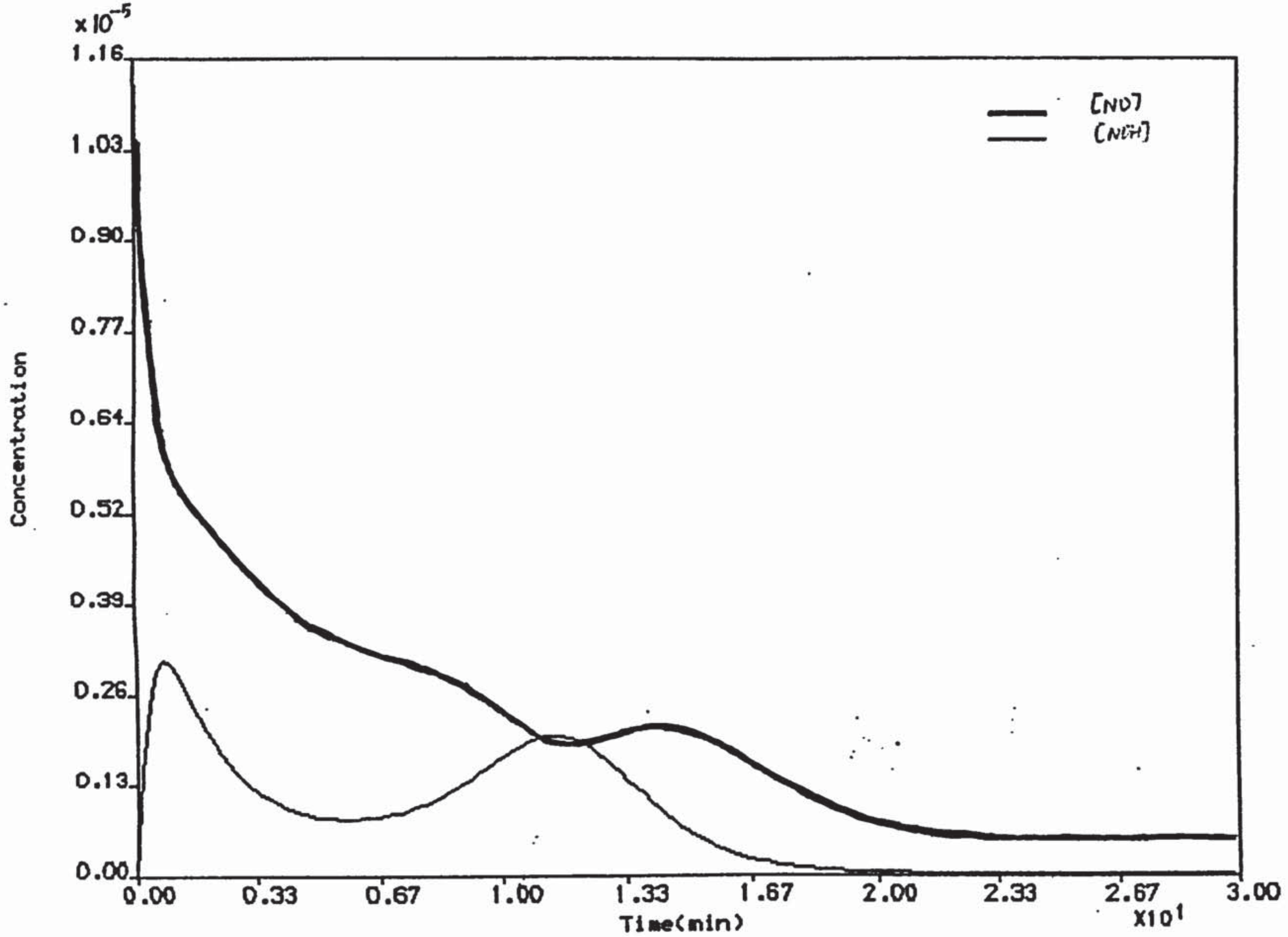


Fig5.22 Simulation result(OM,NO is reagent).

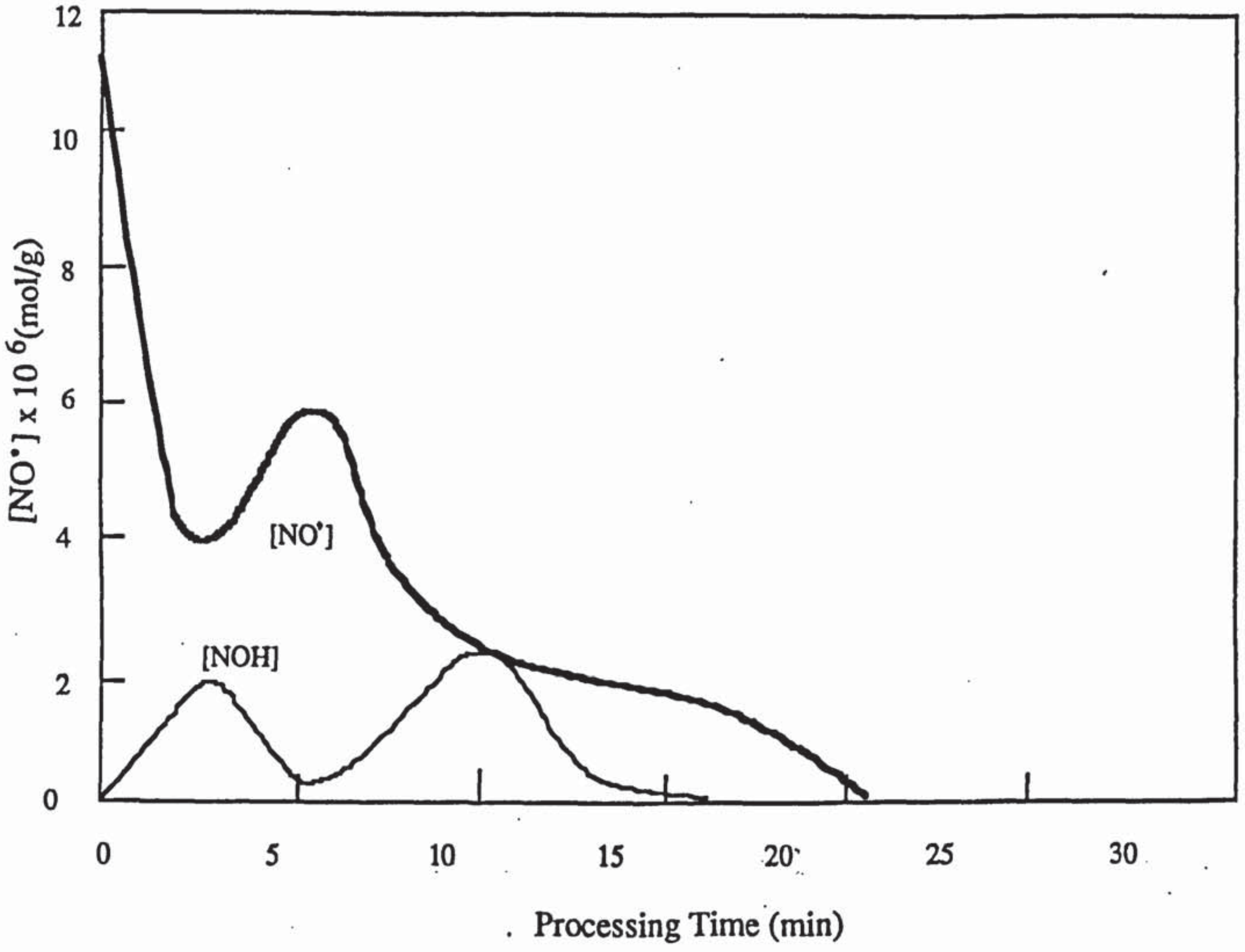


Fig5.23 Changes in [NO•] , [NOH] concentrations (OM), starting with [NO•]

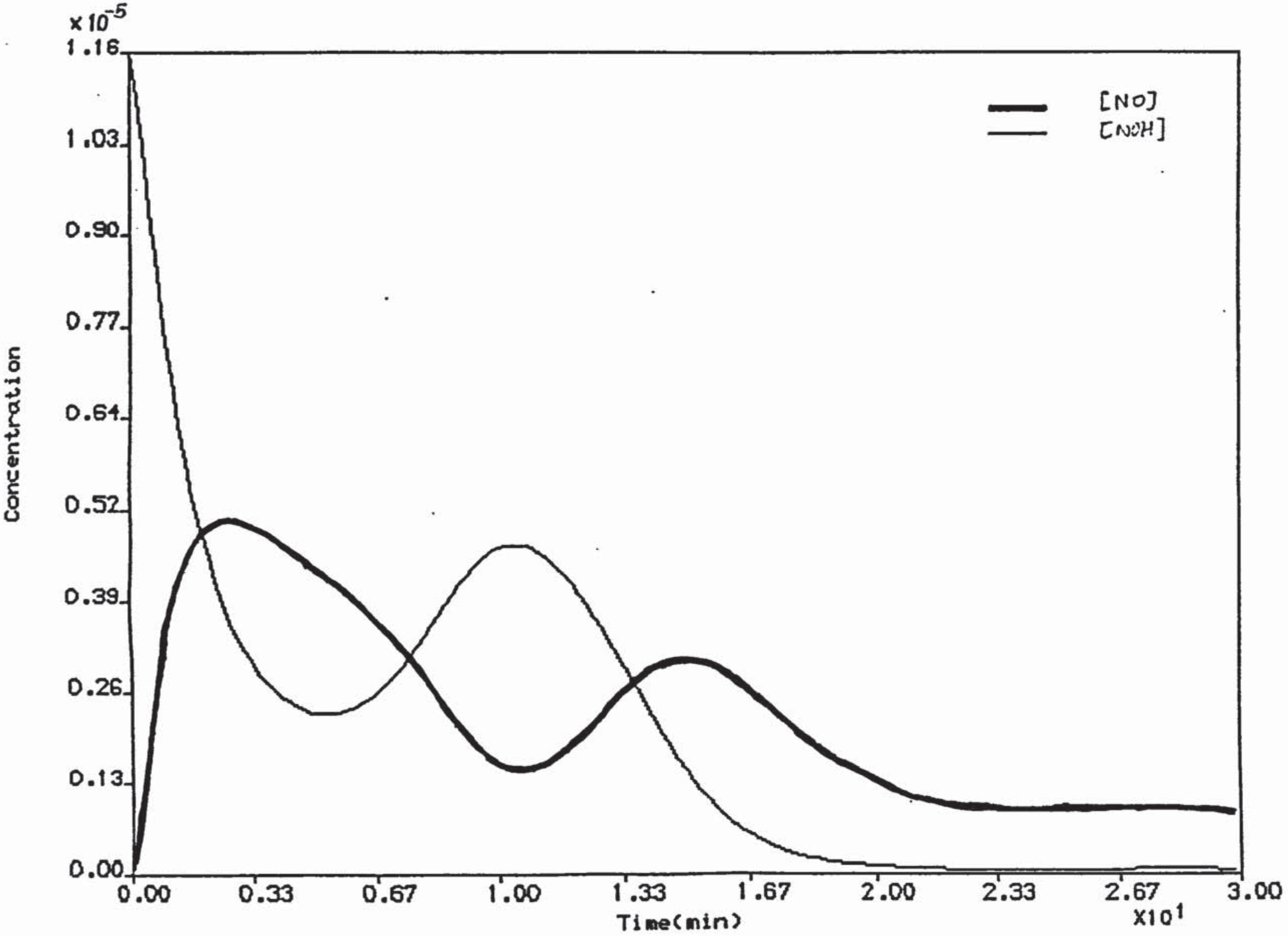


Fig5.24 Simulation result(OH,NOH is reagent).

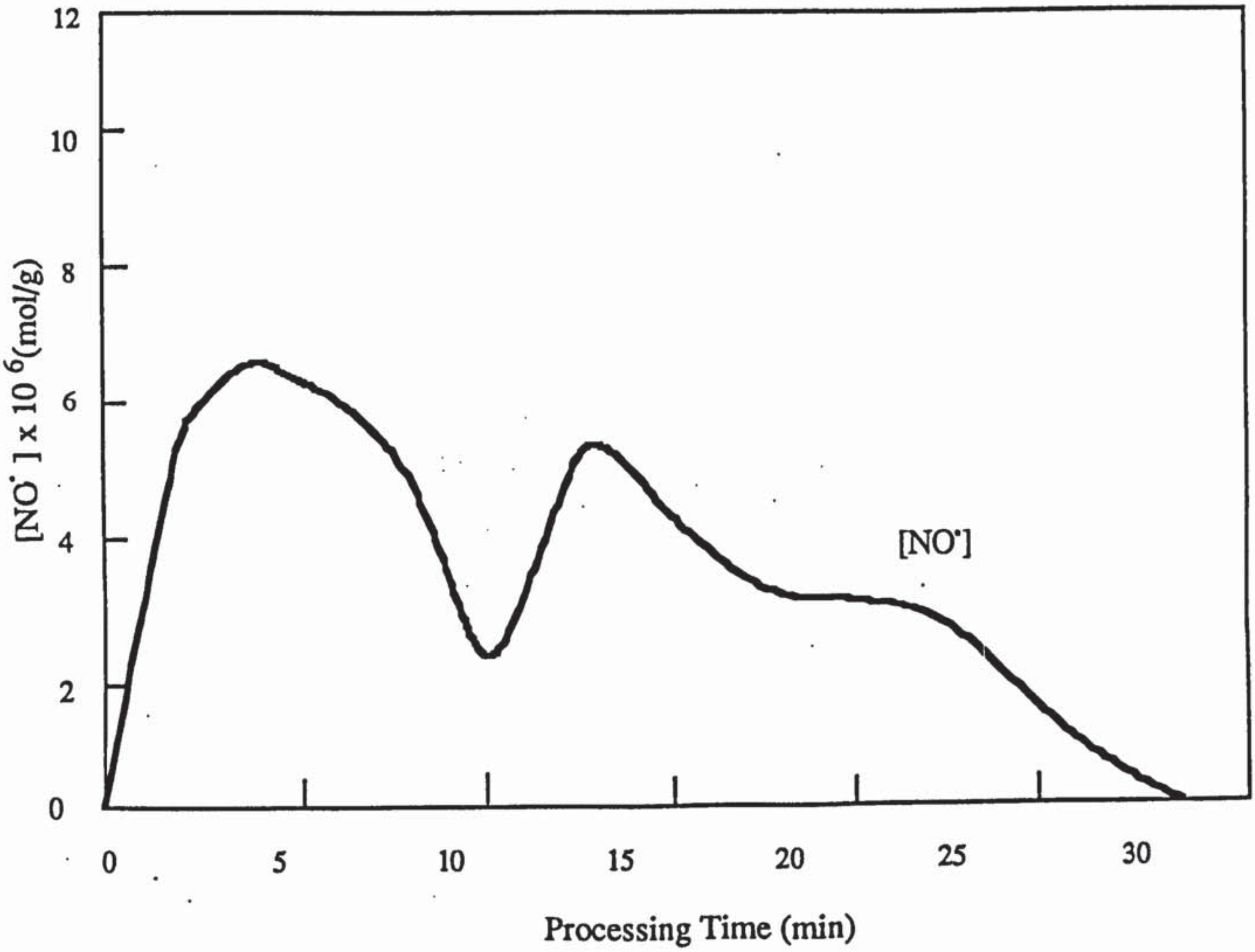


Fig5.25 Changes in [NO˙] and [NOH] (OM), starting with [NOH]

5.4 Conclusions and Suggestions for Further Work

From the discussions in sections5.1 to 5.4, it is clear that chemical oscillation has become a very important part of chemistry. The extensive research into this phenomenon has provided insight into some fundamental aspects of chemical reactions. As regard to our work on polymer mixing, the following points may be made:

1. The oscillation of nitroxyl concentration during their reaction with polymer molecules in the presence of small amount of oxygen is a genuine phenomenon. The explanation of this phenomenon helps us to understand the major reactions in the stabilisation process of polypropylene by nitroxyl radicals.

2. So far, chemical oscillations have mainly been found in the gas phase and liquid phase, or the heterogeneous phase of both. The reactions of nitroxyl radicals with polymer molecules were carried out in the polymer melt. It is reasonable to say that this oscillation should belong to a new catagory.

3. The first mathematical model(see section5.3) formed in this research has similarity to the Brusselator. But, this was found to be a modification. Firstly, the Brusselator only describes the intermediates. In other words, even when the initial concentrations of ιwo variables are zero, it is still possible to observe subsequent oscillations. In our model, one component is necessary at the beginning of the reaction process. Secondly, this model not only displays steady state and sustained periodic oscillation as does the

Brusselator but, it also exhibits chaotic behaviour in certain parameter range. This latter characteristics makes this model more flexible.

4. The detailed modelling on the basis of an actual experimental data derived the model (5.34)~(5.38). Numerical simulations gave the results which are in good agreement with the experimental observations both in the case of closed mixing and open mixing. The rate constants given by the model may serve as basis for further study in the reactions of nitroxyl and polypropylene.

On the basis of the work done so far, the following further works are recommended:

1. Experimentally, the collection of data should be carried out over a longer time period, and at shorter time intervals in order to investigate the kinetic processes in more detail.

2. It would be very interesting to carry out the experiment under the condition of the artificially made oscillation of oxygen concentration, because the model proposed for the changes of oxygen concentration involves certain degree of oscillation. Therefore, if we could experimentally create observable oscillation of oxygen concentration, it is of interest to investigate its influence on the changes of [NO] and [NOH].

3. It is of interest to observe the effect of the nitroxyl oscillation on

the properties of polymer, e.g. oven aging performance, etc.

4. The mathematical model can be modified to include the temperature effect. The values used for the simulation were fixed for one temperature. If the temperature dependence of rate constants were introduced into the model, then it is possible to predict the behaviours of [NO-PP] under different temperatures.

6. It is possible to simulate the oscillation behaviour of galvinoxyl system[163] by using the model constructed.

7. The oscillation of nitroxyl radical concentration in PVC[164]can also be treated in the similar way.

References

1. W. C. Geddes, Rubb. Chem. Tech., **40**, 177(1967).
2. M. Onozuka and M. Asahina, J. Macromol. Sci., Rev. Macromol. Chem., **C3**,235(1969).
3. D. Braun, Pure Appl. Chem., **26**, 173(1971).
4. Z. Mayer, J. Macromol. Sci., Rev. Macromol. Chem., **C10**, 263(1974).
5. G. Ayrey, et al, J. Polym. Sci., Macromol. Rev., **8**, 1(1974)
6. D. Braun, in Degradation and Stabilisation of Polymers, G. Geuskens, Ed., John Wiley, New York, 23(1975)
7. W. H. Starnes, Adv. Chem. Ser., **169**, 309(1978).
8. D. Braun, in Developments in Polymer Degradation-3, N. Grassie, Ed., Applied Science Publishers, London, 101(1981).
9. W. H. Starnes, in Developments in Polymer Degradation-3, N. Grassie, Ed., Applied Science Publishers, London, 135(1981).
10. G. Scott, Ed., Developments in Polymer Stabilisation-2, Applied Science Publishers, London (1980).
11. G. Palma and M. Carenza, J. Appl. Polym. Sci., **14**, 1737(1970).
12. M. Asahina and .M. Onozuka, J. Polym. Sci., **A1**, **2**, 3505,3515(1964).
13. G. Scott , M. Tahan and J. Vyvoda, Eur. Polym. J., **14**, 377(1978).
14. K. B. Abbas and E. M. Sorvik, J. Appl. Polym. Sci., **19**, 2991(1975).
15. W. C. Geddes, Eur. Polym. J., **3**, 747(1967).
16. F. Tudos and T. Kelen, in Macromolecular Chemistry, Vol.8, K. Saarela, Ed., Butterworths, London, 393(1973).

17. K. B. Abbas and R. J. Lawrence, J. Polym. Sci., Polym. Chem. Ed., **13**, 1889(1975).

18. F. Tudos, et al, Pure Appl. Chem., **38**, 201(1974).

19. R. R. Stromberg, et al, J. Polym. Sci., **35**, 355(1959).

20. D. E. Winkler, J. Polym. Sci., **35**, 3(1959).

21. E. J. Arlman, J. Polym. Sci., **12**, 547(1954).

22. C. H. Stapfer and J. D. Granick, J. Polym. Sci., **A1, 9**, 2625(1971).

23. R. V. Albarino, et al, J. Polym. Sci., **A1, 9**, 15(1971).

24. T. Nakagawa and M. Okawara, J. Polym. Sci., **A1, 6**, 1975(1968).

25. G. Palma and M. Carenza, J. Appl. Polym. Sci., **16**, 2485(1972).

26. W. H. Gibb and J. R. MacCallum, Eur. Polym. J., **7**,1231(1971).

27. W. H. Gibb and J. R. MacCallum, Eur. Polym. J., **8**,1223(1971).

28. E. D. Owen and R. J. Bailey, J. Polym. Sci., **A1, 10**, 113(1972).

29. Y. Landler and P. Lebel, J. Polym. Sci., **48**, 477(1960).

30. V. P. Gupta and L. E. St Pierre, J. Polym Sci., **A1, 8**, 37(1970).

31. A. Guyot, et al, J. Polym. Sci., **A1, 8**, 1596(1970).

32. S. A. Liebman, et al, J. Polym. Sci., **A1, 9**, 1823(1971).

33. I. Ouchi, J. Polym. Sci., **A1, 3**, 2685(1970).

34. C. H. Bamford and D. F. Fenton, Polymer, **10**, 63(1969).

35. R. A. Papko and V. S. Padov, Polym. Sci. USSR, **16**, 1636(1974).

36. V. N. Myakov and B. B. Troitskii, Vysokomol. Soyed., **4**, 876(1962).

37. I. C. McNeil and D. Neil, Makromol. Chem., **117**, 265(1968).

38. I. C. McNeil and D. Neil, Eur. Polym. J., **6**, 143 (1970).

39. I. C. McNeil and D. Neil, Eur. Polym. J., **6**, 569(1970).

40. B. Baum, S. P. E. J., **17**, 71(1961).

41. D. Braun and R. F. Bender, Eur. Polym. J. (Suppl.), 269(1969).

42. W. I. Bengough and H. M. Sharpe, Makromol. Chem., **66**, 31(1963).

43. J. N. Hay, J. Polym. Sci., **A1, 8**, 1201(1970).

44. B. B. Troitsikii, et al, J. Polym. Sci., Polym. Symp., **42**, 1347(1973).

45. G. C. Marks, et al, S. C. I. Monogr., **26**, 204(1967).

46. M. M. Zafar and R. Mahmood, Eur. Polym. J., **12**, 333(1976).

47. Z. Mayer, et al, Chem. Ind. (London), 508(1965).

48. S. Van der Ven and W. F. DeWit, Angew. Makromol. Chem., **8**, 143(1969).

49. Z. J. Wolkober, J. Polym. Sci., **58**, 1311(1962).

50. Y. Shindo and T. Hirai, Makromol. Chem., **155**, 1(1972).

51. A. Reiche, et al, Kunststoffe, **52**, 265(1962).

52. L. S. Troitskaya and B. B. Troitskii, Plast. Massy, **7**, 46(1966).

53. T. V. Hoang, et al, Eur. Polym. J., **11**, 469(1975).

54. I. K. Varma and S. S. Grover, Makromol. Chem., **175**, 2515(1974).

55. A. A. Berlin, et al, in Aging and Stabilisation of Polymers, M. B. Neiman, Ed., Consultants Bureau, New York, 175(1968).

56. R. C. Hardon and W. H. Starnes, Polym. Prepr., **18**, 505(1977).

57. R. C. Hardon and W. H. Starnes, ACS Adv. Chem. Ser., **169**, 333(1978).

58. G. Martinez, et al, J. Macromol. Chem., **A17**, 1129(1982).

59. J. Millan and J. Guzman, Eur. Polym. J., **12**, 299(1976).

60. J. Millan, et al, Angew. Makromol. Chem., **45**, 172(1975).

61. G. Martinez, et al, Makromol. Chem., **180**, 2937(1979).

62. J. Millan, et al, J. Polym. Sci., Polym. Chem. Ed., **18**, 505(1980).

63. K. Mitani and T. Ogata, J. Appl. Polym. Sci., **18**, 3205(1974).

64. D. Druesedow and C. G. Gibbs, Nat. Bur. Stand. Circular., **525**, 69(1953).

65. K. B. Abbas and E. M. Sorvik, J. Appl. Polym. Sci., **17**, 3577(1973).

66. T. Hjertberg and E. M. Sorvik, J. Appl. Polym. Sci., **22**, 2415(1973).

67. T. T. Nagy, et al, Polym. Bull., **3**, 613(1980).

68. T. Kelen, J. Macromol. Sci. Chem., **A12**, 349(1978).

69. T. Kelen, et al, Polym. Bull., **1**, 79(1978).

70. G. A. Razuvaev, et al, Dokl. Akad. Nauk. SSSR. Khim., **170**, 1342(1966).

71. B. B. Troikii and V. N. Myakov, Plast. Massy, **3**, 13(1967).

72. E. N. Zilberman, et al, Vysokomol. Soyed., **B16**, 46(1974).

73. M. M. O'Mara, Pure Appl. Chem., **49**, 649(1977).

74. S. A. Liebman, et al, J. Polym. Sci., Polym. Chem. Ed., **16**, 3139(1978).

75. M. M. O'Mara, Pure Appl. Chem., **49**, 649(1977).

76. R. P. Lattimer and W. J. Kroenke, J. Appl. Polym. Sci., **25**, 101(1980).

77. R. P. Lattimer and W. J. Kroenke, J. Appl. Polym. Sci., **27**, 1355(1982).

78. V. Bellenger, et al, Eur. Polym. J., **18**, 337(1982).

79. G. A. Rasuvayev, et al, J. Polym. Sci., **A1, 9**, 2637(1971).

80. T. T. Nagy, et al, Polym. Bull., **3**, 613(1980).

81. T. Suzuki, et al, J. Polym. Sci., **C33**, 281(1971).

82. V. V. Korshak, Russ. Chem. Rev., No.1, 1979.

83. A. B. Robertson, et al, Polym. Prepr., **12**,620(1971).

84. M. L. Bender and N. S. Neveu, J. Amer. Chem. Soc., **80**, 5388(1958).

85. H. Morawetz, in Macromolecular Reactions, E. Fetts, Ed., John Wiley, New York, 1968.

86. P. Flory, J. Amer. Chem. Soc., **61**, 1518(1939).

87. C. L. Arcus, J. Chem. Soc., 2732(1949).

88. N. G. Gaylord, J. Polym. Sci., **C24**, 1(1968).

89. H. Morawetz, in Chemical Reactions of Polymers, M. Fetts, Ed., Interscience Publishers, New York, 110(1964).

90. V. V. Korshak, Russ. Chem. Rev., **2**, 1201(1980).

91. A. Weihe, Kunststoffe, **31**, 52(1941).

92. G. B. Bachmann, et al, J. Org. Chem., **12**, 108(1947).

93. R. C. Fuson, et al, Org. React., **1**, 63(1942).

94. R. C. Houtz, et al, J. Amer. Chem. Soc., **55**, 1609(1933).

95 A. Charlesby, Proc. Roy. Soc. (London), SerA, **215**, 187(1952).

96. B. T. Hayes, et al, Chem. Ind., 1162(1957).

97. R. H. Kienle, et al, J. Amer. Chem. Soc., **61**, 2258(1939).

98. A. H. Frazer, et al, J. Polym. Sci., **A2**, 1157(1964).

99. D. B. Porrit, Trans. Faraday Soc., **16**, 81(1921).

100. K. Tada, J. Makromol. Chem., **71**, 71(1964).

101. N. Grassie, Chemistry of High Polymer Degradation Processes, Butterworths, New York, 1958.

102. O. Bayer, Angew. Chem., **A59**, 257(1947).

103. B. D. Coleman, J. Amer. Chem. Soc., **77**, 5472(1947).

104. A. H. Frazer, J. Polym. Sci., **A2**, 1137(1964).

105. C. S. Marvel, J. Amer. Chem. Soc., **81**, 448(1959).

106. N. W. Johnston, et al, J. Polym. Sci., **C**, 591(1969).

107. S. Iamoda, Nature, **193**, 261(1962).

108. N. Joda, J. Polym. Sci., **B4**, 551(1966).

109. H. Batzer, Makromol. Chem., **8**, 217(1952).

110. E. R. Bertozzi, J. Polym. Sci., **19**, 17(1956).

111. T. Hjertberg and E. M. Sorvik, in Dedradation and Stabilisation fo PVC, E. D. Owen, Ed., Appllied Science Publishers, 21(1984).

112. A. J. Arlman, J. Polym. Sci., **12**, 547(1954).

113. W. C. Geddes, Rubb. Tech. Chem., **40**, 177(1967).

114. D. Braun, Makromol. Chem., **99**, 59(1966).

115. K. B. Abbas, J. Appp. Polym. Sci., **19**, 2991(1975).

116. T. T. Nagy, et al, Polym. Bull., **3**, 613(1980).

117. K. B. Abbas, J. Macromol. Sci., Chem., **A12**, 479(1978).

118. K. P. Popov and L.V. Smirnov, Optika Spectroskopiya, **14**, 787(1963).

119. T. Kelen, Polymer Degradation, VNR, 1983.

120. J. Brandrup and E. H. Immergut, Polymer Handbook, John Wiley, II-141(1967).

121. P. F. Marconi, et al, Chemica ei' industria, Suppl., **7**, 1(1971).

122. K. F. O'Driscoll, in Computers in Polymer Sciences, J. S. Mattson, Ed., Marcell Dekker Inc., New York, 100(1977).

123. J. S. Mattson, ed., Computers in Polymer Sciences, Marcell Dekker Inc., New York, 1977.

124. M. Okawara and Y. Ochinai, in Modification of Polymers, C. E. Carraher,Jr., Ed., ACS Symp. Ser121, 41(1980).

125. Abdull, PhD Thesis, Aston University (1987).

126. J. E. Bras, et al, in Chemical Reactions of Polymers, M. Fetts, Ed., Interscience Publishers, New York, 204(1964).

127. H. R. Allcock and F. W. Lampe, Contemporary Polymer Chemistry, Prentice-Hall, New Jersey, 58(1981).

128. T. R. Crompton, Chemical Analysis of Additives in Plastics, Pergamon Press, 30(1971).

129. J. K. Sears and J. R. Darby, The Technology of Plasticizers, John Wiley, 480(1982).

130. I. Epstein, Chem. Eng. News, March, 2436(1987).

131. I. Epstein, Physica 7D, 47(1983).

132. T.S. Kuhn, The Structure of Scientific Revolution, University of Chicago Press, Chicago, 2nd ed., (1970).

133. L. Onsager, Phys. Rev., **37**, 405(1931).

134. L. Onsager, Phys. Rev., **38**, 2265(1931).

135. I. Prigogine, Introduction to Nonequilibrium Thermodynamics, New York, Wiley-Interscience (1962).

136. B. P. Belousov, Sbornik Referatov po Radiatsionni Meditsine, Medgiz, Moscow, 145(1951).

137. A. M. Zhabotinskii, Biofizika, **9**, 306(1964).

138. R.J. Field and M. Burger, Oscillation and Travelling Waves, John Wiley & Sons (1985).

139. A. T. Winfree, Science, **175**, 634(1972).

140. S. Scott, Chem. Brit., December, 1987.

141. R. Bagheri, K.B. Charkraborty and G. Scott, Polym. Deg. and Stab., 5,145(1983).

142. S. Al-Malaika, E.O. Omikorede and G. Scott, J. Appl. Polym. Sci., 33,703(1987).

143. I. Prigogine and R. Lefever, J. Chem. Phys., 48, 1695(1968).

144. Noyes, Field, Adv. Chem. Phys., 109, (1971).

145. R. J. Field and M. Burger, Oscillations and Travelling Waves in Chemistry, Johne Wiley & Sons (1985).

146. A. J. Lotka, J. Am. Chem. Soc., 42, 1595(1920).

147. P. Heinz-Otto, The Beauty of Fractals, Springer-Verlag, (1986).

148. I. Prigogine and R. Lefever, J. Chem. Phys., 48, 1695(1968).

149. A. Turing, Phil. Trans. Roy. Soc. London, B237, 37(1952).

150. Hao Bai-Lin, Chaos, World Scientific Press, Singapore, (1985).

151. A. V. Holden, Chaos, Manchester University Press,(1985).

152. K. Tomita and T. Kai, Phys. Lett., 66A, 91(1978).

153. R. J. Field, et al, J. Am. Chem. Soc., 94, 8649(1972).

154. ibid, J. Chem. Phys., 60,1877(1974).

155. S. P. Hastings and J. D. Murray, SIAM J. Appl. Math., 28, 678(1975).

156. I. D. Hsu, J. Math. Anal. Appl., 55, 61(1976).

157. Maslov and Zaikov, in Developments in Polymer Stabilisation-8, Ed. G. Scott, Applied Science Publisher, 1(1987).

158. R. M. Noyes and R. J. Field, Ann. Rev. Phys. Chem., Vol.25, 95(1974).

159. E. T. Denisov, in Developments in Polymer Stabilisation-5, Ed. G.

Scott, Applied Science Publisher, (1982).

160. S. Al-Malaika, E.O. Omikorede and G. Scott, Ploym. Commun., **27**, 173(1986).

161. J. T. Brownlie and K. U. Ingold, Can. J. Chem., **45**, 2427(1967)

162. S. Kavun, et al, Vysokomolek Soed., **9B**, 661(1967).

163. K. B. Chakraborty and G. Scott, Ploym. Deg. Stab., 5, 145(1983).

164. J. B. Adenyi, S. Al-Malaika and G. Scott, J. Appl. Polym. Sci., **32**, 6063(1986).

165. B. B. Cooray and G. Scott, in Developments in Polymer Stabilisation-2, Ed., G. Scott, Applied Science Publishers, 53(1980).

166. St epek Jiri, Additives for Plastics, Springer-Verlag, New York, 90(1983).

Appendix A Program to Simulate Binary Copolymerisation

```
C PROGRAM TO SIMULATE BINARY COPOLYMERISATION
C THE INPUTS FOR PROGRAM INCLUDE NUMBER OF MOLECULES TO
C SYNTHESIZE AND THE REACTIVITY RATIOS OF TWO MONOMERS AND
C THE MOLAR RATIO OF ONE MONOMER.  JUN-1987
        PARAMETER (NUM=100,NUPM=10000)
        INTEGER NOU(NUM)
        INTEGER COM(NUM,NUPM),UM1(NUM,NUPM),UM2(NUM,NUPM)
        INTEGER MAL1(NUM),MAL2(NUM),MUS1(NUM),MUS2(NUM)
        REAL RAT1(NUM,NUPM),RAT2(NUM,NUPM),PDIS(NUPM)
        REAL SEED
        EXTERNAL RAN
C READ INPUT
        TYPE *,'ENTER R12,R21,X1,NM'
        READ(5,*) R12,R21,X1,NM
        P12=1.0/(1.0+R12*X1/(1.0-X1))
        P21=1.0/(1.0+R21*(1.0-X1)/X1)
        P11=1.0-P12
        P22=1.0-P21
        Y=P21/(P12+P21)
        SEED=1.0
C START SIMULATION
        DO IM=1,NM
C GENERATE RANDOM NUMBER AS UNITS OF A MOLECULE 100<NOU<NUPM
          NOU(IM)=1000*INT(RAN(SEED))
          RN=RAN(SEED)
          IF (RN .GT. Y) THEN
            M=2
          ELSE
            M=1
          ENDIF
          I=1
          MHASH=M
10        IF (M .GT. 1) THEN
            P=P22
          ELSE
            P=P11
          ENDIF
          RN=RAN(SEED)
          IF (RN .GT. P) THEN
            M=2
          ELSE
            M=1
          ENDIF
          COM(IM,I)=M
          IF (I .LT. NOU(IM)) THEN
            I=I+1
            GOTO 10
          ENDIF
        ENDDO
C THE FOLLOWING UNIT COUNTS AMOUNTS OF DIFFERENT MONOMERS OF
C DIFFERENT LENGTH IN THE COPOLYMER.
        DO IM=1,NM
          LUM=1
          I=1
          MAL1(IM)=0
          MAL2(IM)=0
          DO WHILE (I .LE. NOU(IM))
            IF (COM(IM,I+1) .NE. COM(IM,I)) THEN
              IF (COM(IM,I) .EQ. 1) THEN
                UM1(IM,LUM)=UM1(IM,LUM)+1
                IF (LUM .GT. MAL1(IM)) MAL1(IM)=LUM
                LUM=0
              ELSE IF (COM(IM,I) .EQ. 2) THEN
                UM2(IM,LUM)=UM2(IM,LUM)+1
                IF (LUM .GT. MAL2(IM)) MAL2(IM)=LUM
                LUM=0
              ENDIF
            ENDIF
            I=I+1
            LUM=LUM+1
          ENDDO
          MUS1(IM)=0
          MUS2(IM)=0
          DO I=1,MAL1(IM)
            MUS1(IM)=MUS1(IM)+UM1(IM,I)
          ENDDO
          DO I=1,MAL2(IM)
            MUS2(IM)=MUS2(IM)+UM2(IM,I)
          ENDDO
        ENDDO
C CALCULATE THE RATIO OF DIFFERENT LENGTH MONOMER IN SAMPLE
        DO IM=1,NM
          DO I=1,MAL1(IM)
            RAT1(IM,I)=UM1(IM,I)/MUS1(IM)
          ENDDO
          DO I=1,MAL2(IM)
            RAT2(IM,I)=UM2(IM,I)/MUS2(IM)
          ENDDO
        ENDDO
C INITIALISE DISPLAY DEVICE
        CALL START
C DISPLAY THE RATIO OF EACH LENGTH MONOMER VS LENGTH
        DO I=1,NM
          NPT=MAL1(IM)
          DO J=1,NPT
            PDIS(J)=RAT1(I,J)
          ENDDO
          CALL DISPLAY(NPT,PDIS)
        ENDDO
        CALL FINISH
        END
```

```

C DISPLAY ROUTINE
        SUBROUTINE DISPLAY(NPT,Y)
        REAL Y(NPT)
        XSN=20.0
        YSN=20.0
        XLEN=200.0
        YLEN=150.0
        XBEG=1.0
        XEND=REAL(NPT)
        YBEG=0.0
        YEND=1.0
        XF=XLEN/(XEND-XBEG)
        XC=XSN-XF*XBEG
        YLENTH=YEND-YBEG
        YF=YLEN/(YEND-YBEG)
        YC=YSN-YF*YBEG
C DRAW COORDINATE FRAME
        CALL CORDNT(XSN,YSN,XLEN,YLEN,XBEG,XEND,YBEG,YEND)
        CALL LINCOL(6)
        GX=XBEG*XF+XC
        GY=YBEG*YF+YC
        CALL MOVTO2(GX,GY)
        DO I=1,NPT
```

```
          GX=XF*REAL(I)+XC
          GY=Y(I)*YF+YC
          CALL MOVTO2(GX,GY)
          CALL SYMBOL(8)
        ENDDO
        RETURN
        END

C DISPLAY THE CHAIN STRUCTURE OF POLYMER IN THE 2D
        SUBROUTINE DISMOL(NPT,UM)
        INTEGER UM(NPT)
        DOUBLE PRECISION G05CAF,SEED
        CALL HSVDEF(4,180.0,0.8,1.0)
        CALL HSVDEF(6,290.0,0.7,1.0)
        XMAX=200.0
        XMIN=40.0
        YMAX=160.0
        YMIN=20.0
        R=5.0
        PI=3.141596
        GX=120.0
        GY=90.0
C THIS UNITS DRAWS LINE SEGMENTS IN A RANDOM MANNER
C BUT WITH THE ASSIGNMENT. COLOUR ACCORDING TO WHAT
C MONOMER IT REPRESENTS. RANDOM GENERATOR IS USED
C TO ALTER THE ORIENTATION OF SEGMENT. THE WHOLE SIZE
C IS CONTROLLED BY CHECK IF THE NEXT MOVE IS OUT OF
C THE WINDOW AREA.
        CALL MOVTO2(GX,GY)
        CALL G05CBF(0)
        DO I=1,NPT
          ID=I
          J=0
100       SEED=DBLE(ID)
          BETA=2.0*PI*REAL(G05CAF(SEED))
          X=R*COS(BETA)
          Y=R*SIN(BETA)
          CALL POSPIC(GX,GY,Z)
          GX=GX+X
          GY=GY+Y
          J=J+1
          ID=J
          IF (GX .GE. XMAX .OR. GX .LE. XMIN) GOTO 100
          IF (GY .GE. YMAX .OR. GY .LE. YMIN) GOTO 100
          IF (UM(I) .EQ. 1) THEN
            CALL LINCOL(6)
          ELSE
            CALL LINCOL(4)
          ENDIF
          CALL LINBY2(X,Y)
        ENDDO
        RETURN
        END

C INITIALISE THE DISPLAY ROUTINE
        SUBROUTINE START
        CALL GINO
        CALL T4107
        CALL WINDO(2)
        CALL HSVDEF(0,0.0,0.0,0.0)
        RETURN
        END
```

```
C RELEASE DISPLAY DEVISE
      SUBROUTINE FINISH
      CALL GINEND
      RETURN
      END


      SUBROUTINE CORDNT(AXS,AYS,XLEN,YLEN,XBEG,XEND,YBEG,YEND)
      REAL*4 XBEG,XEND,YBEG,YEND,AXS,AYS,XLEN,YLEN
      CALL LINCOL(10)
      CALL AXIPOS(1,AXS,AYS,XLEN,1)
      CALL AXIPOS(1,AXS,AYS,YLEN,2)
      CALL AXISCA(3,9,XBEG,XEND,1)
      CALL AXISCA(3,9,YBEG,YEND,2)
      CALL AXIDRA(1,1,1)
      CALL AXIDRA(-1,-1,2)
      CALL MOVTO2(AXS,AYS+YLEN)
      CALL LINBY2(XLEN,0.0)
      CALL LINBY2(0.0,-YLEN)
      RETURN
      END
```

Appendix B Program for NO-PP Reaction System

```
C*********************************************************
C***  PROGRAM TO SOLVE THE SYSTEM CONSISTING OF         ***
C***    SIX ORDINARY DIFFERENTIAL EQUATIONS BY          ***
C***    THE FOURTH ORDER RUNGE-KUTTA METHODS            ***
C***      WITH SPECIAL APPLICATION TO                   ***
C***         CHEMICAL KINATICS                          ***
C***                                                    ***
C***                           JAN-1988                 ***
C*** NOTE: This program calls  GINO for graphics and    ***
C***       screen management routines(SMG$) for the     ***
C***       arrangement of text on the screen.           ***
C***       For given problem, it is necessary to supply ***
C***       the equations in subroutine DF. The maximum  ***
C***       is now set to 10, but it can easily be       ***
C***       readjusted to larger number.                 ***
C*********************************************************
C PARAMETER VALUES
C NCOEF...... NUMBER OF COEFFICIENTS USED IN RUNGE-KUTTA
C MNRC ...... MAXIMUM NUMBER OF REACTIONS
C MNVR ...... MAXIMUM NUMBER OF DEs IN SYSTEM
C MINPT ...... MAXIMUM NUMBER OF INPUT VALUES
C OSCPAR ...... OSCILLATION PARAMETER
C ADCON ...... CONSTANT CONCENTRATION VALUES
C KV ...... RATE CONSTANT FOR EACH REACTION IN SYSTEM
C SV ...... STARTING VALUE USED IN RUNGE-KUTTA
C PV ...... VALUE PASSED TO THE INTEGRATION FUNCTION
C X,Y ...... VARIABLE OF INTEREST FOR DISPLAY
C PROMP ...... PROMPT TEXT FOR INPUT
        PROGRAM NITROMOD
        PARAMETER(N=50000,NCOEF=4,MNVR=10,MINPT=50,
     $            NPROMP=10,NADV=8,NOSC=3,NADC=4,MNRC=20)
        REAL X(N),Y(N),GT(N),SV(MNVR),PV(MNVR)
        REAL CV(NCOEF,MNVR)
        REAL OSCPAR(NOSC),ADCON(NADC)
        REAL KV(MNRC),INPT(MINPT)
        INTEGER DIS,PAS,KBD,VROW,VCOL
        CHARACTER*1 IRECT
        CHARACTER*4 ANS
        CHARACTER*25 PROMP(NPROMP)
        LOGICAL MODE
        DATA PROMP/'Rate const. of reaction','Initial conc. of comp',
     $           'Diffusion rate','Osc. freq.','Amplitude',
     $           'Concentration of C1','Concentration of C2',
     $           'Concentration of C3','Reaction time(min)',
     $           'Step size'/
        COMMON /GVAR/OSCPAR
        COMMON /RTCONS/KV
        COMMON /VINIT/ADCON
        COMMON /DISID/ DIS,PAS
        INCLUDE '($SMGDEF)'
        MODE=.FALSE.
C INITIALISE THE GRAPHICS DISPLAY DEVICE
        CALL START
        CALL WINDOW(2)
        CALL CHAMOD
C CREATE SCREEN MANAGEMENT DISPLAY
        CALL SMG$CREATE_VIRTUAL_DISPLAY(23,50,DIS)
        CALL SMG$CREATE_VIRTUAL_KEYBOARD(KBD)
        CALL SMG$CREATE_PASTEBOARD(PAS)
        CALL SMG$PASTE_VIRTUAL_DISPLAY(DIS,PAS,1,1)
        IBPI=30
10      VROW=1
        VCOL=20
C INPUT THE NECESSARY PARAMETERS AND INTIAL CONDITIONS
        CALL SMG$PUT_CHARS(DIS,'Please enter the data',VROW,VCOL)
        VROW=3
        VCOL=5
        CALL SMG$PUT_CHARS(DIS,'How many reactions?',VROW,VCOL)
        CALL SMG$SET_CURSOR_ABS(DIS,VROW,VCOL+IBPI)
        READ(5,*) NRC
        VROW=VROW+1
        CALL SMG$PUT_CHARS(DIS,'How many equations?',VROW,VCOL)
        CALL SMG$SET_CURSOR_ABS(DIS,VROW,VCOL+IBPI)
        READ(5,*) NVR
        NBASE=NRC+NVR
        NINPT=NBASE+NADV
        DO I=1,NINPT
          VROW=VROW+1
          IF (VROW .EQ. 22) THEN
            CALL SMG$DELETE_LINE(DIS,3,20)
            VROW=3
          ENDIF
          IDPR=I-NBASE+2
          IF (I .LE. NRC) IDPR=1
          IF (I .GT. NRC .AND. I .LE. NBASE) IDPR=2
          CALL SMG$PUT_CHARS(DIS,PROMP(IDPR),VROW,VCOL)
          IF (I .LE. NBASE) THEN
            IF (I .LE. NRC) THEN
              WRITE(UNIT=IRECT,FMT='(I1)') I
              IPS=23
            ELSE
              WRITE(UNIT=IRECT,FMT='(I1)') I-NRC
              IPS=21
            ENDIF
            CALL SMG$PUT_CHARS(DIS,IRECT,VROW,VCOL+IPS)
          ENDIF
          CALL SMG$SET_CURSOR_ABS(DIS,VROW,VCOL+IBPI)
          READ(5,*) INPT(I)
        ENDDO
C ASSIGN VALUES FOR COMMON BLOCK
        CORR=210.0/30.0
        NT=INT(CORR*INPT(NINPT-1))
        DH=INPT(NINPT)
        DO I=1,NRC
          KV(I)=INPT(I)
        ENDDO
        DO I=1,NOSC
          OSCPAR(I)=INPT(NBASE+I)
          ADCON(I+1)=INPT(NBASE+NOSC+I)
        ENDDO
        ADCON(1)=INPT(NRC+3)
        CALL SMG$PUT_CHARS(DIS,'Press spacebar to continue',23,10)
        CALL SMG$READ_KEYSTROKE(KBD,TMC)
        CALL SMG$DELETE_LINE(DIS,1,24)
C START THE SOLUTION OF THE SYSTEM BY RUNGE-KUTTA METHOD
100     DO I=1,NVR
          SV(I)=INPT(NRC+I)
          PV(I)=SV(I)
        ENDDO
        X(1)=SV(1)
        Y(1)=SV(2)
        T=0.0
        DO I=2,NT
          T=T+DH
          DO J=1,NCOEF
            IF (J .EQ. 1) THEN
```

```
                  TV=T
                  DO K=1,NVR
                    PV(K)=SV(K)
                  ENDDO
                ELSE IF(J .EQ. NCOEF) THEN
                  TV=T+DH
                  DO K=1,NVR
                    PV(K)=SV(K)+CV(J,K)
                  ENDDO
                ELSE
                  TV=T+DH/2.0
                  DO K=1,NVR
                    PV(K)=SV(K)+CV(J,K)/2.0
                  ENDDO
                ENDIF
                DO K=1,NVR
                  CV(J,K)=DH*DF(TV,K,PV(1),PV(2),PV(3),PV(4),PV(5),
     $                          PV(6),PV(7),PV(8),PV(9),PV(10))
                ENDDO
              ENDDO
              DO J=1,NVR
                SV(J)=SV(J)+(CV(1,J)+2.0*CV(2,J)+2.0*CV(3,J)+CV(4,J))/6.0
              ENDDO
              X(I)=SV(1)
              Y(I)=SV(2)
            ENDDO
C CALL DISPLAY ROUTINE
        CALL PLOTGR(NT,X,Y,GT,DH,MODE)
        CALL CHAMOD
C DISPLAY THE PARAMETERS USED SO FAR
        CALL SMG$READ_KEYSTROKE(KBD,TMC)
        CALL SMG$PUT_CHARS(DIS,'CONTINUE?',23,30)
        CALL SMG$READ_STRING(KBD,ANS,,1,,,,,DIS)
        CALL SMG$DELETE_LINE(DIS,1,23)
        CALL CLEAN
        CALL CHAMOD
        IF(ANS .NE. 'N') GOTO 10
        CALL FINISH
        END

C*******************************************************
C ROUTINE TO DISPLAY THE SPECTRA AND PHASE DIAGRAM  ****
        SUBROUTINE PLOTGR(N,X,Y,GT,DH,MODE)
C*******************************************************
        PARAMETER(MAXNP=5000)
        COMMON /DISID/DIS,PAS
        INTEGER DIS,PAS,TMC
        REAL X(N),Y(N),GX(MAXNP),GY(MAXNP),GT(N)
        LOGICAL MODE
        CALL CHAMOD
        CALL SMG$CREATE_VIRTUAL_KEYBOARD(KBD)
        CALL CLEAN
        CALL MAXMIN(N,X,Y,XM,XN,YM,YN)
        GM=AMAX1(XM,YM)
        GN=AMIN1(XN,YN)
C CORRELATION OF TIME SCALE
        CORR=30.0/21.00
        TMIN=GT(1)*CORR
        TMAX=GT(N)*CORR
C SET VIEWPORT ON DISPLAY DEVICE
        SXN=25.0
        SXM=225.0
        SYN=15.0
        SYM=165.0
        XLEN=SXM-SXN
```

```
      YLEN=SYM-SYN
      DT=XLEN/REAL(N)
      YK=YLEN/(GM-GN)
      YC=SYM-YK*GM
      IF (MODE) THEN
        TK=XLEN/(TMAX-TMIN)
        TC=SXM-TK*TMAX
      ENDIF
C MAPPING THE DISPLAY
      DO I=1,N
        GX(I)=X(I)*YK+YC
        GY(I)=Y(I)*YK+YC
        GT(I)=GT(I)*TK+TC
      ENDDO
      IF (.NOT. MODE) THEN
        TMIN=0.0*CORR
        TMAX=CORR*DH*REAL(N)
        DO I=1,N
          GT(I)=SXN+REAL(I-1)*DT
        ENDDO
      ENDIF
C DRAW COORDINATES AND ILLUATRTIVE TEXT
      CALL CORDNT(SXN,SYN,XLEN,YLEN,TMIN,TMAX,GN,GM)
      CALL MOVTO2(SXN,SYM)
      CALL LINBY2(XLEN,0.0)
      CALL LINBY2(0.0,-YLEN)
      CALL CHAENQ(IT,WID,HEIT,NI,AI,AN)
      CALL MOVTO2(180.0,155.0)
      CALL LINCOL(6)
      CALL LINBY2(10.0,0.0)
      CALL CHASTR(' X*.')
      CALL MOVTO2(180.0,150.0)
      CALL LINCOL(5)
      CALL LINBY2(10.0,0.0)
      CALL CHASTR(' Y*.')
      CALL LINCOL(10)
      CALL MOVTO2(110.0,5.0)
      CALL CHASTR(' Time(min)*.')
      CALL CHASWI(1)
      CALL CHAANG(90.0)
      CALL MOVTO2(5.0,80.0)
      CALL CHASTR('Concentration*.')
      CALL CHAANG(0.0)
      CALL CHAMOD
      NSFT=0
      CALL SMG$DELETE_LINE(DIS,1,23)
      XBK=GX(1)
      YBK=GY(1)
      TLAS=SXN
      DO I=2,N
        J=I+NSFT
        PX=GT(I)
        PY=GX(J)
        CALL LINCOL(6)
        CALL MOVTO2(TLAS,XBK)
        CALL LINTO2(PX,PY)
        XBK=PY
        PY=GY(J)
        CALL LINCOL(5)
        CALL MOVTO2(TLAS,YBK)
        CALL LINTO2(PX,PY)
        TLAS=PX
        YBK=PY
      ENDDO
      CALL CHAMOD
      CALL SMG$READ_KEYSTROKE(KBD,TMC)
```

```
        CALL CLEAN
C DRAW THE PHASE DIAGRAM OF THE PREVIOURS SPECTRA
        CALL CORDNT(SXN,SYN,XLEN,YLEN,YN,YM,XN,XM)
        CALL MOVTO2(SXN,SYM)
        CALL LINBY2(XLEN,0.0)
        CALL LINBY2(0.0,-YLEN)
        CALL MOVTO2(110.0,SYM+5.0)
        CALL LINCOL(2)
        CALL CHASTR('Phase Diagram*.')
        CALL LINCOL(10)
        CALL MOVTO2(110.0,5.0)
        CALL CHASTR('Concentration of Y*.')
        CALL CHAANG(90.0)
        CALL MOVTO2(5.0,70.0)
        CALL CHASTR('Concentration of X*.')
        CALL CHAANG(0.0)
        CALL CHAMOD
        CALL SMG$DELETE_LINE(DIS,1,23)
100     XK=(SXM-SXN)/(YM-YN)
        XC=SXM-XK*YM
        YK=(SYM-SYN)/(XM-XN)
        YC=SYM-YK*XM
        DO I=1,N
          X(I)=X(I)*YK+YC
          Y(I)=Y(I)*XK+XC
        ENDDO
        CALL LINCOL(7)
        CALL MOVTO2(Y(1),X(1))
        DO I=2,N
          CALL LINTO2(Y(I),X(I))
        ENDDO
        CALL CHAMOD
        MODE=.FALSE.
        RETURN
        END

C**********************************************************
C**     SUBROUTINE OF INITAILISE DEVICE                 ***
        SUBROUTINE START
C**********************************************************
        CALL GINO
        CALL T4107
        CALL HSVDEF(0,0.0,0.0,0.0)
        RETURN
        END

C***********************************************************
C***    SUBROUTINE OF RELEASING DEVICE                   ***
        SUBROUTINE FINISH
C***********************************************************
        CALL CLEAN
        CALL GINEND
        RETURN
        END

C***********************************************************
C***    CLEAN THE SCREEN                                 ***
        SUBROUTINE CLEAN
C***********************************************************
        CALL LINCOL(0)
        CALL RFILL(0,0,0.0,0.0,240.0,180.0)
        RETURN
        END
```

```

C***********************************************
C***     SERCH THE MAXIMUM AND MINIMUM VALUES  ***
        SUBROUTINE MAXMIN(NP,X,Y,XM,XMN,YM,YMN)
C***********************************************
        REAL*4 X(NP),Y(NP)
        REAL*4 XM,XMN,YM,YMN
        XM=X(1)
        XMN=X(1)
        YM=Y(1)
        YMN=Y(1)
        DO 100 I=1,NP
        IF(X(I) .LT. XMN) XMN=X(I)
        IF(X(I) .GT. XM) XM=X(I)
        IF(Y(I) .LT. YMN) YMN=Y(I)
        IF(Y(I) .GT. YM) YM=Y(I)
100     CONTINUE
        RETURN
        END

C****************************************************************
C       DRAW THE COORDINATES FOR DISPLAY                      ***
        SUBROUTINE CORDNT(AXS,AYS,XLEN,YLEN,XBEG,XEND,YBEG,YEND)
C****************************************************************
        REAL*4 XBEG,XEND,YBEG,YEND,AXS,AYS,XLEN,YLEN
        CALL LINCOL(10)
        CALL AXIPOS(1,AXS,AYS,XLEN,1)
        CALL AXIPOS(1,AXS,AYS,YLEN,2)
        CALL AXISCA(3,9,XBEG,XEND,1)
        CALL AXISCA(3,9,YBEG,YEND,2)
        CALL AXIDRA(1,1,1)
        CALL AXIDRA(-1,-1,2)
        CALL MOVTO2(AXS,AYS+YLEN)
        CALL LINBY2(XLEN,0.0)
        CALL LINBY2(0.0,-YLEN)
        RETURN
        END

C*************************************************************
C THE DIFFERENTIAL EQUATIONS FOR SOLUTION                  ***
        REAL FUNCTION DF(T,ID,X1,X2,X3,X4,X5,X6,X7,X8,X9,X10)
C*************************************************************
        PARAMETER (MNRC=20,NOSC=3,NADC=4)
        COMMON /GVAR/OSCPAR
        COMMON /RTCONS/ K
        COMMON /VINIT/ADCON
        REAL OSCPAR(NOSC),ADCON(NADC)
        REAL K(MNRC)
        DFC=OSCPAR(1)
        FC=OSCPAR(2)
        CEF=OSCPAR(3)
        CO=ADCON(1)
        C1=ADCON(2)
        C2=ADCON(3)
        C3=ADCON(4)
        CK=K(1)*C2
        DIF=DFC-CK
        GOTO (100,101,102,103,104,105,106,107,108,109),ID
100     DF=K(3)*X2*X4+K(4)*X2*X5+K(5)*X2*C3-K(2)*X1*C2-
     $     K(6)*X1*C1-K(7)*X1*X3
        GOTO 200
101     DF=K(2)*X1*C2+K(6)*X1*C1-K(3)*X2*X4-K(4)*X2*X5-K(5)*X2*C3
        GOTO 200
```

```
102     DF=CO*(DIF*(1.0+CEF*COS(FC*T))-CEF*FC*SIN(FC*T))*EXP(DIF*T)
        GOTO 200
103     DF=K(1)*C2*X3-K(3)*X2*X4
        GOTO 200
104     DF=K(3)*X2*X4-K(4)*X2*X5
        GOTO 200
105     DF=NULL
        GOTO 200
106     DF=NULL
        GOTO 200
107     DF=NULL
        GOTO 200
108     DF=NULL
        GOTO 200
109     DF=NULL
200     END
```

Appendix C The Design of A Small Database

The basic intention was to form an application which will store chemical information for polymer additives and allow easy access of stored information.

Necessity:

The accumulation of chemical and practical data about polymer stabilisers have been piled up quickly. It is desired to have a fast, efficient, accurate way to access the data of the known or widely used stabilisers both for research and application purposes. Although there have been various database environment or even chemical information system, unavailability in our particular site or speciality makes it seem proper to write our own applications..

Requirements:

The storage and access of both text and graphical data, especially the chemical structure data, such as chemical structure, spectroscopic data, etc.

Possibility:

The program skill and the facility should allow such programming be done in a fairly short period of time.

Strategy:

Identify the basic processes involved in such a system. Then design the program. Some of the basic aspects are discussed in the following sections. They are:

(1) design of an interactive drawing procedure of chemical structures, and a procedure of storing spectroscopic data.

(2) data storage file design

(3) processing operations

(4) flow of the program

(5) general layout

(6) implementation

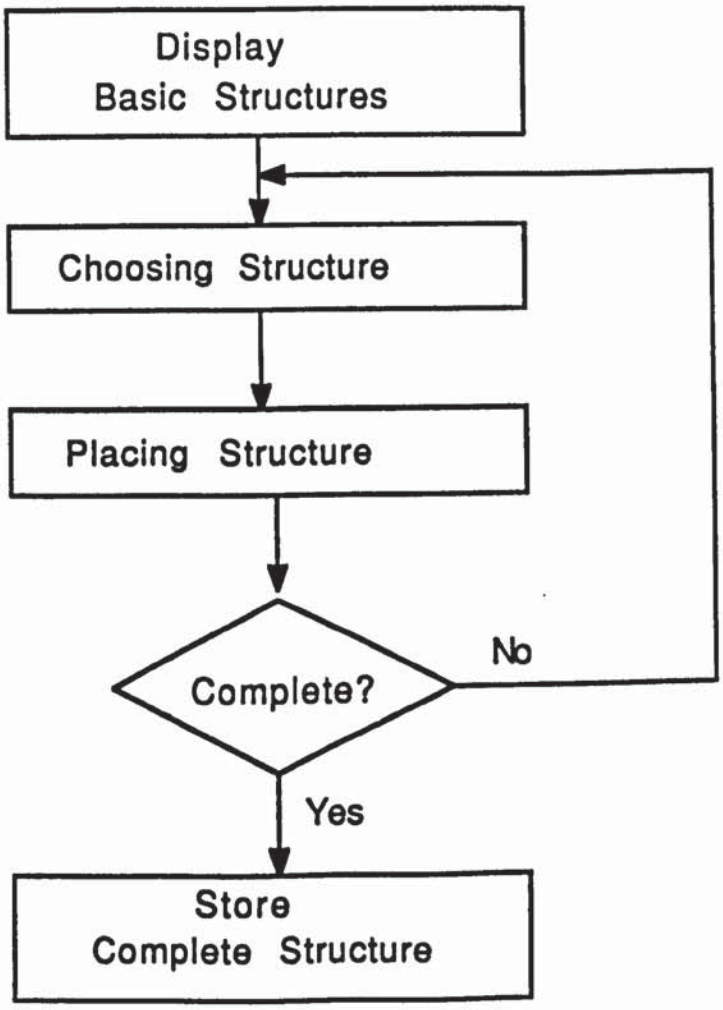
1. Interactive Drawing of Chemical Structures and the Storing of Spectroscopic Data

1.1 Chemical Structure Drawing

Basic Idea:

(1) display a set of basic structures or structural entities used in the polymer stabilisation chemistry (it can easily be extended to general chemistry if necessary);

(2) user pick up structures to build his own structure: this requires a mechanism to locate the choice and positioning of the structure chosen; free motion of chosen structure; the deletion of the chosen structure; the restoring mechanism in case of unsatisfaction; the storing of the structures built. Many ways have been tried. It finally was found to characterise each basic structure by the coordinates of its center. In this way, the anchoring and the redisplay of a structure can be done accurately. The basic flow of this procedure is as follows:



(3) picking structure: when a search cursor indicate that a structure is chosen, the identity of it is known by the following way

```
DO I=1, NR
  DO J=1, NSPR
     IF (xl(I,J)<XC<xr(I,J) || yb(I,J)<YC<yt(I,J)) then
       ID_STR=(I-1)*NSPR + J
        out of loop
     ENDIF
```

```
        ENDDO
    ENDDO
```

where, NSPR is number of basic structures per row, NR is number of rows of displaying basic structures. xl, xr,yb,yt define the rectangle in which the basic structure dwell. XC and YC are the coordinates of picking cursor.

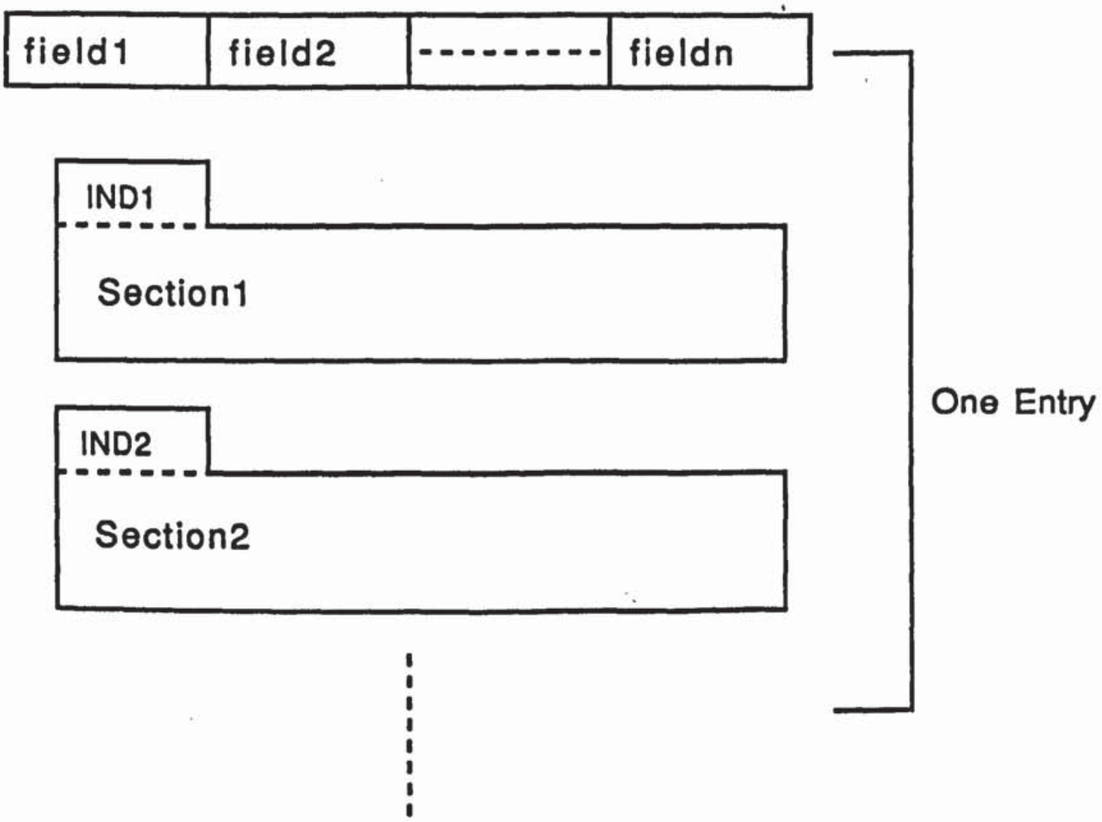
1.2 Storing the Spectroscopic Data

The spectrum are viewed as curve. Therefore, it can be stored easily by the coordinates of each point on the curve. Two ways may be used to input the curve. The efficient way is to use graphical input facilities available on the graphics terminal. Another way is to input the peak table.

2. Data File

The way of the data stored should allow the easy access of the content under each entry by different routes. For example, an entry may be accessed by either its chemical name or commercial names. It should also allow the relational access of the data. For instance, users may enquire by manufacturer's name, major usage, etc. It is possible to achieve all this by simply having the data sorted in different way in many different data files. But, the immediate problem is the storage capacity and the efficiency concern. So, if the data can be stored in a single file or a small number of files, and the above performances may be achieved, it is very useful. In our case, because the text data and graphical data are involved, and it is required that program should allow the updating of any section of the data under each entry, the data file holding all entries and their contents must be carefully designed.

The chosen structure of the data for each entry is as following:



field1~fieldn are the identities of this block of data, which can be used as access routes to it, section1~section--- are detailed information stored for each entry, they have indicators IND1,etc, which are used to specify what are stored in each section and how much, etc.

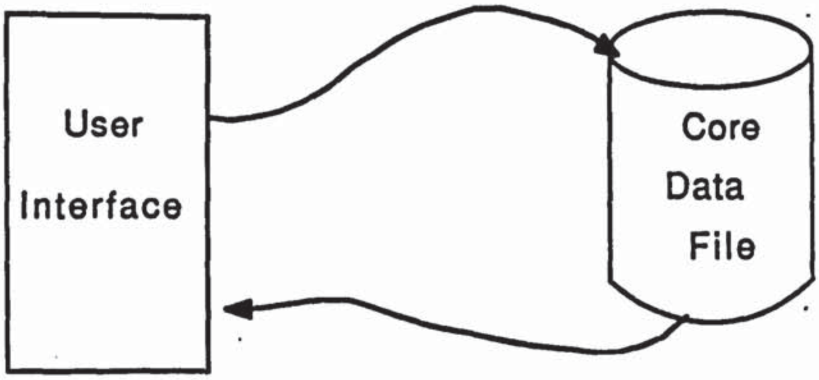
3. Processing Operations

Some simple processing operations like the comparision of the spectroscopic data of different samples, the extraction of the stored information in the database including graphical data and text to form a format which could be useful in the writing of report and the likes, list of products of manufacturer, list of products of a specified year, list of products according to function, etc. This small database also provides facility to set up private data in the system, so only the user who supplies the data has access to the stored data. Therefore, although the whole database is public, but if it is used as a research tool in a group, it also provides facility to set up privately owned data, of cause, they can be made public on specification.

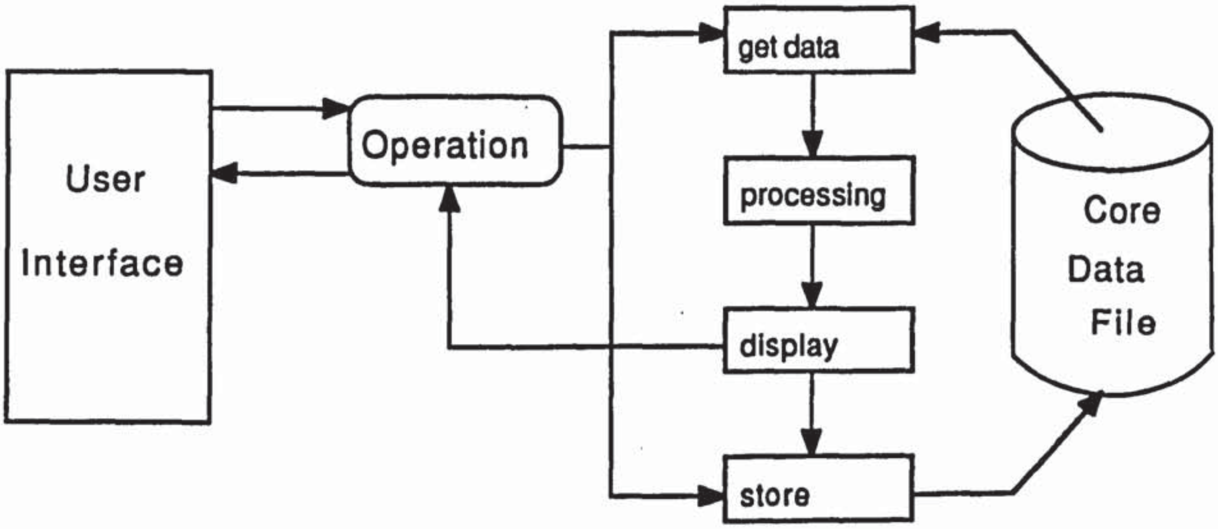
It is planned that when this database has taken enough information, it should also incorporate some expert system functions, like the identification of a not very complicated compound, the recommendation of use of certain compounds under certain conditions, the advice of processing formulation, etc.

4. Flow of the Program

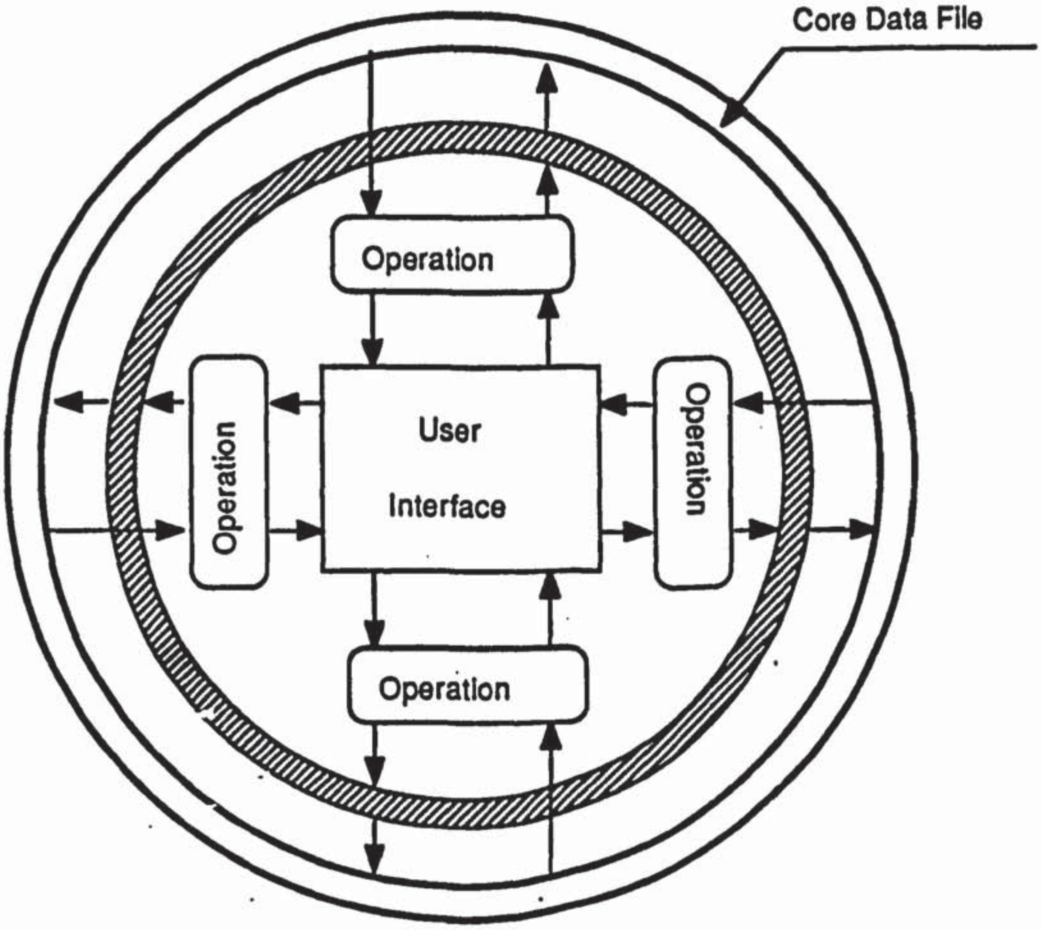
The major flow of the program is between the core data file which holds all entries and the information under them, and the user interface:



But the routes can be various. That is, the actual processes are the subroutines which does the operation chosen. A typical process looks like:



Therefore, the overall flow would be as following:



The shadowed layer where each operation has to go through when it adds information to the core data file is a process which conforms the information going to the core data file into a suitable format for the reason of storage.

5. General Layout

The general layout is done in graphical form. This makes the appearance better and easy to manage. The communication is by two ways, one is using mouse to indicate the choice, especially the menu items; another way is typing simple commands or response through keyboard. The long text information will remain text form, and can be managed by the screen management routines provided by the system services, on VAX computers, these are SMG$ routines.

6. Implementation

The program was written in FORTRAN on VAX machines, graphical routines can either be from GINO package, or from my own graphical primitives for Tektronix terminals. It is planned to write the same program in C, which will be tested on Apollo work station, and graphical routines will be based on international standard GKS. The implementation of the chemical structure drawing program on Macintosh is now in progress.

Note: Enclosed are some sample plots from the program and the listing of it.

Appendix D Program of the Database

```
C PROGRAM TO BUILD A DATABASE OF CHEMICAL INFO
        PROGRAM AIS
C ROUTINE FOR STORING THE SPECTRUM DATA
        PARAMETER (MAXNM=10,NCRTV=2)
        INTEGER SMGID,PBD,KBD,VROW,VCOL
        REAL XMS(MAXNM),YMS(MAXNM)
        CHARACTER*20 ANSCRT(NCRTV),ANSDB
        CHARACTER*20 PRTEXT(MAXNM)
        LOGICAL AISINF,PROMP
        COMMON /PSCREN/ ORX,ORY,XALEN,YALEN
        COMMON /VIEWPT/ SVXLEN,SVYLEN,SVXST,SVYST,
     $                BVXLEN,BVYLEN,BVXST,BVYST
        COMMON /DSHVMW/ DSBFH,DSBFV,WIDEMF,FINT,DFHEIT
        COMMON /VIEWSIZE/ ANSCRT,ANSDB
        COMMON /SMGPAR/ SMGID,PBD,KBD,VROW,VCOL
        COMMON /LOGVAR/ AISINF
        COMMON /DFTCHAR/ DCHWD,DCHHT
        DATA (PRTEXT(I),I=1,4)/'Large or Small View',' ','Start',
     $                     'NAME,COMM,MANF,ASTC'/
        DATA ANSCRT/'LARGE','LARGE VIEW'/
        INCLUDE '($SMGDEF)'
C BASIC PARAMETERS OF SCREEN
        ORX=0.0
        ORY=0.0
        XALEN=238.0
        YALEN=178.0
C SET AIS INFORMATION ON
        AISINF=.TRUE.
C INITIALISE THE DEVICE
        CALL START
        CALL WINDOW(2)
C DEFAULT CHARACTER SIZE
        CALL CHAENQ(ITO,DCHWD,DCHHT,NIO,ANO,AIO)
C CREATE SCREEN MANGAMENT DISPLAY
        CALL CHAMOD
        NROW=24
        NCOL=80
        CALL SMG$CREATE_VIRTUAL_DISPLAY(NROW,NCOL,SMGID)
        CALL SMG$CREATE_VIRTUAL_KEYBOARD(KBD)
        CALL SMG$CREATE_PASTEBOARD(PBD)
        CALL SMG$PASTE_VIRTUAL_DISPLAY(SMGID,PBD,1,1)
C CALL A STARING PROCEDURE
        CALL BEGIN
C DISPLAY THE MAIN MENUE AND INTRODUCTION
1000    PROMP=.TRUE.
        CALL CLEAR(BVXST,BVYST,BVXST+BVXLEN,BVYST+BVYLEN)
        CALL MAINPAD
        IF (.NOT. AISINF) CALL GENDIS
C DISPALY INDICATOR FOR ALLOCATION OF OPERATION
        CALL MENUID(10,IDM)
100     CALL MENUCL
        GOTO (200,201,202,203,204,205,206,206,206,206),IDM
200     CALL SETVW
        GOTO 1000
201     CALL LISTENTR
        GOTO 1000
202     CALL ADDENTRY
        GOTO 1000
203     CALL DELENTRY
        GOTO 1000
```

```
204     CALL SEARCH
        GOTO 1000
205     CALL UTILTY
        GOTO 1000
206     CALL CHAMOD
        CALL FINISH
        END
```

```

C STARTING PROCEDURE
        SUBROUTINE BEGIN
        COMMON /PSCREN/ ORX,ORY,XALEN,YALEN
        XC=XALEN/2.0
        YC=YALEN/2.0
        REML=170.0
        RECL=REML
        IDSEG=1
        DO WHILE(RECL .GT. 0.0)
          RCX=XC-RECL/2.0
          RCY=YC-RECL/2.0
          ICOL=MOD(ICOL,8)+1
          CALL LINCOL(ICOL)
          CALL MOVTO2(RCX,RCY)
          CALL LINBY2(0.0,RECL)
          CALL LINBY2(RECL,0.0)
          CALL LINBY2(0.0,-RECL)
          CALL LINBY2(-RECL,0.0)
          RECL=RECL-2.0
        ENDDO
        ICOL=0
        CALL LINCOL(ICOL)
        DO WHILE(RECL .LE. REML)
          RCX=XC-RECL/2.0
          RCY=YC-RECL/2.0
          CALL MOVTO2(RCX,RCY)
          CALL LINBY2(0.0,RECL)
          CALL LINBY2(RECL,0.0)
          CALL LINBY2(0.0,-RECL)
          CALL LINBY2(-RECL,0.0)
          RECL=RECL+2.0
        ENDDO
        RETURN
        END
```

```

C DISPLAY AN INDICATOR FOR ALLOCATION OF MENU ITEM
        SUBROUTINE MENUID(NMENIT,IDM)
        PARAMETER (MAXNEM=10)
        REAL YMS(MAXNEM)
        COMMON /GFEVEN/KEY,IMPKEY,IMPDAT,NSEG,XIND,YIND,
     $                NARGS,ARGS(80)
        COMMON /VIEWPT/ SVXLEN,SVYLEN,SVXST,SVYST,
     $                 BVXLEN,BVYLEN,BVXST,BVYST
        COMMON /DSHVMW/ DSBFH,DSBFV,WIDEMF,FINT,DFHEIT
100     IND=1000
        XSIND=BVXST+WIDEMF-DSBFH
        YSIND=BVYST+BVYLEN-DSBFH
        CALL INDICATOR(IND,XSIND,YSIND)
        CALL DRAG(IND)
        CALL PICDEL(IND)
        YSMF=BVYST+BVYLEN
        DO IM=1,NMENIT
          YMS(IM)=YSMF
          YSMF=YSMF-FINT
        ENDDO
        DO IM=1,NMENIT
```

```
          IF (YIND .LT. YMS(IM) .AND.
     $        YIND .GT. YMS(IM+1) .AND.
     $        XIND .GT. BVXST .AND.
     $        XIND .LT. (BVXST+WIDEMF)) THEN
            IDM=IM
            RETURN
          ENDIF
        ENDDO
        GOTO 100
        END

C CLEAR A SPECIFIED AREA
        SUBROUTINE CLEAR(RXS,RYS,RXE,RYE)
        CALL LINCOL(0)
        CALL RFILL(0,0,RXS,RYS,RXE,RYE)
        RETURN
        END

C CLEAR THE MENU AREA
        SUBROUTINE MENUCL
        COMMON /DSHVMW/ DSBFH,DSBFV,WIDEM,FINT,DFHEIT
        COMMON /VIEWPT/ SVXLEN,SVYLEN,SVXST,SVYST,
     $                        BVXLEN,BVYLEN,BVXST,BVYST
        CALL CLEAR(BVXST,SVYST,BVXST+WIDEM,SVYST+SVYLEN)
        RETURN
        END

C SET VIEWPORT SIZE
        SUBROUTINE SETVW
        PARAMETER(NDM=2,NCRT=2)
        CHARACTER*15 DMENT(NDM)
        CHARACTER*20 ASCRT(NCRT),VPTCS
        COMMON /VIEWPT/ SVXLEN,SVYLEN,SVXST,SVYST,
     $                        BVXLEN,BVYLEN,BVXST,BVYST
        COMMON /VIEWSIZE/ ASCRT,VPTCS
        COMMON /DSHVMW/ DSBFV,DSBFH,WIDEMF,FINT,XSMF,YSMF
        DATA DMENT/'Large View','Small View'/
        HEITMF=NDM*FINT
        ICF=8
        ICT=6
        CALL MENUDR(XSMF,YSMF,WIDEMF,HEITMF,FINT,ICF,
     $                    NDM,DMENT,ICT)
        CALL MENUID(NDM,IDM)
        CALL MENUCL
        CALL STR$UPCASE(VPTCS,DMENT(IDM))
        RETURN
        END

C CLEAR DIALOG TEXT
        SUBROUTINE DIATCL
        INTEGER SMGID,PBD,KBD,VROW,VCOL
        COMMON /SMGPAR/ SMGID,PBD,KBD,VROW,VCOL
        CALL CHAMOD
        CALL SMG$DELETE_LINE(SMGID,VROW,1)
        RETURN
        END

C DISPLAY PROMPT IN DIALOG BOX
        SUBROUTINE PROMPT(TEXT,ANSER,PROMP)
        INTEGER SMGID,PBD,KBD,VROW,VCOL
        CHARACTER*20 ANSER
        CHARACTER*20 TEXT
        LOGICAL PROMP
        COMMON /DSHVMW/ DSBFH,DSBFV,WIDEM,FINT
        COMMON /DIABOX/ DFXS,DFYS,WIDEPR,DFHEIT
        COMMON /SMGPAR/ SMGID,PBD,KBD,VROW,VCOL
        COMMON /DFTCHAR/ DCHWD,DCHHT
        CALL CHAENQ(ITO,WDO,HTO,NIO,AIO,ANO)
        CHWD=DCHWD
        CHHT=DCHHT
        CALL CHASIZ(CHWD,CHHT)
        WIDEPR=80.0
        BXS=DFXS+DSBFH
        BYS=DFYS+DSBFV
        BXE=DFXS+WIDEPR-DSBFH
        BYE=DFYS+DFHEIT-DSBFV
        IF (PROMP) THEN
          CALL LINCOL(6)
          CALL MOVTO2(DFXS+WIDEPR,DFYS)
          CALL LINBY2(0.0,DFHEIT)
        ENDIF
        CALL LINCOL(4)
        CALL MOVTO2(BXS,BYS+DSBFV)
        CALL CHASTR(TEXT)
        CALL CHAMOD
        VROW=24
        VCOL=31
        CALL SMG$SET_CURSOR_ABS(SMGID,VROW,VCOL)
        READ(5,111) ANSER
        CALL DIATCL
        CALL CLEAR(BXS,BYS,BXE,BYE)
111     FORMAT(A)
        CALL CHASIZ(WDO,HTO)
        RETURN
        END

C*********************************************************
C***      ROUTINE TO SCAN IR  SPECTRUM                ****
        SUBROUTINE IRSCAN(NP,XRC,YRC)
C*********************************************************
        PARAMETER (MAXP=50000)
        REAL XRC(MAXP),YRC(MAXP),RCX(MAXP),RCY(MAXP)
        CHARACTER*20 ASCRT(2),VPTCS
        COMMON /VIEWSIZE/ ASCRT,VPTCS
        COMMON /PSPCHART/ XSSPC,XESPC,YSSPC,YESPC
        COMMON /BCHTREGN/ BVCXS,BVCXE,BVCYS,BVCYE
        COMMON /GFEVEN/KEY,IMPKEY,IMPDAT,NSEG,XPIC,YPIC,
     4                         NARGS,ARGS(80)
        CALL CURSTR('SER')
10      CALL IRCHART
        IND=1000
        XSIND=(XSSPC+XESPC)/2.0
        YSIND=(YSSPC+YESPC)/2.0
        CALL INDICATOR(IND,XSIND,YSIND)
        IP=1
100     CALL MOVTO2(XPIC,YPIC)
500     CALL DRAG(IND)
        GOTO(100,1000,10,200),KEY
200     RCX(IP)=XPIC
        RCY(IP)=YPIC
        CALL LINCOL(8)
        CALL MOVTO2(RCX(1),RCY(1))
        DO JP=1,IP
          CALL LINTO2(RCX(JP),RCY(JP))
        ENDDO
C       CALL IRCHART
```

```
        IP=IP+1
        GOTO 500
1000    NP=IP-1
C THE GRAPHICAL DATA ARE STORED AS IF THEY WERE DONE IN LARGE VIEWPT
        IF (VPTCS .NE. ASCRT(1) .AND. VPTCS .NE. ASCRT(2)) THEN
          CALL TRANSFM(NP,RCX,RCY,XRC,YRC,BVCXS,BVCXE,BVCYS,BVCYE,
     $                            XSSPC,XESPC,YSSPC,YESPC)
        ENDIF
        RETURN
        END

C*********************************************************
C***     ROUTINE TO DISPLAY IR  SPECTRUM IN RECORD    ****
        SUBROUTINE IRDISP(NP,XRC,YRC,ICOL)
        PARAMETER(MAXP=5000)
        REAL XRC(NP),YRC(NP),DSX(MAXP),DSY(MAXP)
        CHARACTER*20 ASCRT(2),VPTCS
        LOGICAL COMPAR
        COMMON /VIEWSIZE/ ASCRT,VPTCS
        COMMON /PSPCHART/ CTXS,CTXE,CTYS,CTYE
        COMMON /BCHTREGN/ BVCXS,BVCXE,BVCYS,BVCYE
        COMMON /COMPARISON/ COMPAR,IDNM
        IF (.NOT. COMPAR .OR. IDNM .LE. 1) THEN
          CALL IRCHART
        ENDIF
C CONVERT DATA TO FIT THE CHART
        IF (VPTCS .NE. ASCRT(1) .AND. VPTCS .NE. ASCRT(2)) THEN
          CALL TRANSFM(NP,XRC,YRC,DSX,DSY,CTXS,CTXE,CTYS,CTYE,
     $                            BVCXS,BVCXE,BVCYS,BVCYE)
        ELSE
          CALL TRANSFM(NP,XRC,YRC,DSX,DSY,BVCXS,BVCXE,BVCYS,BVCYE,
     $                            CTXS,CTXE,CTYS,CTYE)
        ENDIF
        CALL LINCOL(ICOL)
        CALL MOVTO2(DSX(1),DSY(1))
        DO I=1,NP
          CALL LINTO2(DSX(I),DSY(I))
        ENDDO
        RETURN
        END

C DRAW THE IR CHART
        SUBROUTINE IRCHART
        CHARACTER*3 INTS
        CHARACTER*4 WAVN
        CHARACTER*15 XASTR,YASTR
        CHARACTER*20 ASCRT(2),VPTCS
        COMMON /VIEWSIZE/ ASCRT,VPTCS
        COMMON /VIEWPT/ SVXLEN,SVYLEN,SVXST,SVYST,
     $                  BVXLEN,BVYLEN,BVXST,BVYST
        COMMON /DSHVMW/ DSBFH,DSBFV,WIDEM,FINT,XSMF,YSMF,TFHEIT
        COMMON /PARGENPAD/ HGENPAD
        COMMON /PSPCHART/ XST,XED,YST,YED
        COMMON /PRSPEC/ XSCR,YSCR,XECR,YECR
        COMMON /BCHTREGN/BVCXS,BVCXE,BVCYS,BVCYE
C ENQUIRE THE SIZE OF THE CHARACTER
        CALL CHAENQ(ITO,WDO,HTO,NO,AIO,ANO)
C DECIDE THE WINDOW SIZE FOR SPECTRUM
        SHN=4.0
        ANSH=2.0
        CHWD=WDO
        CHHT=HTO
        BVCXS=BVXST+SHN*DSBFV+SHN*DSBFV
        BVCXE=BVXST+BVXLEN-DSBFH-DSBFV
        BVCYS=SVYST+SHN*DSBFH
        BVCYE=SVYST+SVYLEN-DSBFH
        IF (VPTCS .EQ. ASCRT(1) .OR. VPTCS .EQ. ASCRT(2)) THEN
          XST=BVCXS
          XED=BVCXE
          YST=BVCYS
          YED=BVCYE
          XYPOS=YST-ANSH*DSBFH-DSBFV
          YXPOS=XST-ANSH*DSBFH-DSBFV
          XSCR=BVXST
          XECR=BVXST+BVXLEN
          YSCR=SVYST
          YECR=SVYST+SVYLEN
        ELSE
          XST=SVXST+SHN*DSBFH
          XED=SVXST+SVXLEN-DSBFH
          YST=SVYST+HGENPAD+SHN*DSBFH+DSBFV+TFHEIT
          YED=SVYST+SVYLEN-DSBFH
          XYPOS=YST-ANSH*DSBFH
          YXPOS=XST-ANSH*DSBFH
          XSCR=SVXST+DSBFV
          XECR=SVXST+SVXLEN-DSBFV
          YSCR=SVYST+HGENPAD+TFHEIT+DSBFH
          YECR=SVYST+SVYLEN-DSBFV
          CHWD=WDO/2.0
          CHHT=HTO/2.0
        ENDIF
        CALL CHASIZ(CHWD,CHHT)
        CALL CLEAR(XSCR,YSCR,XECR,YECR)
        XLEN=XED-XST
        YLEN=YED-YST
        XXPOS=XST+XLEN/2.0
        YYPOS=YST+YLEN/2.0-DSBFH
        ICAT=14
        CALL HSVDEF(ICAT,0.0,0.0,0.3)
        CALL LINCOL(ICAT)
        CALL MOVTO2(XST,YST)
        CALL LINBY2(0.0,YLEN)
        CALL LINBY2(XLEN,0.0)
        CALL LINBY2(0.0,-YLEN)
        CALL LINBY2(-XLEN,0.0)
        NHUNIT=56
        WUNIT=XLEN/NHUNIT
        HUNIT=WUNIT
        NVUNIT=INT(YLEN/HUNIT)
        DO I=1,NHUNIT
          CALL MOVTO2(XST+I*WUNIT,YST)
          CALL LINBY2(0.0,YLEN)
        ENDDO
        DO I=1,NVUNIT+1
          CALL MOVTO2(XST,YST+I*HUNIT)
          IF (I .EQ. NVUNIT+1) CALL MOVTO2(XST,YST+YLEN)
          CALL LINBY2(XLEN,0.0)
        ENDDO
C PUT SCALE VALUES ON THE CHARTAXISES
        CALL LINCOL(4)
        PX=XST-WUNIT
        PY=YST-DSBFH-DSBFV
        NWAV=4000
        DO I=1,5
          WRITE(UNIT=WAVN,FMT='(I4)') NWAV
          CALL MOVTO2(PX,PY)
          CALL CHASTR(WAVN)
          PX=PX+5*WUNIT
          NWAV=NWAV-500
        ENDDO
```

```
      PX=PX-WUNIT
      NWAV=1800
      DO I=1,10
        WRITE(UNIT=WAVN,FMT='(I4)') NWAV
        CALL MOVTO2(PX,PY)
        CALL CHASTR(WAVN)
        PX=PX+4*WUNIT
        NWAV=NWAV-200
      ENDDO
      NWAV=0
      NLP=3
      PX=XST-2.0*DSBFH
      PY=YST-CHHT/2.0
      NUNIT=10
      VINT=YLEN/FLOAT(NUNIT)
      DO I=1,NUNIT+1
        WRITE(UNIT=INTS,FMT='(I3)') NWAV
        CALL MOVTO2(PX,PY)
        CALL CHASTR(INTS)
        PY=PY+VINT
        NWAV=NWAV+10
      ENDDO
C PUT TITLE
      XASTR='Wavenumber'
      YASTR='Transmittence'
      CALL MOVTO2(XXPOS,XYPOS)
      CALL CHASTR(XASTR)
      CALL CHASWI(1.0)
      CALL CHAANG(90.0)
      CALL MOVTO2(YXPOS,YYPOS)
      CALL CHASTR(YASTR)
      CALL CHAANG(0.0)
C RESTORE THE ORIGINAL SIZE OF CHARACTER
      CALL CHASIZ(WDO,HTO)
      RETURN
      END

C*******************************************************
C***    ROUTINE TO SCAN UV  SPECTRUM                ****
      SUBROUTINE UVSCAN(NP,XRC,YRC)
C*******************************************************
      PARAMETER (MAXP=50000)
      REAL XRC(MAXP),YRC(MAXP),RCX(MAXP),RCY(MAXP)
      CHARACTER*20 ASCRT(2),VPTCS
      LOGICAL SCAN
      COMMON /VIEWSIZE/ ASCRT,VPTCS
      COMMON /PSPCHART/ XSSPC,XESPC,YSSPC,YESPC
      COMMON /BCHTREGN/ BVCXS,BVCXE,BVCYS,BVCYE
      COMMON /GFEVEN/KEY,IMPKEY,IMPDAT,NSEG,XPIC,YPIC,
     4                  NARGS,ARGS(80)
      COMMON /SCANORDISP/SCAN
      SCAN=.TRUE.
10    CALL CURSTR('SER')
      CALL UVCHART
      IND=1000
      XSIND=(XSSPC+XESPC)/2.0
      YSIND=(YSSPC+YESPC)/2.0
      CALL INDICATOR(IND,XSIND,YSIND)
      IP=1
100   CALL MOVTO2(XPIC,YPIC)
500   CALL DRAG(IND)
      GOTO(100,1000,10,200),KEY
200   RCX(IP)=XPIC
      RCY(IP)=YPIC
      CALL LINCOL(8)
```

```
        CALL MOVTO2(RCX(1),RCY(1))
        DO JP=1,IP
          CALL LINTO2(RCX(JP),RCY(JP))
        ENDDO
C       CALL IRCHART
        IP=IP+1
        GOTO 500
1000    NP=IP-1
C THE GRAPHICAL DATA ARE STORED AS IF THEY WERE DONE IN LARGE VIEWPT
        IF (VPTCS .NE. ASCRT(1) .AND. VPTCS .NE. ASCRT(2)) THEN
          CALL TRANSFM(NP,RCX,RCY,XRC,YRC,BVCXS,BVCXE,BVCYS,BVCYE,
     $                                 XSSPC,XESPC,YSSPC,YESPC)
        ENDIF
        RETURN
        END

C*********************************************************
C***     ROUTINE TO DISPLAY UV  SPECTRUM IN RECORD      ****
        SUBROUTINE UVDISP(NP,XRC,YRC,ICOL)
        PARAMETER(MAXP=5000)
        REAL XRC(NP),YRC(NP),DSX(MAXP),DSY(MAXP)
        LOGICAL SCAN,COMPAR
        CHARACTER*20 ASCRT(2),VPTCS
        COMMON /VIEWSIZE/ ASCRT,VPTCS
        COMMON /PSPCHART/ CTXS,CTXE,CTYS,CTYE
        COMMON /BCHTREGN/ BVCXS,BVCXE,BVCYS,BVCYE
        COMMON /SCANORDISP/ SCAN
        COMMON /UVSCALE/WAVS,WAVE,APSS,APSE
        COMMON /COMPARISON/ COMPAR,IDNM
        SCAN=.FALSE.
        IF (.NOT. COMPAR .OR. IDNM .LE. 1) THEN
          CALL UVCHART
        ENDIF
C CONVERT DATA TO FIT THE CHART
        IF (VPTCS .NE. ASCRT(1) .AND. VPTCS .NE. ASCRT(2)) THEN
          CALL TRANSFM(NP,XRC,YRC,DSX,DSY,CTXS,CTXE,CTYS,CTYE,
     $                                 BVCXS,BVCXE,BVCYS,BVCYE)
        ELSE
          CALL TRANSFM(NP,XRC,YRC,DSX,DSY,BVCXS,BVCXE,BVCYS,BVCYE,
     $                                 CTXS,CTXE,CTYS,CTYE)
        ENDIF
        CALL LINCOL(ICOL)
        CALL MOVTO2(DSX(1),DSY(1))
        DO I=1,NP
          CALL LINTO2(DSX(I),DSY(I))
        ENDDO
        RETURN
        END

C DRAW THE UV SPECTRUM CHART
        SUBROUTINE UVCHART
        CHARACTER*5 WAVL,APSV
        CHARACTER*15 XASTR,YASTR
        CHARACTER*20 ASCRT(2),VPTCS,MESSG
        LOGICAL SCAN
        COMMON /VIEWSIZE/ ASCRT,VPTCS
        COMMON /VIEWPT/ SVXLEN,SVYLEN,SVXST,SVYST,
     $                       BVXLEN,BVYLEN,BVXST,BVYST
        COMMON /DSHVMW/ DSBFH,DSBFV,WIDEM,FINT,XSMF,YSMF,TFHEIT
        COMMON /PARGENPAD/ HGENPAD
        COMMON /PSPCHART/ XST,XED,YST,YED
        COMMON /PRSPEC/ XSCR,YSCR,XECR,YECR
        COMMON /BCHTREGN/BVCXS,BVCXE,BVCYS,BVCYE
        COMMON /UVSCALE/ WAVS,WAVE,APSS,APSE
```

```
      COMMON /SCANORDISP/ SCAN
      DATA MESSG /'Wmin,Wmax,Absn,Absm'/
C ENQUIRE THE SIZE OF THE CHARACTER
      CALL CHAENQ(ITO,WDO,HTO,NO,AIO,ANO)
C DECIDE THE WINDOW SIZE FOR SPECTRUM
      SHN=5.0
      ANSH=3.0
      CHWD=WDO
      CHHT=HTO
      BVCXS=BVXST+(SHN+1.0)*DSBFH+SHN*DSBFV
      BVCXE=BVXST+SVYLEN-DSBFH-DSBFV
      BVCYS=SVYST+SHN*DSBFH
      BVCYE=SVYST+SVYLEN-DSBFH
      IF (VPTCS .EQ. ASCRT(1) .OR. VPTCS .EQ. ASCRT(2)) THEN
        XST=BVCXS
        XED=BVCXE
        YST=BVCYS
        YED=BVCYE
        XYPOS=YST-ANSH*DSBFH-DSBFV
        YXPOS=XST-(ANSH+1.0)*DSBFH-DSBFV
        XSCR=BVXST
        XECR=BVXST+BVXLEN
        YSCR=SVYST
        YECR=SVYST+SVYLEN
      ELSE
        XST=SVXST+(SHN+1.0)*DSBFH
        YST=SVYST+HGENPAD+SHN*DSBFH+DSBFV+TFHEIT
        YED=SVYST+SVYLEN-DSBFH
        XED=XST+YED-YST
        XYPOS=YST-ANSH*DSBFH
        YXPOS=XST-(ANSH+1.0)*DSBFH
        XSCR=SVXST+DSBFV
        XECR=SVXST+SVXLEN-DSBFV
        YSCR=SVYST+HGENPAD+TFHEIT+DSBFH
        YECR=SVYST+SVYLEN-DSBFV
        CHWD=WDO/2.0
        CHHT=HTO/2.0
      ENDIF
      CALL CHASIZ(CHWD,CHHT)
      CALL CLEAR(XSCR,YSCR,XECR,YECR)
      XLEN=XED-XST
      YLEN=YED-YST
      XXPOS=XST+XLEN/2.0
      YYPOS=YST+YLEN/2.0-DSBFH
      ICAT=14
      CALL HSVDEF(ICAT,0.0,0.0,0.3)
      CALL LINCOL(ICAT)
      CALL MOVTO2(XST,YST)
      CALL LINBY2(0.0,YLEN)
      CALL LINBY2(XLEN,0.0)
      CALL LINBY2(0.0,-YLEN)
      CALL LINBY2(-XLEN,0.0)
      NHUNIT=10
      NVUNIT=NHUNIT
      WUNIT=XLEN/NHUNIT
      HUNIT=YLEN/NVUNIT
      DO I=1,NHUNIT
        CALL MOVTO2(XST+I*WUNIT,YST)
        CALL LINBY2(0.0,YLEN)
      ENDDO
      DO I=1,NVUNIT
        CALL MOVTO2(XST,YST+I*HUNIT)
        IF (I .EQ. NVUNIT) CALL MOVTO2(XST,YST+YLEN)
        CALL LINBY2(XLEN,0.0)
      ENDDO
C TAKE SCALE VALUES
```

```
    IF (SCAN)THEN
      CALL READVAL(MESSG)
    ENDIF
C PUT SCALE VALUES ON THE CHARTAXISES
    CALL LINCOL(4)
    PX=XST-5.0*CHWD/2.0
    PY=YST-DSBFH-DSBFV
    WAV=WAVS
    WAVINT=(WAVE-WAVS)/NHUNIT
    DO I=1,NHUNIT-4
      WRITE(UNIT=WAVL,FMT='(F5.1)') WAV
      CALL MOVTO2(PX,PY)
      CALL CHASTR(WAVL)
      PX=PX+2*WUNIT
      WAV=WAV+2.0*WAVINT
    ENDDO
    NWAV=0
    NLP=3
    PX=XST-ANSH*DSBFH
    PY=YST-CHHT/2.0
    APSINT=(APSE-APSS)/NVUNIT
    APS=APSS
    DO I=1,NVUNIT-4
      WRITE(UNIT=APSV,FMT='(F5.3)') APS
      CALL MOVTO2(PX,PY)
      CALL CHASTR(APSV)
      PY=PY+2.0*HUNIT
      APS=APS+2.0*APSINT
    ENDDO
C PUT TITLE
    XASTR='Wavelength'
    YASTR='Absorption'
    CALL MOVTO2(XXPOS,XYPOS)
    CALL CHASTR(XASTR)
    CALL CHASWI(1.0)
    CALL CHAANG(90.0)
    CALL MOVTO2(YXPOS,YYPOS)
    CALL CHASTR(YASTR)
    CALL CHAANG(0.0)
C RESTORE THE ORIGINAL SIZE OF CHARACTER
    CALL CHASIZ(WDO,HTO)
    RETURN
    END

C READ VALUES FROM DIALOG BOX
    SUBROUTINE READVAL(TEXT)
    INTEGER SMGID,PBD,KBD,VROW,VCOL
    CHARACTER*20 ANSER
    CHARACTER*20 TEXT
    LOGICAL PROMP
    COMMON /DSHVMW/ DSBFH,DSBFV,WIDEM,FINT
    COMMON /DIABOX/ DFXS,DFYS,WIDEPR,DFHEIT
    COMMON /SMGPAR/ SMGID,PBD,KBD,VROW,VCOL
    COMMON /DFTCHAR/ DCHWD,DCHHT
    COMMON /UVSCALE/VAL1,VAL2,VAL3,VAL4
    CALL CHAENQ(ITO,WDO,HTO,NIO,AIO,ANO)
    CHWD=DCHWD
    CHHT=DCHHT
    CALL CHASIZ(CHWD,CHHT)
    WIDEPR=80.0
    BXS=DFXS+DSBFH
    BYS=DFYS+DSBFV
    BXE=DFXS+WIDEPR-DSBFH
    BYE=DFYS+DFHEIT-DSBFV
    CALL LINCOL(6)
```

```
        CALL MOVTO2(DFXS+WIDEPR,DFYS)
        CALL LINBY2(0.0,DFHEIT)
        CALL LINCOL(4)
        CALL MOVTO2(BXS,BYS+DSBFV)
        CALL CHASTR(TEXT)
        CALL CHAMOD
        VROW=24
        VCOL=31
        CALL SMG$SET_CURSOR_ABS(SMGID,VROW,VCOL)
        READ(5,*) VAL1,VAL2,VAL3,VAL4
        CALL DIATCL
        CALL CLEAR(BXS,BYS,BXE,BYE)
        CALL CHASIZ(WDO,HTO)
        RETURN
        END
```

```

C TRANSFORM THE DATA TO A SPECIAL WINDOW
        SUBROUTINE TRANSFM(NPT,X,Y,RX,RY,TXN,TXM,TYN,TYM,
     $                                  OXN,OXM,OYN,OYM)
        REAL X(NPT),Y(NPT),RX(NPT),RY(NPT)
        XK=(TXM-TXN)/(OXM-OXN)
        XC=TXN-XK*OXN
        YK=(TYM-TYN)/(OYM-OYN)
        YC=TYN-YK*OYN
        DO I=1,NPT
         RX(I)=X(I)*XK+XC
         RY(I)=Y(I)*YK+YC
        ENDDO
        RETURN
        END
```

```

C DRAW A SYMBOL AS INDICATOR
        SUBROUTINE INDICATOR(IND,XST,YST)
        ICOL=15
        CALL HSVDEF(15,0.0,0.0,1.0)
        CALL PICBEG(IND)
           CALL LINCOL(ICOL)
           CALL MOVTO2(XST,YST)
           CALL MOVTO2(XST-1.0,YST)
           CALL LINBY2(2.0,0.0)
           CALL MOVTO2(XST,YST-1.0)
           CALL LINBY2(0.0,2.0)
        CALL PICEND
        RETURN
        END
```

```

C**************************************************
C****     MAIN MENU AND GENERAL INFORMATION PAD ****
        SUBROUTINE MAINPAD
C**************************************************
        PARAMETER (NMMAX=10,MNTEXT=100)
        CHARACTER*15 MENIT(NMMAX)
        CHARACTER*20 MESSG,ASFD
        CHARACTER*80 INTRT(MNTEXT),OPGIDE(MNTEXT)
        LOGICAL AISINF,PROMP
        COMMON /PSCREN/ ORX,ORY,SXLEN,SYLEN
        COMMON /VIEWPT/ SVXLEN,SVYLEN,SVXST,SVYST,
     $                  BVXLEN,BVYLEN,BVXST,BVYST
        COMMON /DSHVMW/ DSBFH,DSBFV,WIDEMF,FINT,XSMF,YSMF
        COMMON /DIABOX/ DFXS,DFYS,WIDEPR,DFHEIT
        COMMON /LOGVAR/ AISINF
        DATA (MENIT(I),I=1,7)/'Set Viewport','List Entries',
     $                              'Add New Entry','Delete Entry',
     $                        'Search','Utilities','Exit'/
        DATA MESSG /'[RET] to start'/
        NMENIT=7
C DISTANCE BETWEEN FRAME
        DSBFV=2.0
        DSBFH=5.0
C DRAW THE LARGE SURROUNDING
        CALL LGFRAME
C DISPLAY THE TITLE
        TFHEIT=8.0
        TFWIDE=SXLEN-2.0*DSBFH
        TFXS=ORX+DSBFH
        TFYS=ORY+SYLEN-TFHEIT-DSBFV
        CALL FEFRAME(TFXS,TFYS,TFWIDE,TFHEIT,7)
        CALL LINCOL(5)
        CALL MOVTO2(TFXS+TFWIDE/3.0,TFYS+DSBFV)
        CALL CHASTR('  A  I  S*.')
C DISPLAY THE MENU
        DFHEIT=10.0
        HEITMF=SYLEN-4.0*DSBFV-TFHEIT-DFHEIT
        WIDEMF=40.0
        FINT=14.2
        XSMF=TFXS
        YSMF=TFYS-DSBFV
        ICOLF=7
        ICOLT=6
        CALL MENUDR(XSMF,YSMF,WIDEMF,HEITMF,FINT,ICOLF,
     $                  NMENIT,MENIT,ICOLT)
C DRAW DIALOG FRAME AT BOTTOM
        DFWIDE=SXLEN-2.0*DSBFH
        DFXS=XSMF
        DFYS=ORY+DSBFV
        CALL FEFRAME(DFXS,DFYS,DFWIDE,DFHEIT,6)
C INFORMATION PAD ABOUT AIS
        CALL CHAENQ(ITO,WDO,HTO,NIO,AIO,ANO)
        WDICH=2.0*WDO
        HTICH=HTO
        SHN=3.0
        IF (AISINF) THEN
C READ INTRODUCTION AND OPERATION GUIDE TEXT FROM A FILE
          OPEN(UNIT=12,STATUS='OLD',FILE='INTRTEXT.AIS')
            READ(12,111) NINTEX
            DO I=1,NINTEX
              READ(12,222) INTRT(I)
            ENDDO
            READ(12,111) NOPTEX
            DO I=1,NOPTEX
              READ(12,222) OPGIDE(I)
            ENDDO
          CLOSE(UNIT=12)
          CALL CHASIZ(WDICH,HTICH)
          TOPHEIT=60.0
          TOPWIDE=SXLEN-3.0*DSBFH-WIDEMF
          TOPXS=ORX+2.0*DSBFH+WIDEMF
          TOPYS=TFYS-DSBFV-TOPHEIT
          CALL FEFRAME(TOPXS,TOPYS,TOPWIDE,TOPHEIT,7)
          XSPIT=TOPXS+SHN*DSBFH
          YSPIT=TOPYS+TOPHEIT-SHN*DSBFH
          CALL MOVTO2(XSPIT,YSPIT)
          CALL LINCOL(8)
          CALL CHASTR(INTRT(1))
          CALL CHASIZ(WDO,HTO)
          CALL LINCOL(5)
          DO I=2,NINTEX
            YSPIT=YSPIT-DSBFH
            CALL MOVTO2(XSPIT,YSPIT)
            CALL CHASTR(INTRT(I))
          ENDDO
          BOTHEIT=HEITMF-TOPHEIT-DSBFV
          BOTWIDE=TOPWIDE
          BOTXS=TOPXS
          BOTYS=TOPYS-DSBFV-BOTHEIT
          CALL FEFRAME(BOTXS,BOTYS,BOTWIDE,BOTHEIT,7)
          XSPIT=BOTXS+SHN*DSBFH
          YSPIT=BOTYS+BOTHEIT-(SHN+1.0)*DSBFH
          CALL CHASIZ(WDICH,HTICH)
          CALL LINCOL(8)
          CALL MOVTO2(XSPIT,YSPIT)
          CALL CHASTR(OPGIDE(1))
          CALL CHASIZ(WDO,HTO)
          CALL LINCOL(6)
          DO I=2,NOPTEX
            YSPIT=YSPIT-DSBFH
            CALL MOVTO2(XSPIT,YSPIT)
            CALL CHASTR(OPGIDE(I))
          ENDDO
          PROMP=.FALSE.
          CALL PROMPT(MESSG,ASFD,PROMP)
        ENDIF
        AISINF=.FALSE.
        CALL CHAMOD
C PARAMETERS FOR SMALL AND LARGE VIEWPORT
        SVXST=BOTXS
        SVYST=BOTYS
        SVXLEN=BOTWIDE
        SVYLEN=TOPHEIT+BOTHEIT+DSBFV
        BVXST=DFXS
        BVYST=DFYS
        BVXLEN=DFWIDE
        BVYLEN=HEITMF+DFHEIT+DSBFV
111     FORMAT(5X,I)
222     FORMAT(5X,A)
        RETURN
        END
```

```

C DRAW MENU WITH TEXT
        SUBROUTINE MENUDR(XPOS,YPOS,WIDE,HEIT,DINT,ICF,NTEXT,TEXT,ICT)
        COMMON /DSHVMW/DSBFH,DSBFV,WIDEMF,FINT
        COMMON /SIZMNC/ CHAWIDE,CHAHEIT
        CHARACTER*15 TEXT(NTEXT)
        CHAWIDE=2.0
        CHAHEIT=3.0
        XS=XPOS
        YS=YPOS
        CALL FEFRAME(XPOS,YPOS,WIDE,-HEIT,ICF)
        CALL HSFRAME(XPOS,YPOS,WIDE,HEIT,-DINT)
        XS=XS+DSBFV
        YS=YS-DINT/2.0
C ENQUIRE PREVIOUR CHARACTER SIZE
        CALL CHAENQ(ITO,WDO,HTO,NO,AIO,ANO)
        CALL CHASIZ(CHAWIDE,CHAHEIT)
        CALL LINCOL(ICT)
        DO I=1,NTEXT
          CALL MOVTO2(XS,YS)
          CALL CHASTR(TEXT(I))
          YS=YS-DINT
        ENDDO
        CALL CHASIZ(WDO,HTO)
        RETURN
        END
```

```
C DRAW THE GENERAL INFORMATION PAD
        SUBROUTINE GENPAD
        PARAMETER (MAXNL=20,HGENPAD_DFT=120.0,HEITFT_DFT=10.0,
     $             TFHEIT_DFT=25.0,CHWDT_DFT=5.0,CHHTT_DFT=5.0,
     $             NCRTV=2,NSUB=6)
        REAL FTXS(MAXNL),FTYS(MAXNL),VLXS(MAXNL),VLYS(MAXNL)
        REAL CHXP(MAXNL),CHYP(MAXNL)
        REAL TBWIDE(MAXNL),TBHEIT(MAXNL),TBXS(MAXNL),TBYS(MAXNL)
        REAL ASXP(MAXNL),ASYP(MAXNL)
        CHARACTER*20 ANSCRT(NCRTV),VPTCS
        CHARACTER*15 GMITEM(MAXNL),DMENT(NSUB)
        COMMON /PSCREN/ ORX,ORY,SXLEN,SYLEN
        COMMON /VIEWPT/ SVXLEN,SVYLEN,SVXST,SVYST,
     $                  BVXLEN,BVYLEN,BVXST,BVYST
        COMMON /DSHVMW/ DSBFH,DSBFV,WIDEMF,FINT,XSMF,YSMF,TFHEIT
        COMMON /DIABOX/ DFXS,DFYS,WIDEPR,DFHEIT
        COMMON /VIEWSIZE/ ANSCRT,VPTCS
        COMMON /GINFITEM/ NITEM,GMITEM
        COMMON /POSITEM/ ASXP,ASYP
        COMMON /PARGENPAD/ HGENPAD
        COMMON /PRSPEC/ XSCR,YSCR,XECR,YECR
        COMMON /SUBMENU1/ NSUBM,DMENT
        DATA (GMITEM(I),I=1,14)/'Chemical Name','Commercial Names',
     $        '1','2','3','Manufacturer','Aston Code','Cost',
     $        'Mol. Weight','Melting Point','Boiling Point',
     $        'Colour','Other Descp. ','Note'/
        DATA (DMENT(I),I=1,NSUB)/'Set View','Chem. Structure',
     $                          'Infrared Spectr','UV Spectr',
     $                          'Other Inf.','Quit'/
        NITEM=14
        NSUBM=NSUB
C ASSIGN SOME BASIC PARAMETERS FOR FRAMES
        IF (VPTCS .EQ. ANSCRT(1) .OR. VPTCS .EQ. ANSCRT(2)) THEN
          HGENPAD=HGENPAD_DFT
          HEITFT=HEITFT_DFT
          TFHEIT=TFHEIT_DFT+DSBFV
          CHWDT=CHWDT_DFT
          CHHTT=CHHTT_DFT
          CHTXS=SVXLEN/4.0
          CHTYS=2.0*TFHEIT/5.0
          DSBTF=DSBFH
          CALL CLEAR(SVXST,SVYST,SVXST+SVXLEN,SVYST+SVYLEN)
        ELSE
          HGENPAD=HGENPAD_DFT/2.0
          HEITFT=HEITFT_DFT/2.0
          TFHEIT=TFHEIT_DFT/2.0
          CHWDT=CHWDT_DFT/2.0
          CHHTT=CHHTT_DFT/2.0
          CHTXS=SVXLEN/3.0
          CHTYS=2.0*TFHEIT/5.0
          DSBTF=DSBFV
          CALL CLEAR(SVXST,SVYST,SVXST+SVXLEN,SVYST+SVYLEN)
        ENDIF
        RTCY=SVYST+HGENPAD
        WIDEFT=SXLEN/3.0
        WIDESF=(SVXLEN-DSBFV)/4.0
C DRAW THE SURROUNDING FRAME
        CALL FEFRAME(SVXST,SVYST,SVXLEN,SVYLEN,7)
C DRAW THE TITLE AND FRAME WITH IT
        TFXST=SVXST
        TFYST=SVYST+HGENPAD+DSBTF
        TFWIDE=SVXLEN
        ICGENPAD=7
        CALL LINCOL(ICGENPAD)
        CALL FEFRAME(TFXST,TFYST,TFWIDE,TFHEIT,7)
        CALL CHAENQ(ITO,WDO,HTO,NO,AIO,ANO)
        CALL CHASIZ(CHWDT,CHHTT)
        CALL MOVTO2(TFXST+CHTXS,TFYST+CHTYS)
        CALL LINCOL(8)
        CALL CHASTR('General Information*.')
        CALL CHASIZ(WDO,HTO)
C DRAW OTHER FRAMES IN THE GENPAD
C ASSIGN HORIZONTAL LINES' COORDINATES
        FTXS(1)=SVXST
        FTYS(1)=RTCY-HEITFT
        FTXS(2)=SVXST
        FTYS(2)=RTCY-4.0*HEITFT
        FTXS(3)=SVXST+2.0*WIDEFT/3.0
        FTYS(3)=RTCY-2.0*HEITFT
        FTXS(4)=FTXS(3)
        FTYS(4)=RTCY-3.0*HEITFT
        NHL=12
        DO I=5,NHL
          FTXS(I)=SVXST
          FTYS(I)=RTCY-FLOAT(I)*HEITFT
          IF ( I .GT. 9) THEN
            FTYS(I)=RTCY-FLOAT(I-1)*HEITFT
          ENDIF
          IF (I .EQ. 9) THEN
            FTXS(I)=FTXS(I-1)+2.0*WIDESF+DSBFV
            FTYS(I)=FTYS(I-1)
          ENDIF
        ENDDO
        CALL LINCOL(ICGENPAD)
        CALL HOLINE(SVXST,RTCY,SVXLEN)
        DO I=1,NHL
          WIDEF=SVXLEN
          IF (I .EQ. 3 .OR. I .EQ. 4) THEN
            WIDEF=SVXLEN-2.0*WIDEFT/3.0
          ENDIF
          IF (I .EQ. 8 .OR. I .EQ. 9) THEN
            WIDEF=2.0*WIDESF
          ENDIF
          CALL HOLINE(FTXS(I),FTYS(I),WIDEF)
        ENDDO
C ASSIGN VERTICAL LINES' COORDINATES
        WIDECM=5.0*WIDEFT/6.0
        VLXS(1)=SVXST+WIDECM
        VLYS(1)=FTYS(1)
        VLXS(2)=SVXST+2.0*WIDEFT/3.0
        VLYS(2)=FTYS(2)
        VLXS(3)=SVXST+WIDEFT
        VLYS(3)=VLYS(2)
        VLXS(4)=VLXS(1)
        VLYS(4)=FTYS(7)
        VLXS(5)=SVXST+WIDESF
        VLYS(5)=FTYS(10)
        VLXS(6)=VLXS(5)+WIDESF
        VLYS(6)=VLYS(5)
        VLXS(7)=VLXS(6)+DSBFV
        VLYS(7)=VLYS(6)
        VLXS(8)=VLXS(7)+WIDESF
        VLYS(8)=VLYS(7)
        VLXS(9)=VLXS(1)
        VLYS(9)=SVYST
        NVL=9
        DO I=1,NVL
          HEITSF=3.0*HEITFT
          IF (I .EQ. 1) THEN
            HEITSF=HEITFT
```

```
         ELSE IF(I .GE. 5 .AND. I .LE. 8) THEN
           HEITSF=2.0*HEITFT
         ENDIF
         CALL VELINE(VLXS(I),VLYS(I),HEITSF)
       ENDDO
C ASSIGN COORDINATES FOR THE DESCRIPTERS IN PAD
       CHXS=WIDEFT/5.0
       CHYS=HEITFT/3.0
       CALL LINCOL(6)
       TBWIDE(1)=WIDECM
       TBHEIT(1)=HEITFT
       TBWIDE(2)=(2.0*WIDEFT/3.0)
       TBHEIT(2)=(3.0*HEITFT)
       DO I=3,5
         TBWIDE(I)=(WIDEFT/3.0)
         TBHEIT(I)=HEITFT
       ENDDO
       DO I=6,8
         TBWIDE(I)=WIDECM
         TBHEIT(I)=HEITFT
       ENDDO
       DO I=9,12
         TBWIDE(I)=WIDESF
         TBHEIT(I)=HEITFT
       ENDDO
       DO I=13,NITEM
         TBWIDE(I)=WIDECM
         TBHEIT(I)=HEITFT
       ENDDO
       DO I=1,NITEM
         TBXS(I)=TBWIDE(I)/5.0
         TBYS(I)=TBHEIT(I)/4.0
       ENDDO
       DO I=1,NITEM
         CHXP(I)=FTXS(I)+TBXS(I)
         CHYP(I)=FTYS(I)+TBYS(I)
         IF (I .GT. 5) THEN
           CHXP(I)=FTXS(I-1)+TBXS(I)
           CHYP(I)=FTYS(I-1)+TBYS(I)
         ENDIF
         IF (I .EQ. 2) CHYP(I)=FTYS(4)+HEITFT/2.0
         IF (I .EQ. 5) THEN
           CHXP(I)=FTXS(3)+TBXS(I)
           CHYP(I)=FTYS(2)+TBYS(I)
         ENDIF
         IF (I .EQ. 10 .OR. I .EQ. 12) THEN
           CHXP(I)=CHXP(I-1)+2.0*WIDESF+DSBFV
           CHYP(I)=CHYP(I-1)
         ENDIF
         IF (I .GT. 12) THEN
           CHXP(I)=CHXP(1)
           CHYP(I)=FTYS(10)-FLOAT(I-12)*HEITFT+TBYS(I)
         ENDIF
         CALL MOVTO2(CHXP(I),CHYP(I))
         CALL CHASTR(GMITEM(I))
       ENDDO
       ASXP(1)=CHXP(1)+TBWIDE(1)
       ASYP(1)=CHYP(1)
       DO I=2,NITEM-1
         ASXP(I)=CHXP(I+1)+TBWIDE(I+1)
         ASYP(I)=CHYP(I+1)
       ENDDO
C SET WINDOW VALUES FOR SPECTRUM CHART
       IF (VPTCS .EQ. ANSCRT(1) .OR. VPTCS .EQ. ANSCRT(2)) THEN
         XSCR=BVXST
         XECR=BVXST+BVXLEN
```

```
          YSCR=SVYST
          YECR=SVYST+SVYLEN
        ELSE
          XSCR=SVXST+DSBFV
          XECR=SVXST+SVXLEN-DSBFV
          YSCR=SVYST+HGENPAD+TFHEIT+DSBFH
          YECR=SVYST+SVYLEN-DSBFV
        ENDIF
        RETURN
        END

C LIST ALL ENTRIES IN THE BASE, ENTRIES ARE SORTED
        SUBROUTINE LISTENTR
        PARAMETER (MAXENTRY=1000,MAXITM=20,NMSG=2)
        CHARACTER*2 EOS
        CHARACTER*20 MESSG(NMSG),ANSFD
        CHARACTER*20 ITEMS(MAXENTRY,MAXITM),ENTRL(MAXENTRY)
        COMMON /VIEWPT/ SVXLEN,SVYLEN,SVXST,SVYST,
     $                  BVXLEN,BVYLEN,BVXST,BVYST
        COMMON /DSHVMW/ DSBFH,DSBFV
        DATA (MESSG(I),I=1,NMSG)/'Continue?','[RET] to continue!'/
        DATA EOS/'*.'/
C READ ALL ENTRIES FROM THE NAMELIS FILE
        NIT=6
        NST=3
        NENT=0
        OPEN(UNIT=12,STATUS='OLD',FILE='NAMELIS.AIS')
          REWIND(UNIT=12)
          DO WHILE(.TRUE.)
            NENT=NENT+1
            DO I=1,NIT,NST
              READ(12,111,END=100) (ITEMS(NENT,J),J=I,I+NST-1)
            ENDDO
            ENTRL(NENT)=ITEMS(NENT,1)
          ENDDO
100     CLOSE(UNIT=12)
        NENT=NENT-1
C DISPLAY THE FIRST NAME ALL IDENTITIES, IT IS POSSIBLE TO
C EXTEND TO COVER ALL IDENTIFYING NAMES OR REQUEST
        CALL CLEAR(SVXST,SVYST,SVXST+SVXLEN,SVYST+SVYLEN)
        CALL FEFRAME(SVXST,SVYST,SVXLEN,SVYLEN,7)
        TLHEIT=10.0
        TYS=SVYST+SVYLEN-TLHEIT
        CALL FEFRAME(SVXST,TYS,SVXLEN,TLHEIT,7)
        CALL MOVTO2(SVXST+40.0,TYS+3.0)
        CALL LINCOL(8)
        CALL CHASTR('Entries in the Database*.')
        NLPP=15
        NST=10
        TXN=SVXST+DSBFH
        RTX=SVXST+SVXLEN-DSBFH
        TPY=TYS-2.0*DSBFH
        BTY=SVYST+DSBFH
        YSHT=2.0*DSBFH
        CALL LINCOL(4)
        DO I=1,NENT
          IF (I .EQ. 1 .OR. TXS .GT. RTX) THEN
            TXS=TXN
            TYS=TYS-YSHT
          ELSE
            CALL POSPIC(XRTC,YRTC,ZRTC)
            TXS=XRTC+DSBFV
          ENDIF
          IF (TYS .LT. BTY) THEN
            CALL CLEAR(TAXS,TAYS,TAXE,TAYE)
```

```
C DISPLAY PROMP TO DECIDE CONT. OR QUIT
            PROMP=.FALSE.
            CALL PROMPT(MESSG(1),ANSFD,PROMP)
            IF (ANSFD .EQ. 'N' .OR. ANSFD .EQ. 'NO') GOTO 200
            TYS=TPY
            TXS=TXN
          ENDIF
          CALL MOVTO2(TXS,TYS)
          CALL CHASTR(ENTRL(I))
        ENDDO
111     FORMAT(10X,3A20)
        CALL PROMPT(MESSG(2),ANSFD,PROMP)
200     RETURN
        END

C ADD NEW ENTRY TO THE DATABASE
        SUBROUTINE ADDENTRY
        PARAMETER (MAXNL=20,MAXENTRY=1000)
        INTEGER IDSTRN(MAXENTRY)
        REAL XP(MAXNL),YP(MAXNL)
        CHARACTER*15 GMITEM(MAXNL)
        CHARACTER*20 GINFRP(MAXNL),DSITEM(MAXNL)
        CHARACTER*20 SITEM(MAXENTRY),TSITEM(MAXENTRY,MAXNL)
        LOGICAL PROMP
        COMMON /GINFITEM/ NITEM,GMITEM
        COMMON /AVGENINF/ NDITEM,GINFRP
        COMMON /POSITEM/ XP,YP
        EXTERNAL STR$CONCAT
C DISPLAY THE GENERAL INF. PAD
        CALL GENPAD
C NUMBER OF DISPLAYED ITEMS
        NDITEM=NITEM-1
        DSITEM(1)=GMITEM(1)
C CONCAT FUNCION
        DO I=2,4
          CALL STR$CONCAT(DSITEM(I),GMITEM(2),GMITEM(I+1))
        ENDDO
        DO I=5,NDITEM
          DSITEM(I)=GMITEM(I+1)
        ENDDO
C READ INPUT AND DISPLAY IT IN GENPAD
        PROMP=.TRUE.
        DO I=1,NDITEM
          CALL PROMPT(DSITEM(I),GINFRP(I),PROMP)
          CALL LINCOL(5)
          CALL MOVTO2(XP(I),YP(I))
          CALL CHASTR(GINFRP(I))
        ENDDO
C STORE THE INPUTS TO THE MAIN FILE
        NSTEP=3
        OPEN (UNIT=12,STATUS='OLD',ACCESS='APPEND',
     $        FILE='MFILE.AIS')
          WRITE(12,111) NDITEM
          DO I=1,NDITEM,NSTEP
            IF ((NDITEM-I) .LT. NSTEP) THEN
              JE=NDITEM
            ELSE
              JE=I+NSTEP-1
            ENDIF
            WRITE(12,222) (GINFRP(J),J=I,JE)
          ENDDO
        CLOSE(UNIT=12)
C STORE THE IDENTIFYING NAMES IN NAMELIS FILE
        NIT=6
        NST=3
        OPEN (UNIT=12,STATUS='OLD',ACCESS='APPEND',
     $        FILE='NAMELIS.AIS')
          DO I=1,NIT,NST
            WRITE(12,222) (GINFRP(J),J=I,I+NST-1)
          ENDDO
        CLOSE(UNIT=12)
C SORT THE NAMELIS FILE ACCORDING TO FIRST NAME(CHEM. NAME)
C READ ALL ENTRYIES IN NAMELIS FILE
        NENT=0
        OPEN(UNIT=12,STATUS='OLD',FILE='NAMELIS.AIS')
          REWIND(UNIT=12)
          DO WHILE(.TRUE.)
            NENT=NENT+1
            DO I=1,NIT,NST
              READ(12,222,END=100) (TSITEM(NENT,J),J=I,I+NST-1)
            ENDDO
            SITEM(NENT)=TSITEM(NENT,1)
            IDSTRN(NENT)=NENT
          ENDDO
100     CLOSE(UNIT=12)
        NENT=NENT-1
        CALL SSORT(NENT,SITEM,IDSTRN)
C STORE THE SORTED ENTRIES BACK TO NAMELIS FILE
        OPEN (UNIT=12,STATUS='NEW',FILE='NAMELIS.AIS')
          DO I=1,NENT
            IDS=IDSTRN(I)
            DO J=1,NIT,NST
              WRITE(12,222) (TSITEM(IDS,K),K=J,J+NST-1)
            ENDDO
          ENDDO
        CLOSE(UNIT=12)
C CALL INPUT PROCEDURE FOR THE DETAILED CONTENTS
        CALL INPTINF
111     FORMAT(10X,I)
222     FORMAT(10X,3A20)
        RETURN
        END


C INPUT THE INFORMATION INDICATED BY USER
        SUBROUTINE INPTINF
        PARAMETER(MAXGF=20,MSTRC=100,MAXSPD=2000,MNLT=10,MNLPT=500,
     $            NSEC=4,MAXM=10,MNDM=10,NCRT=2)
        INTEGER IDST(MSTRC)
        REAL XCST(MSTRC),YCST(MSTRC)
        REAL PXAC(MAXSPD),PYAC(MAXSPD)
        CHARACTER*15 DMENT(MAXM),UNAVLM
        CHARACTER*20 ASFD,ASCRT(NCRT),VPTCS
        CHARACTER*10 SECID,AVLSEC(NSEC),DSECID(NSEC)
        CHARACTER*20 GFITEM(MAXGF),REQUST,UPIT
        CHARACTER*80 TEXTPL(MNLPT)
        LOGICAL PROMP,NEWSTRC
        COMMON /PSCREN/ ORX,ORY,XALEN,YALEN
        COMMON /AVGENINF/ NGFIT,GFITEM
        COMMON /VIEWPT/ SVXLEN,SVYLEN,SVXST,SVYST,
     $                  BVXLEN,BVYLEN,BVXST,BVYST
        COMMON /VIEWSIZE/ ASCRT,VPTCS
        COMMON /DSHVMW/ DSBFV,DSBFH,WIDEMF,FINT,XSMF,YSMF
        COMMON /PSPCHART/ XSSPC,XESPC,YSSPC,YESPC
        COMMON /PRSPEC/ XSRSP,YSRSP,XERSP,YERSP
        COMMON /SECTID/ DSECID
        COMMON /SUBMENU1/ NDM,DMENT
        COMMON /UVSCALE/WAVS,WAVE,APSS,APSE
        DATA UNAVLM/'Not Available'/
        DATA DSECID/'STRUCTURE','IRSPECTR','UVSPECTR','LONTEXT'/
C PARAMETERS FOR STORING DATA
```

```
        NST=3
        NIDN=6
        NSSC=3
        NSP=4
        HEITMF=NDM*FINT
        ICF=8
        ICT=6
C DISPLAY THE WORKING MENU
        OPEN (UNIT=12,STATUS='OLD',ACCESS='APPEND',
     $        FILE='MFILE.AIS')
        CALL GENDIS
100     CALL CLEAR(BVXST,SVYST,BVXST+WIDEMF,SVYST+SVYLEN)
        CALL MENUDR(XSMF,YSMF,WIDEMF,HEITMF,FINT,ICF,
     $              NDM,DMENT,ICT)
        CALL MENUID(NDM,IDK)
        CALL MENUCL
        IDM=IDK-1
        GOTO(300,301,302,303,304,304),IDK
300     CALL SETVW
        CALL GENDIS
        GOTO 100
301     NEWSTRC=.TRUE.
        CALL CHEMSDC(NEWSTRC,NSTRC,IDST,XCST,YCST)
        CALL CLEAR(ORX,ORY,XALEN,YALEN)
        CALL MAINPAD
        WRITE(12,111) DSECID(IDM),NSTRC
        DO IS=1,NSTRC,NSSC
          IF ((NSTRC-IS) .LT. NSSC) THEN
            JE=NSTRC
          ELSE
            JE=IS+NSSC-1
          ENDIF
          WRITE(12,222) (IDST(JS),XCST(JS),YCST(JS),JS=IS,JE)
        ENDDO
        GOTO 100
302     CALL IRSCAN(NSPT,PXAC,PYAC)
        WRITE(12,111) DSECID(IDM),NSPT
        DO I=1,NSPT,NSP
          IF ((NSPT-I) .LT. NSP) THEN
            JEN=NSPT
          ELSE
            JEN=I+NSP-1
          ENDIF
          WRITE(12,333) (PXAC(J),PYAC(J),J=I,JEN)
        ENDDO
        IF (VPTCS .EQ. ASCRT(1) .OR. VPTCS .EQ. ASCRT(2))THEN
          CALL CLEAR(XSRSP,YSRSP,XERSP,YERSP)
        ENDIF
        GOTO 100
303     CALL UVSCAN(NUVSP,PXAC,PYAC)
        WRITE(12,111) DSECID(IDM),NUVSP
        WRITE(12,444) WAVS,WAVE,APSS,APSE
        DO I=1,NUVSP,NSP
          IF ((NUVSP-I) .LT. NSP) THEN
            JEN=NUVSP
          ELSE
            JEN=I+NSP-1
          ENDIF
          WRITE(12,333) (PXAC(J),PYAC(J),J=I,JEN)
        ENDDO
        IF (VPTCS .EQ. ASCRT(1) .OR. VPTCS .EQ. ASCRT(2)) THEN
          CALL CLEAR(XSRSP,YSRSP,XERSP,YERSP)
        ENDIF
        GOTO 100
304     CLOSE(UNIT=12)
111     FORMAT(10X,A10,I)
```

```
222     FORMAT(10X,3(I3,2F8.3))
333     FORMAT(10X,8F8.3)
444     FORMAT(10X,2F5.1,2F5.3)
        RETURN
        END

C THIS ROUTINE SORTE THE STRINGS
        SUBROUTINE SSORT(NS,INS,IDOS)
        CHARACTER*20 INS(NS),TEMP
        INTEGER FIRST,PASS,IDOS(NS)
        LOGICAL NOEXCH
        NOEXCH = .FALSE.
        PASS = 1
        DO WHILE (.NOT. NOEXCH)
           NOEXCH = .TRUE.
           DO FIRST = 1,NS-1
              IF (INS(FIRST) .GT. INS(FIRST+1)) THEN
                 TEMP = INS(FIRST)
                 INS(FIRST) = INS(FIRST+1)
                 INS(FIRST+1) = TEMP
                 IDT = IDOS(FIRST)
                 IDOS(FIRST) = IDOS(FIRST+1)
                 IDOS(FIRST+1) = IDT
                 NOEXCH = .FALSE.
               ENDIF
           ENDDO
           PASS = PASS +1
        ENDDO
        RETURN
        END

C DELTE AN ENTRY FROM THE BASE, ACTUALLY DELETES ENTRIES BOTH
C IN NAMELIS AND MFILE
        SUBROUTINE DELENTRY
        PARAMETER(MAXGF=20,MSTRC=100,MAXSPD=2000,MNLT=10,MNLPT=500,
     $            NSEC=3,NMESG=2)
        INTEGER IDST(MSTRC)
        REAL XCST(MSTRC),YCST(MSTRC)
        REAL PXAC(MAXSPD),PYAC(MAXSPD)
        CHARACTER*10 SECID,AVLSEC(NSEC),DSECID(NSEC)
        CHARACTER*20 ENTRYNM,MESSG(NMESG)
        CHARACTER*20 GFITEM(MAXGF),REQUST,UPIT
        CHARACTER*80 TEXTPL(MNLPT)
        LOGICAL PROMP
        DATA MESSG /'Please enter name','Not found,try again'/
        NIT=6
        NST=3
        PROMP=.TRUE.
        CALL PROMPT(MESSG(1),ENTRYNM,PROMP)
        CALL STR$UPCASE(ENTRYNM,ENTRYNM)
C DELETE THE ENTRY FROM NAMELIS FILE
        OPEN (UNIT=13,STATUS='NEW',FILE='TEMP.AIS')
        OPEN (UNIT=12,STATUS='OLD',FILE='NAMELIS.AIS')
          DO WHILE(.TRUE.)
100         DO I=1,NIT,NST
              READ(12,111,END=200)(GFITEM(J),J=I,I+NST-1)
            ENDDO
            DO I=1,NIT
              CALL STR$UPCASE(UPIT,GFITEM(I))
              IF (UPIT .EQ. ENTRYNM) THEN
                GOTO 100
              ENDIF
            ENDDO
            DO J=1,NIT,NST
```

```
       WRITE(13,111)(GFITEM(K),K=J,J+NST-1)
      ENDDO
     ENDDO
200 CLOSE(UNIT=12)
    CLOSE(UNIT=13)
    CALL LIB$SPAWN('RENAME TEMP.AIS NAMELIS.AIS')
111 FORMAT(10X,3A20)
    RETURN
    END

C SEARCH AN ENTRY IN THE BASE
    SUBROUTINE SEARCH
    PARAMETER(MAXNL=20)
    CHARACTER*20 REQUST,ITEMS(MAXNL),UPIT
    CHARACTER*20 DIAMT(3),ASFD
    EXTERNAL STR$UPCASE
    LOGICAL PROMP
    DATA DIAMT /'Please enter name','Request found!',
  $               'Not found,try again?'/
    NIT=6
    NSL=3
10  PROMP=.TRUE.
    CALL PROMPT(DIAMT(1),REQUST,PROMP)
C CHECK NAMELIS FILE TO SEE IF THE REQUEST IS IN THE BASE
C IF YES THEN OPEN THE MAIN FILE TO LOAD THE INFORMATION
    OPEN (UNIT=12,STATUS='OLD',FILE='NAMELIS.AIS')
100  DO I=1,NIT,NSL
      READ(12,111,END=200)(ITEMS(J),J=I,I+NSL-1)
     ENDDO
     DO I=1,NIT
      CALL STR$UPCASE(UPIT,ITEMS(I))
      CALL STR$UPCASE(REQUST,REQUST)
C LOAD THE CONTENTS OF THE REQUEST FOUND IN BASE
      IF (UPIT .EQ. REQUST) THEN
       PROMP=.FALSE.
       CALL PROMPT(DIAMT(2),ASFD,PROMP)
       CALL LOADINF(REQUST)
       CALL DISINF
       RETURN
      ENDIF
     ENDDO
     GOTO 100
200 CLOSE(UNIT=12)
C DISPLAY THE UNAVAILABLE MESSAGE
    PROMP=.FALSE.
    CALL PROMPT(DIAMT(3),ASFD,PROMP)
    IF (ASFD .EQ. 'YES' .OR. ASFD .EQ. 'Y') THEN
     GOTO 10
    ENDIF
111 FORMAT(10X,3A20)
    RETURN
    END

C LOAD THE CONTENTS OF A REQUESTED ENTRY
    SUBROUTINE LOADINF(REQUST)
    PARAMETER(MAXGF=20,MSTRC=100,MAXSPD=2000,MNLT=10,MNLPT=500,
  $              NSEC=4)
    INTEGER IDST(MSTRC)
    REAL XCST(MSTRC),YCST(MSTRC)
    REAL PXAC(MAXSPD),PYAC(MAXSPD),PXUV(MAXSPD),PYUV(MAXSPD)
    CHARACTER*10 SECID,AVLSEC(NSEC),DSECID(NSEC)
    CHARACTER*20 GFITEM(MAXGF),REQUST,UPIT
    CHARACTER*80 TEXTPL(MNLPT)
    LOGICAL SCAV(NSEC),SECAV,ENDSEC
```

```
        COMMON /AVLOG/ SCAV,SECAV
        COMMON /AVGENINF/ NGFIT,GFITEM
        COMMON /PARSTRC/NSTRC,IDST,XCST,YCST
        COMMON /PARSPEC/NSPT,PXAC,PYAC
        COMMON /PUVSPEC/NUVSP,PXUV,PYUV
        COMMON /PARTEXT/NTEXT,TEXTPL
        COMMON /SECTID/ DSECID
        COMMON /UVSCALE/ WAVS,WAVE,APSS,APSE
C READ THE NUMBER OF GENERAL INF. AND THE CONTENTS AFTERWARS
        NST=3
        NIDN=6
        NSSC=3
        NSP=4
        OPEN (UNIT=12,STATUS='OLD',FILE='MFILE.AIS')
10        READ(12,111,END=100) NGFIT
20        DO I=1,NGFIT,NST
            IF ((NGFIT-I) .LT. NST) THEN
              JE=NGFIT
            ELSE
              JE=I+NST-1
            ENDIF
            READ(12,222)(GFITEM(J),J=I,JE)
          ENDDO
          ENDSEC=.FALSE.
50        READ(12,333,ERR=100,END=100) SECID,NCONT
          IF (SECID .EQ. DSECID(1)) THEN
            SCAV(1)=.TRUE.
            NSTRC=NCONT
            DO IS=1,NSTRC,NSSC
              IF ((NSTRC-IS) .LT. NSSC) THEN
                JE=NSTRC
              ELSE
                JE=IS+NSSC-1
              ENDIF
              READ(12,444)(IDST(J),XCST(J),YCST(J),J=IS,JE)
            ENDDO
          ELSE IF(SECID .EQ. DSECID(2)) THEN
            SCAV(2)=.TRUE.
            NSPT=NCONT
            DO IP=1,NSPT,NSP
              IF ((NSPT-IP) .LT. NSP) THEN
                JE=NSPT
              ELSE
                JE=IP+NSP-1
              ENDIF
              READ(12,555)(PXAC(J),PYAC(J),J=IP,JE)
            ENDDO
          ELSE IF(SECID .EQ. DSECID(3)) THEN
            SCAV(3)=.TRUE.
            NUVSP=NCONT
            READ(12,666) WAVS,WAVE,APSS,APSE
            DO IP=1,NUVSP,NSP
              IF ((NUVSP-IP) .LT. NSP) THEN
                JE=NUVSP
              ELSE
                JE=IP+NSP-1
              ENDIF
              READ(12,555)(PXUV(J),PYUV(J),J=IP,JE)
            ENDDO
          ELSE IF(SECID .EQ. DSECID(4)) THEN
            SCAV(4)=.TRUE.
            NTEXT=NCONT
            DO IT=1,NTEXT
              READ(12,777)TEXTPL(IT)
            ENDDO
          ELSE
```

```
            ENDSEC=.TRUE.
            NGFIT=13
            GOTO 100
          ENDIF
          GOTO 50
100       DO I=1,NIDN
            CALL STR$UPCASE(UPIT,GFITEM(I))
            IF (UPIT .EQ. REQUST) THEN
              SECAV=.TRUE.
              GOTO 200
            ENDIF
          ENDDO
C SET LOGICAL VALUE FOR ALL SECTIONS TO FALSE
          DO I=1,NSEC
            SCAV(I)=.FALSE.
          ENDDO
          IF (ENDSEC) THEN
            GOTO 20
          ELSE
            GOTO 10
          ENDIF
C READ THE REST OF THE INFORMATION
200     CLOSE(UNIT=12)
111     FORMAT(10X,I)
222     FORMAT(10X,3A20)
333     FORMAT(10X,A10,I)
444     FORMAT(10X,3(I3,2F8.3))
555     FORMAT(10X,8F8.3)
666     FORMAT(10X,2F5.1,2F5.3)
777     FORMAT(10X,A)
        RETURN
        END
```

```

C DISPLAY ALL AVAILABLE INFORMATION ON REQUEST
        SUBROUTINE DISINF
        PARAMETER(MAXGF=20,MSTRC=100,MAXSPD=2000,MNLT=10,MNLPT=500,
     $            NSEC=4,MAXM=10,NMESG=2,NCRT=2)
        INTEGER IDST(MSTRC)
        REAL XCST(MSTRC),YCST(MSTRC)
        REAL PXAC(MAXSPD),PYAC(MAXSPD),PXUV(MAXSPD),PYUV(MAXSPD)
        CHARACTER*10 SECID,AVLSEC(NSEC),DSECID(NSEC)
        CHARACTER*15 DMENT(MAXM)
        CHARACTER*20 ASFD,ASCRT(NCRT),VPTCS
        CHARACTER*20 GFITEM(MAXGF),REQUST,UPIT,MESSG(NMESG)
        CHARACTER*80 TEXTPL(MNLPT)
        LOGICAL SCAV(NSEC),SECAV,PROMP
        COMMON /VIEWPT/ SVXLEN,SVYLEN,SVXST,SVYST,
     $                  BVXLEN,BVYLEN,BVXST,BVYST
        COMMON /VIEWSIZE/ ASCRT,VPTCS
        COMMON /DSHVMW/ DSBFV,DSBFH,WIDEMF,FINT,XSMF,YSMF
        COMMON /AVLOG/ SCAV,SECAV
        COMMON /PARSTRC/NSTRC,IDST,XCST,YCST
        COMMON /PARSPEC/NSPT,PXAC,PYAC
        COMMON /PUVSPEC/NUVSP,PXUV,PYUV
        COMMON /PARTEXT/NTEXT,TEXTPL
        COMMON /PRSPEC/ XSRSP,YSRSP,XERSP,YERSP
        COMMON /AVGENINF/ NITEM, GFITEM
        COMMON /SUBMENU1/ NDM,DMENT
        COMMON /UVSCALE/WAVS,WAVE,APSS,APSE
        DATA (MESSG(I),I=1,NMESG)/'Not Available','[RET] continue!'/
C IF ANY SECTIONS AVAILABLE THEN DISPLAY GENERAL INF.
C AND THE MENU FOR THE REST
        HEITMF=NDM*FINT
        ICF=8
        ICT=6
        CALL GENDIS
100     IF (SECAV) THEN
          CALL MENUDR(XSMF,YSMF,WIDEMF,HEITMF,FINT,ICF,
     $                NDM,DMENT,ICT)
          CALL MENUID(NDM,IDK)
          CALL MENUCL
          IDM=IDK-1
          GOTO(300,301,302,303,304,304),IDK
C SET VIEWPORT
300     CALL SETVW
        CALL GENDIS
        GOTO 100
C DISPLAY CHEMICAL STRUCTURE
301     IF (SCAV(IDM)) THEN
          NEWSTRC=.FALSE.
          CALL CHEMSDC(NEWSTRC,NSTRC,IDST,XCST,YCST)
        ELSE
          PROMP=.FALSE.
          CALL PROMPT(MESSG(1),ASFD,PROMP)
        ENDIF
        GOTO 100
C DISPLAY SPECTRUM
302     IF (SCAV(IDM)) THEN
          CALL CLEAR(XSRSP,YSRSP,XERSP,YERSP)
          CALL IRDISP(NSPT,PXAC,PYAC,8)
          PROMP=.FALSE.
          CALL PROMPT(MESSG(2),ASFD,PROMP)
          IF (VPTCS .EQ. ASCRT(1) .OR. VPTCS .EQ. ASCRT(2)) THEN
            CALL CLEAR(XSRSP,YSRSP,XERSP,YERSP)
          ENDIF
        ELSE
          PROMP=.FALSE.
          CALL PROMPT(MESSG(1),ASFD,PROMP)
        ENDIF
        GOTO 100
303     IF (SCAV(IDM)) THEN
          CALL CLEAR(XSRSP,YSRSP,XERSP,YERSP)
          CALL UVDISP(NUVSP,PXUV,PYUV,8)
          PROMP=.FALSE.
          CALL PROMPT(MESSG(2),ASFD,PROMP)
          IF (VPTCS .EQ. ASCRT(1) .OR. VPTCS .EQ. ASCRT(2)) THEN
            CALL CLEAR(XSRSP,YSRSP,XERSP,YERSP)
          ENDIF
        ELSE
          PROMP=.FALSE.
          CALL PROMPT(MESSG(1),ASFD,PROMP)
        ENDIF
        GOTO 100
C OTHER TEXT INFORMATION
304       RETURN
        ELSE
          PROMP=.FALSE.
          CALL PROMPT(MESSG(1),ASFD,PROMP)
          RETURN
        ENDIF
        END


C UTILITY ROUTINE FOR INFORMATION RETRIEVAL
        SUBROUTINE UTILTY
        PARAMETER(NDM=2,NCRT=2)
        CHARACTER*15 DMENT(NDM)
        CHARACTER*20 ASFD,ASCRT(NCRT),VPTCS
        COMMON /VIEWPT/ SVXLEN,SVYLEN,SVXST,SVYST,
     $                  BVXLEN,BVYLEN,BVXST,BVYST
        COMMON /VIEWSIZE/ ASCRT,VPTCS
        COMMON /DSHVMW/ DSBFV,DSBFH,WIDEMF,FINT,XSMF,YSMF
        DATA DMENT/'Compare Spectra','Quit'/
C IF ANY SECTIONS AVAILABLE THEN DISPLAY GENERAL INF.
C AND THE MENU FOR THE REST
        HEITMF=NDM*FINT
        ICF=8
        ICT=6
100     CALL MENUDR(XSMF,YSMF,WIDEMF,HEITMF,FINT,ICF,
     $              NDM,DMENT,ICT)
        CALL MENUID(NDM,IDM)
        CALL MENUCL
        GOTO(300,301),IDM
300     CALL COMPSPECT
        GOTO 100
301     RETURN
        END

C COMPARE SPECTRA OF SEVERAL SAMPLES
        SUBROUTINE COMPSPECT
        PARAMETER(MAXGF=20,MSTRC=100,MAXSPD=2000,MNLT=10,MNLPT=500,
     $            NSEC=4,MNCOMP=10)
        INTEGER IDST(MSTRC),NIRSP(MNCOMP),NUVSP(MNCOMP)
        REAL XCST(MSTRC),YCST(MSTRC)
        REAL PXIR(MAXSPD),PYIR(MAXSPD),PXUV(MAXSPD),PYUV(MAXSPD)
        REAL IRXCP(MNCOMP,MAXSPD),IRYCP(MNCOMP,MAXSPD)
        REAL UVXCP(MNCOMP,MAXSPD),UVYCP(MNCOMP,MAXSPD)
        REAL WVS(MNCOMP),WVE(MNCOMP),APS(MNCOMP),APE(MNCOMP)
        CHARACTER*10 SECID,DSECID(NSEC)
        CHARACTER*20 GFITEM(MAXGF),REQUST(MNCOMP),UPIT
        CHARACTER*20 MESSG(MNCOMP),ASFD
        CHARACTER*80 TEXTPL(MNLPT)
        LOGICAL SCAV(MNCOMP,NSEC),SECAV(MNCOMP),ENDSEC(MNCOMP),PROMP
        COMMON /SECTID/ DSECID
        COMMON /NAMESTOCOMP/ NBNMS,REQUST
        COMMON /NIRUV/NIRSP,NUVSP
        COMMON /IRCOMP/ IRXCP,IRYCP
        COMMON /UVCOMP/ UVXCP,UVYCP,WVS,WVE,APS,APE
        DATA (MESSG(I),I=1,2)/'How many samples','Enter name'/
C READ IN NAMES OF SAMPLES TO COMPARE
        PROMP=.TRUE.
        CALL PROMPT(MESSG(1),ASFD,PROMP)
        READ(UNIT=ASFD,FMT='(I2)') NBNMS
        CALL READNAME(MESSG(2))
C READ THE NUMBER OF GENERAL INF. AND THE CONTENTS AFTERWARS
        NST=3
        NIDN=6
        NSSC=3
        NSP=4
        OPEN (UNIT=12,STATUS='OLD',FILE='MFILE.AIS')
        DO 200 IDN=1,NBNMS
          REWIND(UNIT=12)
10        READ(12,111,END=100) NGFIT
20        DO I=1,NGFIT,NST
            IF ((NGFIT-I) .LT. NST) THEN
              JE=NGFIT
            ELSE
              JE=I+NST-1
            ENDIF
            READ(12,222)(GFITEM(J),J=I,JE)
          ENDDO
          ENDSEC(IDN)=.FALSE.
50        READ(12,333,ERR=100,END=100) SECID,NCONT
          IF (SECID .EQ. DSECID(1)) THEN
            SCAV(IDN,1)=.TRUE.
            NSTRC=NCONT
            DO IS=1,NSTRC,NSSC
              IF ((NSTRC-IS) .LT. NSSC) THEN
                JE=NSTRC
              ELSE
                JE=IS+NSSC-1
              ENDIF
              READ(12,444)(IDST(J),XCST(J),YCST(J),J=IS,JE)
            ENDDO
          ELSE IF(SECID .EQ. DSECID(2)) THEN
            SCAV(IDN,2)=.TRUE.
            NIRSP(IDN)=NCONT
            DO IP=1,NIRSP(IDN),NSP
              IF ((NIRSP(IDN)-IP) .LT. NSP) THEN
                JE=NIRSP(IDN)
              ELSE
                JE=IP+NSP-1
              ENDIF
              READ(12,555)(PXIR(J),PYIR(J),J=IP,JE)
            ENDDO
          ELSE IF(SECID .EQ. DSECID(3)) THEN
            SCAV(IDN,3)=.TRUE.
            NUVSP(IDN)=NCONT
            READ(12,666) WAVS,WAVE,APSS,APSE
            DO IP=1,NUVSP(IDN),NSP
              IF ((NUVSP(IDN)-IP) .LT. NSP) THEN
                JE=NUVSP(IDN)
              ELSE
                JE=IP+NSP-1
              ENDIF
              READ(12,555)(PXUV(J),PYUV(J),J=IP,JE)
            ENDDO
          ELSE IF(SECID .EQ. DSECID(4)) THEN
            SCAV(IDN,4)=.TRUE.
            NTEXT=NCONT
            DO IT=1,NTEXT
              READ(12,777)TEXTPL(IT)
            ENDDO
          ELSE
            ENDSEC(IDN)=.TRUE.
            NGFIT=13
            GOTO 100
          ENDIF
          GOTO 50
100       DO I=1,NIDN
            CALL STR$UPCASE(UPIT,GFITEM(I))
            CALL STR$UPCASE(REQUST(IDN),REQUST(IDN))
            IF (UPIT .EQ. REQUST(IDN)) THEN
              SECAV(IDN)=.TRUE.
              DO IR=1,NIRSP(IDN)
                IRXCP(IDN,IR)=PXIR(IR)
                IRYCP(IDN,IR)=PYIR(IR)
              ENDDO
              WVS(IDN)=WAVS
              WVE(IDN)=WAVE
              APS(IDN)=APSS
              APE(IDN)=APSE
              DO IUV=1,NUVSP(IDN)
                UVXCP(IDN,IUV)=PXUV(IUV)
                UVYCP(IDN,IUV)=PYUV(IUV)
              ENDDO
              GOTO 200
            ENDIF
          ENDDO
C SET LOGICAL VALUE FOR ALL SECTIONS TO FALSE
          DO I=1,NSEC
            SCAV(IDN+1,I)=.FALSE.
          ENDDO
          IF (ENDSEC(IDN)) THEN
            GOTO 20
          ELSE
            GOTO 10
          ENDIF
C READ THE REST OF THE INFORMATION
200     CONTINUE
        CLOSE(UNIT=12)
        CALL DISCOMP
111     FORMAT(10X,I)
222     FORMAT(10X,3A20)
333     FORMAT(10X,A10,I)
444     FORMAT(10X,3(I3,2F8.3))
555     FORMAT(10X,8F8.3)
666     FORMAT(10X,2F5.1,2F5.3)
777     FORMAT(10X,A)
        RETURN
        END

C DISPLAY THE COMPARISION OF IR AND UV SPECTRA
        SUBROUTINE DISCOMP
        PARAMETER(MAXGF=20,MSTRC=100,MAXSPD=2000,MNLT=10,MNLPT=500,
     $            NSEC=4,MAXM=10,NMESG=2,NCRT=2,MNCOMP=10,NDM=4)
        INTEGER NIRSP(MNCOMP),NUVSP(MNCOMP)
        REAL IRXCP(MNCOMP,MAXSPD),IRYCP(MNCOMP,MAXSPD)
        REAL UVXCP(MNCOMP,MAXSPD),UVYCP(MNCOMP,MAXSPD)
        REAL XDS(MAXSPD),YDS(MAXSPD)
        REAL WVS(MNCOMP),WVE(MNCOMP),APS(MNCOMP),APE(MNCOMP)
        CHARACTER*20 REQUST(MNCOMP)
        CHARACTER*20 MESSG(MNCOMP)
        LOGICAL PROMP,COMPAR
        CHARACTER*10 SECID,AVLSEC(NSEC),DSECID(NSEC)
        CHARACTER*15 DMENT(MAXM)
        CHARACTER*20 ASFD,ASCRT(NCRT),VPTCS
        COMMON /VIEWPT/ SVXLEN,SVYLEN,SVXST,SVYST,
     $                  BVXLEN,BVYLEN,BVXST,BVYST
        COMMON /VIEWSIZE/ ASCRT,VPTCS
        COMMON /DSHVMW/ DSBFV,DSBFH,WIDEMF,FINT,XSMF,YSMF
        COMMON /PRSPEC/ XSRSP,YSRSP,XERSP,YERSP
        COMMON /COMPARISON/ COMPAR,IDN
        COMMON /NAMESTOCOMP/ NBNMS,REQUST
        COMMON /NIRUV/ NIRSP,NUVSP
        COMMON /IRCOMP/ IRXCP,IRYCP
        COMMON /UVCOMP/ UVXCP,UVYCP,WVS,WVE,APS,APE
        COMMON /UVSCALE/ WAVS,WAVE,APSS,APSE
        DATA (MESSG(I),I=1,NMESG)/'Not Available','[RET] continue!'/
        DATA (DMENT(I),I=1,NDM)/'Set View','Infrared Spect',
     $                          'UV Spectra','Quit'/
        COMPAR=.TRUE.
C IF ANY SECTIONS AVAILABLE THEN DISPLAY GENERAL INF.
C AND THE MENU FOR THE REST
        HEITMF=NDM*FINT
        ICF=8
        ICT=6
100     CALL MENUDR(XSMF,YSMF,WIDEMF,HEITMF,FINT,ICF,
     $              NDM,DMENT,ICT)
        CALL MENUID(NDM,IDK)
        CALL MENUCL
        IDM=IDK-1
        GOTO(300,301,302,303),IDK
C SET VIEWPORT
300     CALL SETVW
        CALL GENDIS
        GOTO 100
```

```
C DISPLAY SPECTRUM
301     CALL CLEAR(XSRSP,YSRSP,XERSP,YERSP)
        DO IDN=1,NBNMS
          DO J=1,NIRSP(IDN)
            XDS(J)=IRXCP(IDN,J)
            YDS(J)=IRYCP(IDN,J)
          ENDDO
          CALL IRDISP(NIRSP(IDN),XDS,YDS,IDN+1)
        ENDDO
        PROMP=.FALSE.
        CALL PROMPT(MESSG(2),ASFD,PROMP)
        IF (VPTCS .EQ. ASCRT(1) .OR. VPTCS .EQ. ASCRT(2)) THEN
          CALL CLEAR(XSRSP,YSRSP,XERSP,YERSP)
        ENDIF
        GOTO 100
302     CALL CLEAR(XSRSP,YSRSP,XERSP,YERSP)
        DO IDN=1,NBNMS
          WAVS=WVS(IDN)
          WAVE=WVE(IDN)
          APSS=APS(IDN)
          APSE=APE(IDN)
          DO J=1,NUVSP(IDN)
            XDS(J)=UVXCP(IDN,J)
            YDS(J)=UVYCP(IDN,J)
          ENDDO
          CALL UVDISP(NUVSP(IDN),XDS,YDS,IDN+1)
        ENDDO
        PROMP=.FALSE.
        CALL PROMPT(MESSG(2),ASFD,PROMP)
        IF (VPTCS .EQ. ASCRT(1) .OR. VPTCS .EQ. ASCRT(2)) THEN
          CALL CLEAR(XSRSP,YSRSP,XERSP,YERSP)
        ENDIF
        GOTO 100
C OTHER TEXT INFORMATION
303     COMPAR=.FALSE.
        RETURN
        END

C ROUTINE TO READ IN A LIST OF NAMES
        SUBROUTINE READNAME(TEXT)
        PARAMETER(MNAME=10)
        INTEGER SMGID,PBD,KBD,VROW,VCOL
        CHARACTER*20 IDNAME
        CHARACTER*20 NAMES(MNAME)
        CHARACTER*20 TEXT
        CHARACTER*25 DISPRP
        LOGICAL PROMP
        COMMON /DSHVMW/ DSBFH,DSBFV,WIDEM,FINT
        COMMON /DIABOX/ DFXS,DFYS,WIDEPR,DFHEIT
        COMMON /SMGPAR/ SMGID,PBD,KBD,VROW,VCOL
        COMMON /DFTCHAR/ DCHWD,DCHHT
        COMMON /NAMESTOCOMP/ NINPT,NAMES
        CALL CHAENQ(ITO,WDO,HTO,NIO,AIO,ANO)
        CHWD=DCHWD
        CHHT=DCHHT
        CALL CHASIZ(CHWD,CHHT)
        WIDEPR=80.0
        BXS=DFXS+DSBFH
        BYS=DFYS+DSBFV
        BXE=DFXS+WIDEPR-DSBFH
        BYE=DFYS+DFHEIT-DSBFV
        VROW=24
        VCOL=31
        DO IN=1,NINPT
          WRITE(UNIT=IDNAME,FMT='(I2)')IN
          CALL LINCOL(6)
          CALL MOVTO2(DFXS+WIDEPR,DFYS)
          CALL LINBY2(0.0,DFHEIT)
          CALL LINCOL(4)
          CALL STR$CONCAT(DISPRP,TEXT,IDNAME)
          CALL MOVTO2(BXS,BYS+DSBFV)
          CALL CHASTR(DISPRP)
          CALL CHAMOD
          CALL SMG$SET_CURSOR_ABS(SMGID,VROW,VCOL)
          READ(5,111) NAMES(IN)
          CALL DIATCL
          CALL CLEAR(BXS,BYS,BXE,BYE)
        ENDDO
        CALL CHASIZ(WDO,HTO)
111     FORMAT(A)
        RETURN
        END
```

```

C ROUTINE TO DISPLAY THE GENERAL INFORMATION PAD AND CONTENTS
        SUBROUTINE GENDIS
        PARAMETER (MAXGF=20)
        REAL XPG(MAXGF),YPG(MAXGF)
        CHARACTER*20 GMITEM(MAXGF)
        COMMON /AVGENINF/ NITEM,GMITEM
        COMMON /POSITEM/ XPG,YPG
        CALL GENPAD
        DO I=1,NITEM
          CALL LINCOL(5)
          CALL MOVTO2(XPG(I),YPG(I))
          CALL CHASTR(GMITEM(I))
        ENDDO
        RETURN
        END
```

```

C DRAW HORIZONTOL LINE
        SUBROUTINE HOLINE(XSPOS,YSPOS,XLEN)
        CALL MOVTO2(XSPOS,YSPOS)
        CALL LINBY2(XLEN,0.0)
        RETURN
        END
```

```

C DRAW VERTICAL LINE
        SUBROUTINE VELINE(XSPOS,YSPOS,YLEN)
        CALL MOVTO2(XSPOS,YSPOS)
        CALL LINBY2(0.0,YLEN)
        RETURN
        END
```

```

C*****************************************************
C***      ROUTINE TO DRAW A FRAME                  ***
        SUBROUTINE FEFRAME(XSPOS,YSPOS,FWIDE,FHEIT,FCOL)
        INTEGER FCOL
        CALL LINCOL(FCOL)
        CALL MOVTO2(XSPOS,YSPOS)
        CALL LINBY2(0.0,FHEIT)
        CALL LINBY2(FWIDE,0.0)
        CALL LINBY2(0.0,-FHEIT)
        CALL LINBY2(-FWIDE,0.0)
        RETURN
        END
```

```

C***************************************************
C***    DRAW HORIZONTAL INTERVALED FRAME         ***
        SUBROUTINE HSFRAME(XSPOS,YSPOS,FWIDE,FHEIT,FINTV)
        NFR=INT(ABS(FHEIT/FINTV))
        XS=XSPOS
        YS=YSPOS
        DO I=1,NFR
          YS=YS+FINTV
          CALL MOVTO2(XS,YS)
          CALL LINBY2(FWIDE,0.0)
        ENDDO
        RETURN
        END
```

```

C***************************************************
C***    DRAW VERTICAL INTERVALED FRAME           ***
        SUBROUTINE VSFRAME(XSPOS,YSPOS,FWIDE,FHEIT,FINTV)
        XS=XSPOS
        YS=YSPOS
        NFR=INT((FWIDE/FINTV))
        DO I=1,NFR
          XS=XS+FINTV
          CALL MOVTO2(XS,YS)
          CALL LINBY2(0.0,FHEIT)
        ENDDO
        RETURN
        END
```

```


C***************************************************
C****     DRAW THE LARGE FRAME                  ****
        SUBROUTINE LGFRAME
C***************************************************
        COMMON /PSCREN/XST,YST,XLEN,YLEN
        ICOL=8
        CALL FEFRAME(XST,YST,XLEN,YLEN,ICOL)
        RETURN
        END
```

```

C SUBROUTINE TO DISPLAY AND CONSTRUCTE CHEMICAL STRUCTURES
        SUBROUTINE CHEMSDC(NEWSTRC,NCSG,ID,XPS,YPS)
        PARAMETER(NCSTC=200,NCRITV=2)
        INTEGER ID(NCSTC)
        REAL XPS(NCSTC),YPS(NCSTC)
        CHARACTER*20 ASCRIT(NCRITV),VIEWCS
        LOGICAL NEWSTRC
        COMMON /PSCREN/ ORX,ORY,XALEN,YALEN
        COMMON /VIEWPT/ SVXLEN,SVYLEN,SVXST,SVYST,
     $                  BVXLEN,BVYLEN,BVXST,BVYST
        COMMON /VIEWSIZE/ ASCRIT,VIEWCS
        COMMON /DSHVMW/ DSBFH,DSBFV,WIDEMF,FINT,XSMF,YSMF,TFHEIT
        COMMON /PARGENPAD/ HGENPAD
        COMMON /BSTCSIZ/HEIT,WIDS,R
C DEFINE KEYS
C F...... FETCH THE OBJECT
C P...... PLACE IT
C M...... MOVE THE OBJECT CHOSEN
C D...... DELETE THE OBJECT
C C...... CLEAN OUT EVERYTHING HAS BEING DRAWN
C Q...... QUIT CONSTRUCTION OPERATRION
C E...... END CONSTRUCTION
        CALL CURSTR('FPMDCQE')
```

```
C GLOBAL VARIABLES FOR THE SIZE OF BASIC STRUCTURES
        HEIT=12.0
        WIDS=10.0
        R=3.0
C DEFINE A REGION FOR THE DISPLAY OF STRUCTURE IN AIS
        IF (VIEWCS .EQ. ASCRIT(1) .OR. VIEWCS .EQ. ASCRIT(2)) THEN
          XSDR=BVXST
          XEDR=BVXST+BVXLEN
          YSDR=SVYST
          YEDR=SVYST+SVYLEN
        ELSE
          XSDR=SVXST+DSBFV
          XEDR=SVXST+SVXLEN-DSBFV
          YSDR=SVYST+HGENPAD+2.0*DSBFV+TFHEIT
          YEDR=SVYST+SVYLEN-DSBFV
        ENDIF
        CALL CLEAR(XSDR,YSDR,XEDR,YEDR)
        XCDR=XSDR+(XEDR-XSDR)/2.0
        YCDR=YSDR+(YEDR-YSDR)/2.0
        IF (NEWSTRC) THEN
           CALL CLEAR(ORX,ORY,XALEN,YALEN)
           CALL CHEMFORM(NCSG,ID,XPS,YPS)
        ELSE
C DISPLAY THE STRUCTURE IN A DEFINED REGION
           CALL MOVTO2(XSDR+10.0,YSDR+DSBFV)
           CALL LINCOL(5)
           CALL CHASTR('Chemical Structure*.')
           CALL LINCOL(6)
           CALL MAXMIN(NCSG,XPS,YPS,XMAX,XMIN,YMAX,YMIN)
           XCST=XMIN+(XMAX-XMIN)/2.0
           YCST=YMIN+(YMAX-YMIN)/2.0
           DO I=1,NCSG
              XCDS=XPS(I)+(XCDR-XCST)
              YCDS=YPS(I)+(YCDR-YCST)
              CALL DRBSTC(ID(I),XCDS,YCDS)
           ENDDO
        ENDIF
        RETURN
        END
```

```

C****************************************
        SUBROUTINE CLEARBUF(NSTC)
C****************************************
C CLEAR ALL PICTURE SEGMENTS
        DO I=1,NSTC
           CALL PICDEL(I)
        ENDDO
        RETURN
        END
```

```

C PROGRAM TO DRAW SOME SIMPLE CHEMICAL STRUCTURES INTERACTIVELY
C NR ...... NUMBER OF ROWS WHICH DISPLAY THE BASIC STRUCTURES
C NBS ...... NUMBER OF BASIC STRUCTURES IN EACH ROW
C HEIT ...... HEIGHT OF BASIC STRUCTURE
C WIDS ...... WIDTH OF BASIC STRUCTURE
C R ...... RADIUS OF CIRCLE IN BENZENE RING
        SUBROUTINE CHEMFORM(NSEG,STRC,XPS,YPS)
        PARAMETER (NR=2,NBS=12,NBST=24,MSEG=2000)
        REAL XL(NR,NBS),XR(NR,NBS),YB(NR,NBS),YT(NR,NBS)
        REAL XPS(MSEG),YPS(MSEG)
        INTEGER STRC(MSEG)
        COMMON /BSTCSIZ/HEIT,WIDS,R
        COMMON /GFEVEN/KEY,IMPKEY,IMPDAT,NSG,XCRT,YCRT,
     4                  NARGS,ARGS(80)
        CALL FRAME
        RLEN=20.0
        RHEIT=15.0
C SET THE COORDINATES RANGE FOR EACH STRUCTURE
        DO I=1,NR
           DO J=1,NBS
              XL(I,J)=(J-1)*RLEN
              XR(I,J)=J*RLEN
              YB(I,J)=(I-1)*RHEIT
              YT(I,J)=I*RHEIT
           ENDDO
        ENDDO
C PICK UP A STRUCTURE
        NSEG=0
10      CALL INDICAT(XRC,YRC,IKEY)
        GOTO (101,102,103,104,105,106,107,10),IKEY
101     DO I=1,NR
           DO J=1,NBS
              IF (XRC .GT. XL(I,J) .AND. XRC .LT. XR(I,J) .AND.
     $            YRC .GT. YB(I,J) .AND. YRC .LT. YT(I,J)) THEN
                 ID_STRC=(I-1)*NBS+J
                 GOTO 10
              ENDIF
           ENDDO
        ENDDO
C IF THE POSITION INDICATED FOR PLACING STRUC. IS INTO MENU AREA
C THEN START OVER AGAIN
102     IF (YRC .LE. 30.0) GOTO 10
C PLACE THE STRUCTURE AND START AS A NEW SEGMENT
        NSEG=NSEG+1
        XPS(NSEG)=XRC
        YPS(NSEG)=YRC
        STRC(NSEG)=ID_STRC
        CALL PICBEG(NSEG)
        CALL LINCOL(6)
        CALL MOVTO2(XRC,YRC)
        CALL DRBSTC(ID_STRC,XRC,YRC)
        CALL PICEND
        GOTO 10
103     CALL DRAG(NSEG)
        XPS(NSEG)=XCRT
        YPS(NSEG)=YCRT
        STRC(NSEG)=ID_STRC
        GOTO 10
104     CALL PICDEL(NSEG)
        GOTO 10
105     CALL CLEARBUF(NSEG)
        NSEG=0
        GOTO 10
107     CALL CHAMOD
106     CALL CLEARBUF(NSEG)
1000    RETURN
        END
```

```

C DRAW THE FRAME WITH THE DISPLAY OF BASIC STRUCTURES
        SUBROUTINE FRAME
        PARAMETER (NR=2,NBS=12)
        REAL XC(NR,NBS),YC(NR,NBS)
        CHARACTER*20 MENITM(7)
        DATA MENITM /'E ... End ','Q ... Quit',
     $               'C ... Clean ','D ... Delete ',
     $               'M ... Move ','P ... Posit',
     $               'F ... Fetch '/
        COMMON /BSTCSIZ/HEIT,WIDS,R
```

```
      XS=0.0
      YS=0.0
      XLEN=238.0
      YLEN=178.0
      CALL LINCOL(8)
      CALL MOVTO2(XS,YS)
      CALL LINBY2(0.0,YLEN)
      CALL LINBY2(XLEN,0.0)
      CALL LINBY2(0.0,-YLEN)
      CALL LINBY2(-XLEN,0.0)
      HETTT=YLEN-20.0
      CALL MOVTO2(0.0,HETTT)
      CALL LINBY2(XLEN,0.0)
C DISPLAY TITLE
      XPT=35.0
      YPT=HETTT+5.0
      CALL CHAENQ(IENQ,OWID,OHET,NENQ,AEM,AEN)
      CALL CHASIZ(5.0,5.0)
      CALL MOVTO2(XPT,YPT)
      CALL LINCOL(4)
      CALL CHASTR('Interactive Drawing of Structures*.')
      CALL CHASIZ(OWID,OHET)
C DRAW FRAMES FOR THE BASIC STRUCTURES
      CALL LINCOL(8)
      RHEIT=30.0
      CALL MOVTO2(0.0,RHEIT)
      CALL LINBY2(XLEN,0.0)
      CALL MOVTO2(0.0,RHEIT/2.0)
      CALL LINBY2(XLEN,0.0)
      RLEN=20.0
      DO I=1,12
         CALL MOVTO2(I*RLEN,RHEIT)
         CALL LINBY2(0.0,-RHEIT)
      ENDDO
C DRAW AREA FOR THE MENUE
      XST=10*RLEN
      YST=RHEIT
      CALL MOVTO2(XST,YST)
      CALL LINBY2(0.0,HETTT-RHEIT)
      CALL MOVTO2(XST,HETTT-20.0)
      CALL LINBY2(38.0,0.0)
C ASSIGN COODINATES OF CENTER FOR EACH BASIC STRUCTURE
C AND DISPLAY THEM
      CALL LINCOL(4)
      DO I=1,NR
         DO J=1,NBS
            XC(I,J)=RLEN/2.0+(J-1)*RLEN
            YC(I,J)=RHEIT/4.0+(I-1)*(RHEIT/2.0)
            ID_STRC=(I-1)*NBS+J
            CALL DRBSTC(ID_STRC,XC(I,J),YC(I,J))
         ENDDO
      ENDDO
C DISPLAY THE MENUE ON THE RIGHT HAND SIDE OF THE FRAME
      CALL LINCOL(5)
      CALL MOVTO2(XST+15,HETTT-10.0)
      CALL CHASTR('KEYS*.')
      DO I=1,7
         CALL MOVTO2(XST+5,RHEIT+I*10.0)
         CALL CHASTR(MENITM(I))
      ENDDO
      RETURN
      END

C ****************************************
      SUBROUTINE INDICAT(XRC,YRC,IKEY)
      COMMON /GFEVEN/KEY,IMPKEY,IMPDAT,NSEG,XPIC,YPIC,
    4               NARGS,ARGS(80)
      IND=1000
      XST=120.0
      YST=60.0
      CALL PICBEG(IND)
         CALL LINCOL(3)
         CALL MOVTO2(XST,YST)
         CALL MOVTO2(XST-1.0,YST)
         CALL LINBY2(2.0,0.0)
         CALL MOVTO2(XST,YST-1.0)
         CALL LINBY2(0.0,2.0)
      CALL PICEND
      CALL LINCOL(4)
      CALL DRAG(IND)
      CALL PICDEL(IND)
      IKEY=KEY
      XRC=XPIC
      YRC=YPIC
      RETURN
      END

C DRAW BASIC STRUCTURES FOR DISPLAY OR CONSTRUCTION
      SUBROUTINE DRBSTC(IDS,XANC,YANC)
      XRC=XANC
      YRC=YANC
      GOTO (201,202,203,204,205,206,207,208,209,210,211,212,
    $       213,214,215,216,217,218,219,220,221,222,223,224),IDS
201   CALL STRC1(XRC,YRC)
      GOTO 10
202   CALL STRC2(XRC,YRC)
      GOTO 10
203   CALL STRC3(XRC,YRC)
      GOTO 10
204   CALL STRC4(XRC,YRC)
      GOTO 10
205   CALL STRC5(XRC,YRC)
      GOTO 10
206   CALL STRC6(XRC,YRC)
      GOTO 10
207   CALL STRC7(XRC,YRC)
      GOTO 10
208   CALL STRC8(XRC,YRC)
      GOTO 10
209   CALL STRC9(XRC,YRC)
      GOTO 10
210   CALL STRC10(XRC,YRC)
      GOTO 10
211   CALL STRC11(XRC,YRC)
      GOTO 10
212   CALL STRC12(XRC,YRC)
      GOTO 10
213   CALL STRC13(XRC,YRC)
      GOTO 10
214   CALL STRC14(XRC,YRC)
      GOTO 10
215   CALL STRC15(XRC,YRC)
      GOTO 10
216   CALL STRC16(XRC,YRC)
      GOTO 10
217   CALL STRC17(XRC,YRC)
      GOTO 10
218   CALL STRC18(XRC,YRC)
      GOTO 10
219   CALL STRC19(XRC,YRC)
```

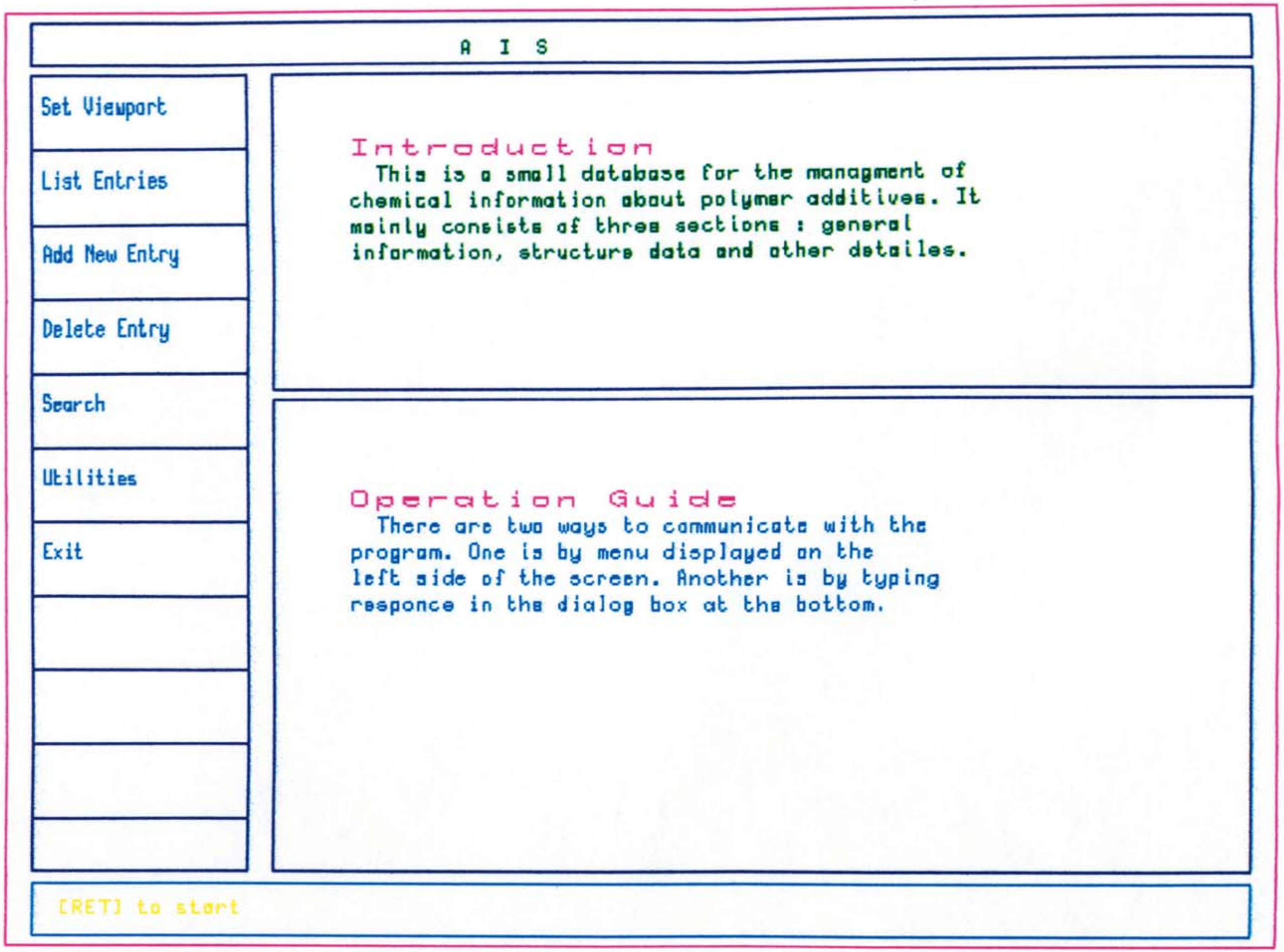
```
              GOTO 10
      220     CALL STRC20(XRC,YRC)
              GOTO 10
      221     CALL STRC21(XRC,YRC)
              GOTO 10
      222     CALL STRC22(XRC,YRC)
              GOTO 10
      223     CALL STRC23(XRC,YRC)
              GOTO 10
      224     CALL STRC24(XRC,YRC)
      10      RETURN
              END

      C SUBROUTINE TO DRAW A BENZENE RING
              SUBROUTINE STRC1(XC,YC)
              COMMON /BSTCSIZ/HEIT,WIDS,R
              X_TOP=XC
              Y_TOP=YC+HEIT/2.0
              CALL BENS1(-WIDS,-HEIT,X_TOP,Y_TOP)
              CALL BENS1(WIDS,-HEIT,X_TOP,Y_TOP)
              CALL CIRCL(XC,YC,R)
              RETURN
              END

      C DRAW CIRCLE
              SUBROUTINE CIRCL(XC,YC,R)
              NP=20
              TWPI=2.0*3.141596
              THINC=TWPI/NP
              THETA=0.0
              CALL MOVTO2(XC+R,YC)
              DO I=1,NP+1
                 X=XC+R*COS(THETA)
                 Y=YC+R*SIN(THETA)
                 CALL LINTO2(X,Y)
                 THETA=THETA+THINC
              ENDDO
              RETURN
              END

      C DRAW TWO SIDES OF A BENZENE RING
              SUBROUTINE BENS1(HMOV,VMOV,XS,YS)
              CALL MOVTO2(XS,YS)
              CALL LINBY2(0.5*HMOV,VMOV/4.0)
              CALL LINBY2(0.0,VMOV/2.0)
              CALL LINBY2(-0.5*HMOV,VMOV/4.0)
              RETURN
              END

      C DRAW TWO SIDES OF A HBENZENE RING
              SUBROUTINE BENS2(HMOV,VMOV,XS,YS)
              CALL MOVTO2(XS,YS)
              CALL LINBY2(HMOV/4.0,0.5*VMOV)
              CALL LINBY2(HMOV/2.0,0.0)
              CALL LINBY2(HMOV/4.0,-0.5*VMOV)
              RETURN
              END

      C DRAW HRIZONTAL BENZENE RING
              SUBROUTINE STRC2(XC,YC)
              COMMON /BSTCSIZ/HEIT,WIDS,R
```

```
        X_TOP=XC-HEIT/2.0
        Y_TOP=YC
        CALL BENS2(HEIT,WIDS,X_TOP,Y_TOP)
        CALL BENS2(HEIT,-WIDS,X_TOP,Y_TOP)
        CALL CIRCL(XC,YC,R)
        RETURN
        END

C DRAW NITROXYL STRUCTURE
        SUBROUTINE STRC3(XC,YC)
        COMMON /BSTCSIZ/HEIT,WIDS,R
        CALL BSD3(WIDS,HEIT,XC,YC)
        CALL BSD3(WIDS,-HEIT,XC,YC)
        RETURN
        END

C DRAW BACKWARD NITROXYL
        SUBROUTINE STRC4(XC,YC)
        COMMON /BSTCSIZ/HEIT,WIDS,R
        CALL BSD3(-WIDS,HEIT,XC,YC)
        CALL BSD3(-WIDS,-HEIT,XC,YC)
        RETURN
        END

C BASIC DRAWING ROUTINE
        SUBROUTINE BSD3(HMOV,VMOV,XC,YC)
        XS=XC+HMOV/2.0
        YS=YC
        CALL MOVTO2(XS,YS)
        CALL LINBY2(-HMOV/4.0,VMOV/3.0)
        CALL LINBY2(-HMOV/2.0,0.0)
        CALL POSPIC(XN,YN,ZN)
        CALL LINBY2(-HMOV/8.0,-VMOV/8.0)
        CALL MOVTO2(XN,YN)
        CALL LINBY2(-HMOV/8.0,VMOV/6.0)
        CALL MOVTO2(XN,YN)
        CALL LINBY2(HMOV/8.0,VMOV/6.0)
        NC=ICHAR('N')
        XS=XC-HMOV/3.0
        YS=YC
        CALL DRACHA(NC,XS,YS)
        RETURN
        END

C DRAW STRUCTURE OCO
        SUBROUTINE STRC5(XC,YC)
        COMMON /BSTCSIZ/HEIT,WIDS,R
        CLEN=HEIT/4.0
        NC=ICHAR('O')
        XS=XC-1.0*CLEN
        YS=YC
        CALL DRACHA(NC,XS,YS)
        NC=ICHAR('C')
        XS=XC
        CALL DRACHA(NC,XS,YS)
        NC=ICHAR('O')
        XS=XC+1.0*CLEN
        CALL DRACHA(NC,XS,YS)
        RETURN
        END

```

```
C DRAW STRUCTURE COO
        SUBROUTINE STRC6(XC,YC)
        COMMON /BSTCSIZ/HEIT,WIDS,R
        CLEN=HEIT/4.0
        NC=ICHAR('C')
        XS=XC-1.0*CLEN
        YS=YC
        CALL DRACHA(NC,XS,YS)
        NC=ICHAR('O')
        XS=XC
        CALL DRACHA(NC,XS,YS)
        XS=XC+1.0*CLEN
        CALL DRACHA(NC,XS,YS)
        RETURN
        END

C DRAW STRUCTURE tBu
        SUBROUTINE STRC7(XC,YC)
        COMMON /BSTCSIZ/HEIT,WIDS,R
        CLEN=HEIT/4.0
        NC=ICHAR('t')
        XS=XC-1.0*CLEN
        YS=YC
        CALL DRACHA(NC,XS,YS)
        NC=ICHAR('B')
        XS=XC
        CALL DRACHA(NC,XS,YS)
        NC=ICHAR('u')
        XS=XC+1.0*CLEN
        CALL DRACHA(NC,XS,YS)
        RETURN
        END

C DRAW CHARACTER C
        SUBROUTINE STRC8(XC,YC)
        NC=ICHAR('C')
        CALL DRACHA(NC,XC,YC)
        RETURN
        END

C DRAW CHARACTER H
        SUBROUTINE STRC9(XC,YC)
        NC=ICHAR('H')
        CALL DRACHA(NC,XC,YC)
        RETURN
        END

C DRAW CH2
        SUBROUTINE STRC10(XC,YC)
        COMMON /BSTCSIZ/HEIT,WIDS,R
        CLEN=HEIT/4.0
        NCH=ICHAR('2')
        CALL BSD5(CLEN,NCH,XC,YC)
        RETURN
        END

C DRAW CH3
        SUBROUTINE STRC11(XC,YC)
        COMMON /BSTCSIZ/HEIT,WIDS,R
        CLEN=HEIT/4.0
        NCH=ICHAR('3')
        CALL BSD5(CLEN,NCH,XC,YC)
        RETURN
        END

C DRAW DOT .
        SUBROUTINE STRC12(XC,YC)
        CALL MOVTO2(XC,YC)
        CALL DOT(0.3)
        RETURN
        END

C DRAW COMBINED CHARACTER
        SUBROUTINE BSD5(CHLEN,NC,XC,YC)
        CLEN=CHLEN
        XS=XC-1.0*CLEN
        YS=YC
        CALL STRC8(XS,YS)
        XS=XC
        CALL STRC9(XS,YS)
        XS=XC+1.0*CLEN
        YS=YC-CLEN/2.0
        CALL DRACHA(NC,XS,YS)
        RETURN
        END

C DRAW A VERICAL LINE
        SUBROUTINE STRC13(XC,YC)
        COMMON /BSTCSIZ/HEIT,WIDS,R
        XS=XC
        YS=YC+HEIT/4.0
        CALL MOVTO2(XS,YS)
        CALL LINBY2(0.0,-HEIT/2.0)
        RETURN
        END

C DRAW A HORIZONTAL LINE
        SUBROUTINE STRC14(XC,YC)
        COMMON /BSTCSIZ/HEIT,WIDS,R
        XS=XC-HEIT/4.0
        YS=YC
        CALL MOVTO2(XS,YS)
        CALL LINBY2(HEIT/2.0,0.0)
        RETURN
        END

C DRAW A DOUBLE VERTICAL LINE
        SUBROUTINE STRC15(XC,YC)
        COMMON /BSTCSIZ/HEIT,WIDS,R
        DIS=HEIT/8.0
        XNC=XC-DIS/2.0
        YNC=YC
        CALL STRC13(XNC,YNC)
        XNC=XNC+DIS
        CALL STRC13(XNC,YNC)
        RETURN
        END

C DRAW A DOUBLE HORIZONTAL LINE
        SUBROUTINE STRC16(XC,YC)
        COMMON /BSTCSIZ/HEIT,WIDS,R
```

```
      DIS=HEIT/8.0
      YNC=YC+DIS/2.0
      XNC=XC
      CALL STRC14(XNC,YNC)
      YNC=YNC-DIS
      CALL STRC14(XNC,YNC)
      RETURN
      END

C DRAW CHARACTER
      SUBROUTINE DRACHA(NCHAR,XC,YC)
      COMMON /BSTCSIZ/HEIT,WIDS,R
      CALL CHAENQ(IT,OSX,OSY,NI,AI,AN)
      CSX=HEIT/4.0
      CSY=HEIT/4.0
      CALL CHASIZ(CSX,CSY)
      XS=XC-CSX/4.0
      YS=YC-CSY/2.0
      CALL MOVTO2(XS,YS)
      CALL CHAASC(NCHAR)
      CALL CHASIZ(OSX,OSY)
      RETURN
      END

C DRAW LEFT INCLINED LINE
      SUBROUTINE STRC17(XC,YC)
      COMMON /BSTCSIZ/HEIT,WIDS,R
      CALL BSD4(WIDS,HEIT,XC,YC)
      RETURN
      END

C DRAW RIGHT INCLINED LINE
      SUBROUTINE STRC18(XC,YC)
      COMMON /BSTCSIZ/HEIT,WIDS,R
      CALL BSD4(WIDS,-HEIT,XC,YC)
      RETURN
      END

C DRAW  INCLINED LINE
      SUBROUTINE BSD4(RLEN,RHEIT,XC,YC)
      HMOV=RLEN/2.0
      VMOV=RHEIT/4.0
      XS=XC-HMOV/2.0
      YS=YC-VMOV/2.0
      CALL MOVTO2(XS,YS)
      CALL LINBY2(HMOV,VMOV)
      RETURN
      END

C DRAW CHARACTER N
      SUBROUTINE STRC19(XC,YC)
      NC=ICHAR('N')
      CALL DRACHA(NC,XC,YC)
      RETURN
      END

C DRAW CHARACTER O
      SUBROUTINE STRC20(XC,YC)
      NC=ICHAR('O')
      CALL DRACHA(NC,XC,YC)
      RETURN
      END

C DRAW CHARACTER P
      SUBROUTINE STRC21(XC,YC)
      NC=ICHAR('P')
      CALL DRACHA(NC,XC,YC)
      RETURN
      END

C DRAW CHARACTER S
      SUBROUTINE STRC22(XC,YC)
      NC=ICHAR('S')
      CALL DRACHA(NC,XC,YC)
      RETURN
      END

C DRAW CHARACTER X
      SUBROUTINE STRC23(XC,YC)
      NC=ICHAR('X')
      CALL DRACHA(NC,XC,YC)
      RETURN
      END

C DRAW CHARACTER Me
      SUBROUTINE STRC24(XC,YC)
      COMMON /BSTCSIZ/HEIT,WIDS,R
      CLEN=HEIT/4.0
      NC=ICHAR('M')
      XS=XC-CLEN/2.0
      YS=YC
      CALL DRACHA(NC,XS,YS)
      NC=ICHAR('e')
      XS=XC+CLEN/2.0
      CALL DRACHA(NC,XS,YS)
      RETURN
      END
```

AppendixE Samples of Database Running

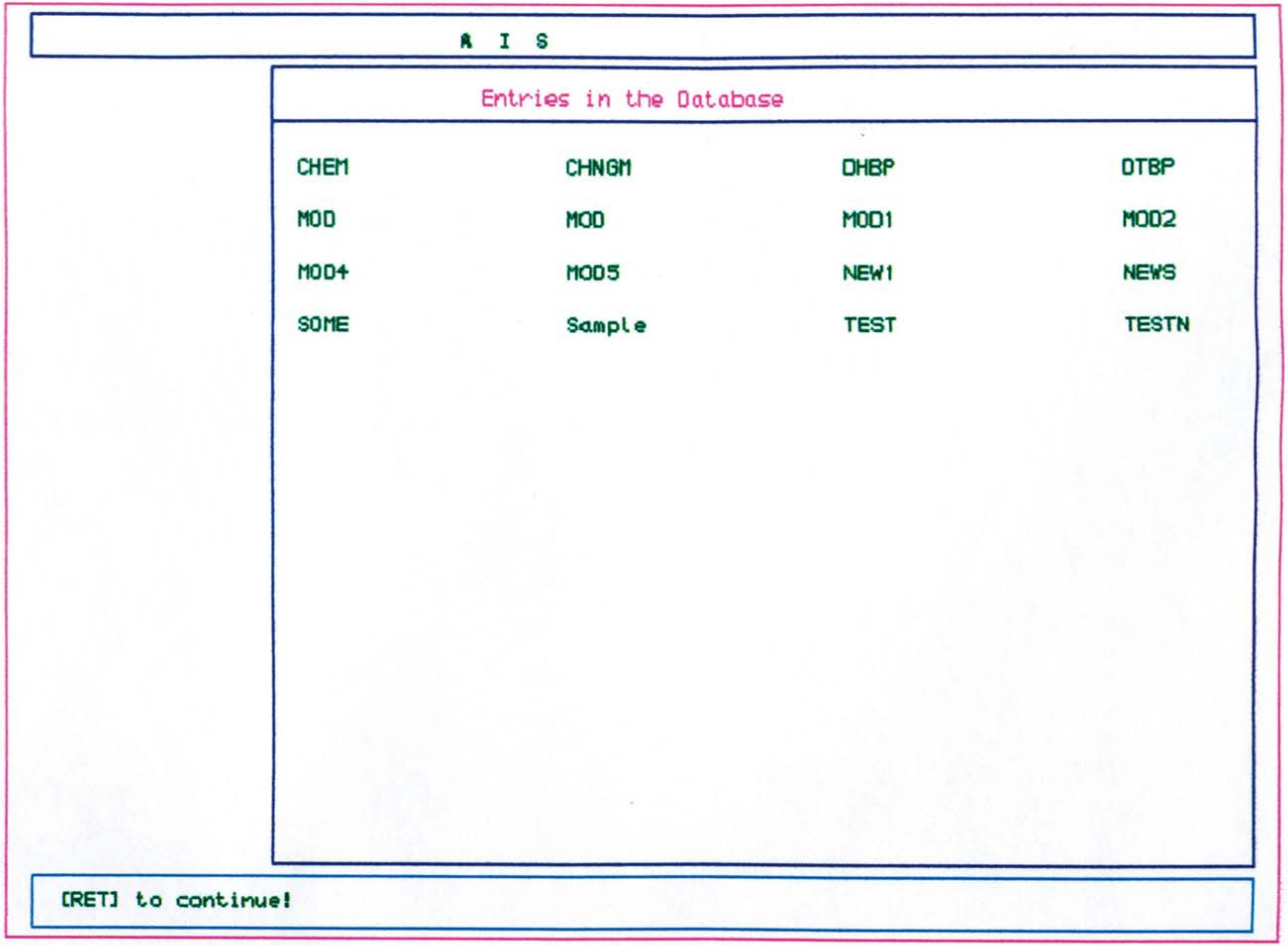


Introduction

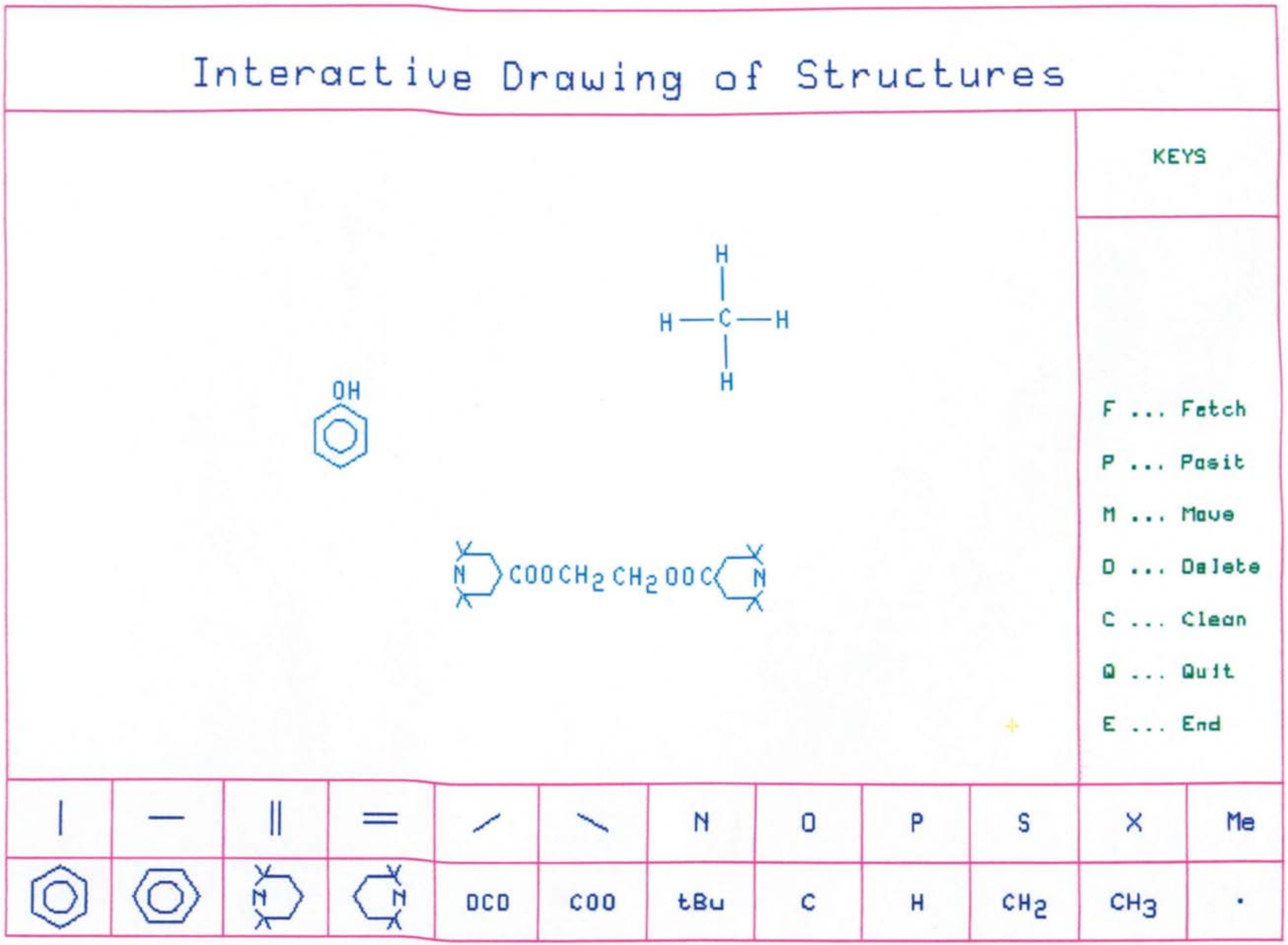
This is a small database for the managment of chemical information about polymer additives. It mainly consists of three sections : general information, structure data and other detailes.

Operation Guide

There are two ways to communicate with the program. One is by menu displayed on the left side of the screen. Another is by typing responce in the dialog box at the bottom.



```
                              A I S

                    Entries in the Database

CHEM           CHNGM          DHBP           DTBP

MOD            MOD            MOD1           MOD2

MOD4           MOD5           NEW1           NEWS

SOME           Sample         TEST           TESTN


[RET] to continue!
```



Interactive Drawing of Structures

KEYS

F ... Fetch

P ... Posit

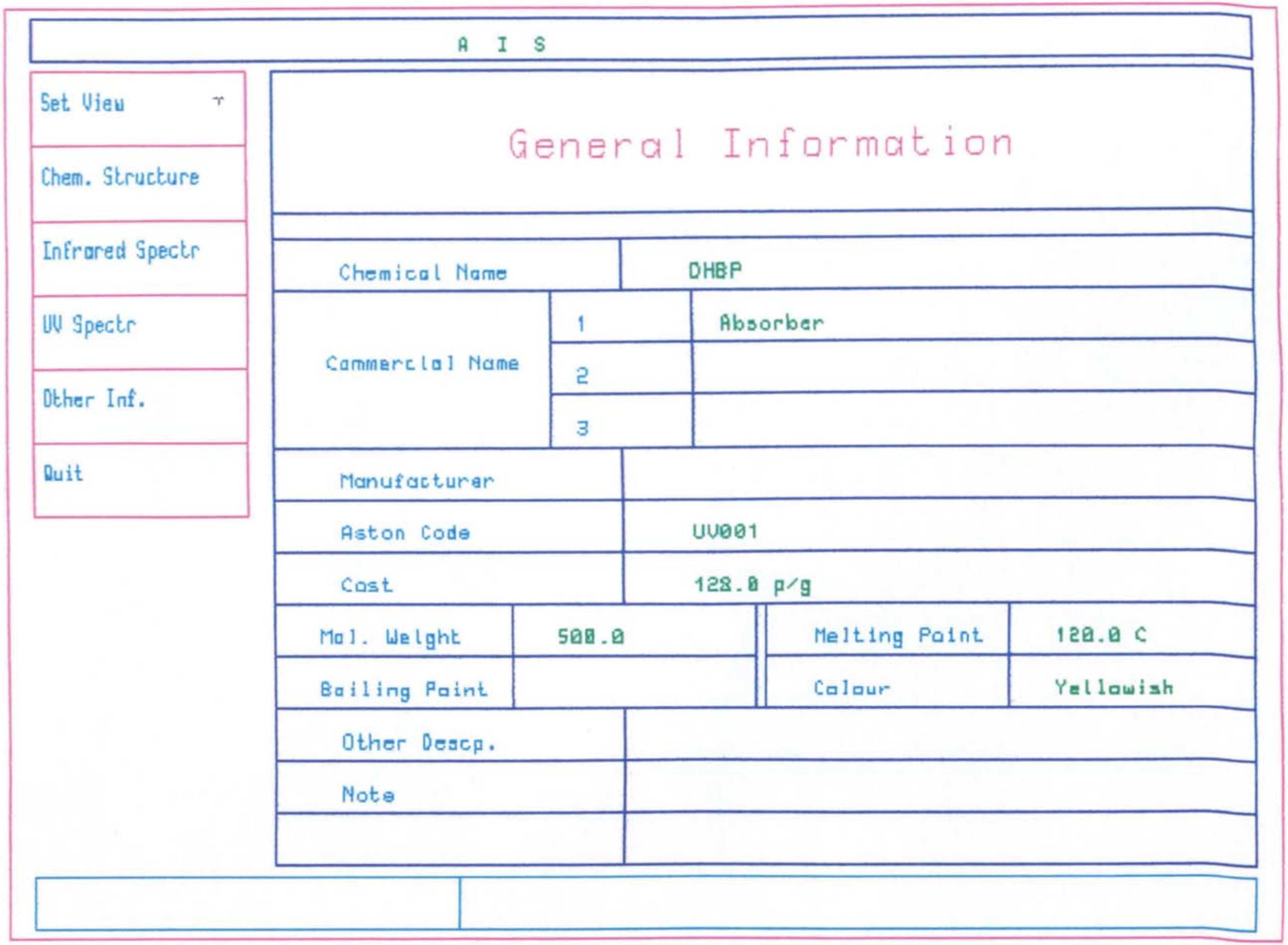
M ... Move

D ... Delete

C ... Clean

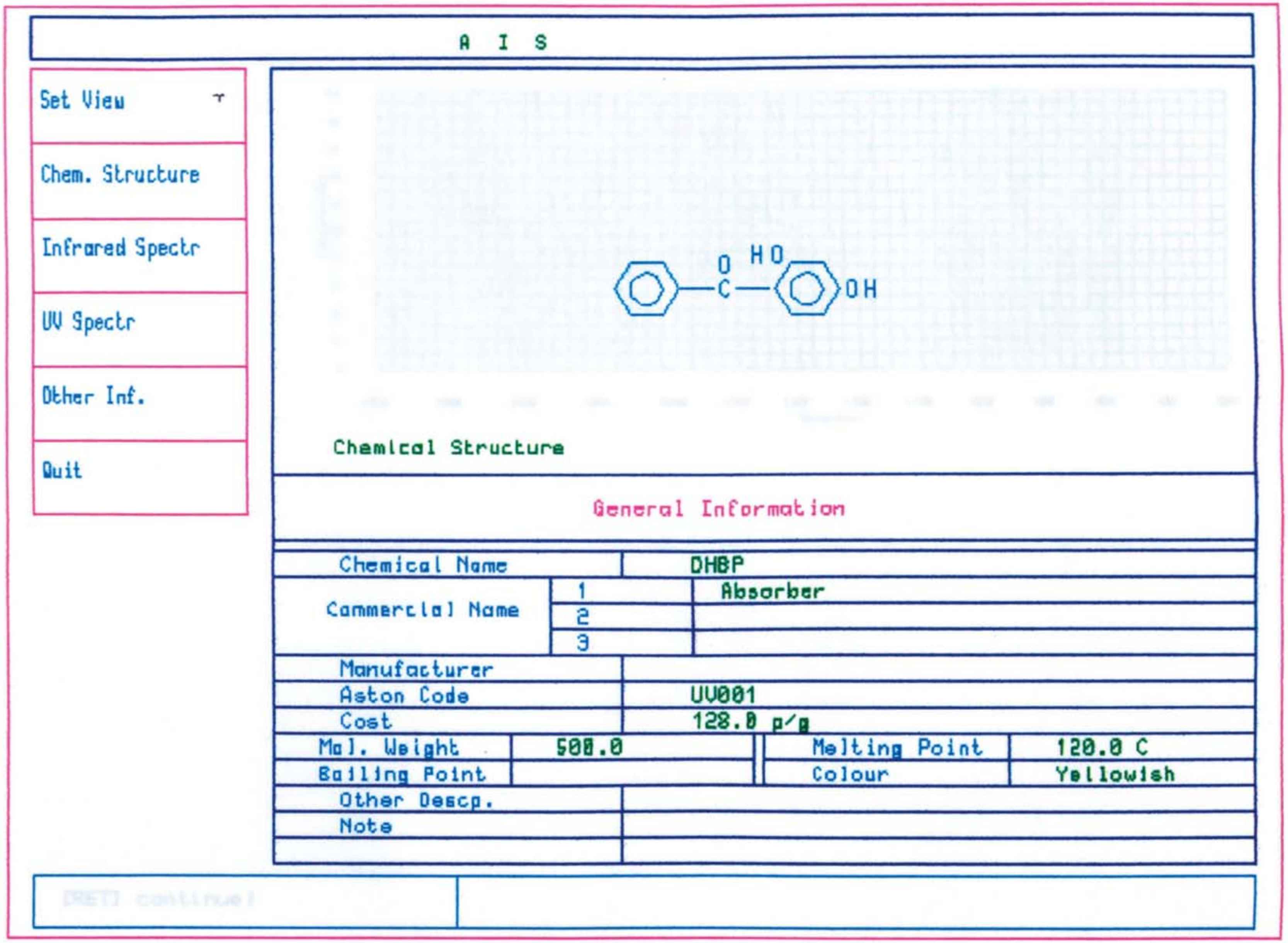
Q ... Quit

E ... End



General Information

Chemical Name	DHBP		
Commercial Name	1	Absorber	
	2		
	3		
Manufacturer			
Aston Code	UV001		
Cost	128.0 p/g		
Mol. Weight	500.0	Melting Point	120.0 C
Boiling Point		Colour	Yellowish
Other Descp.			
Note			



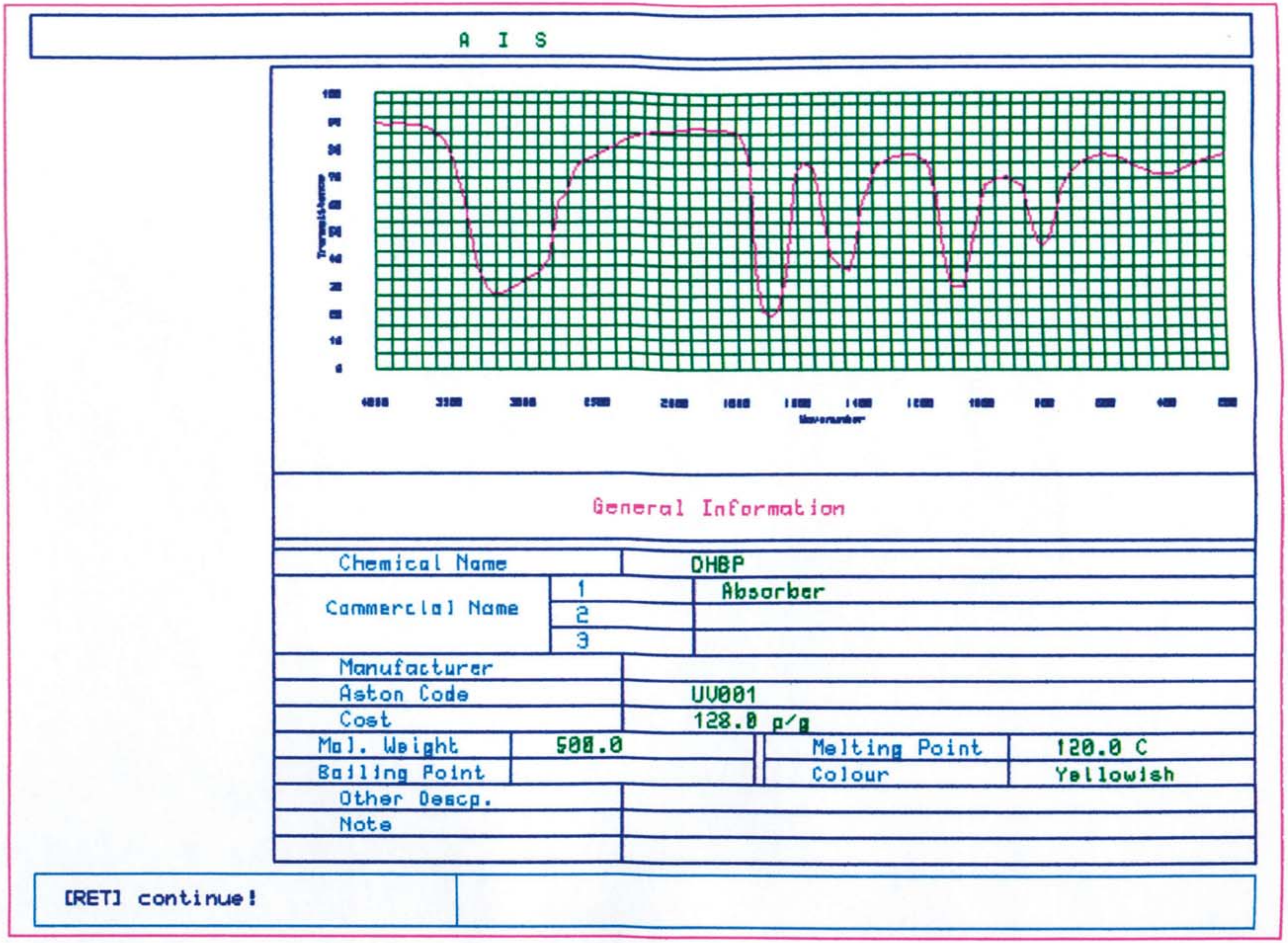
A I S

Chemical Structure

General Information

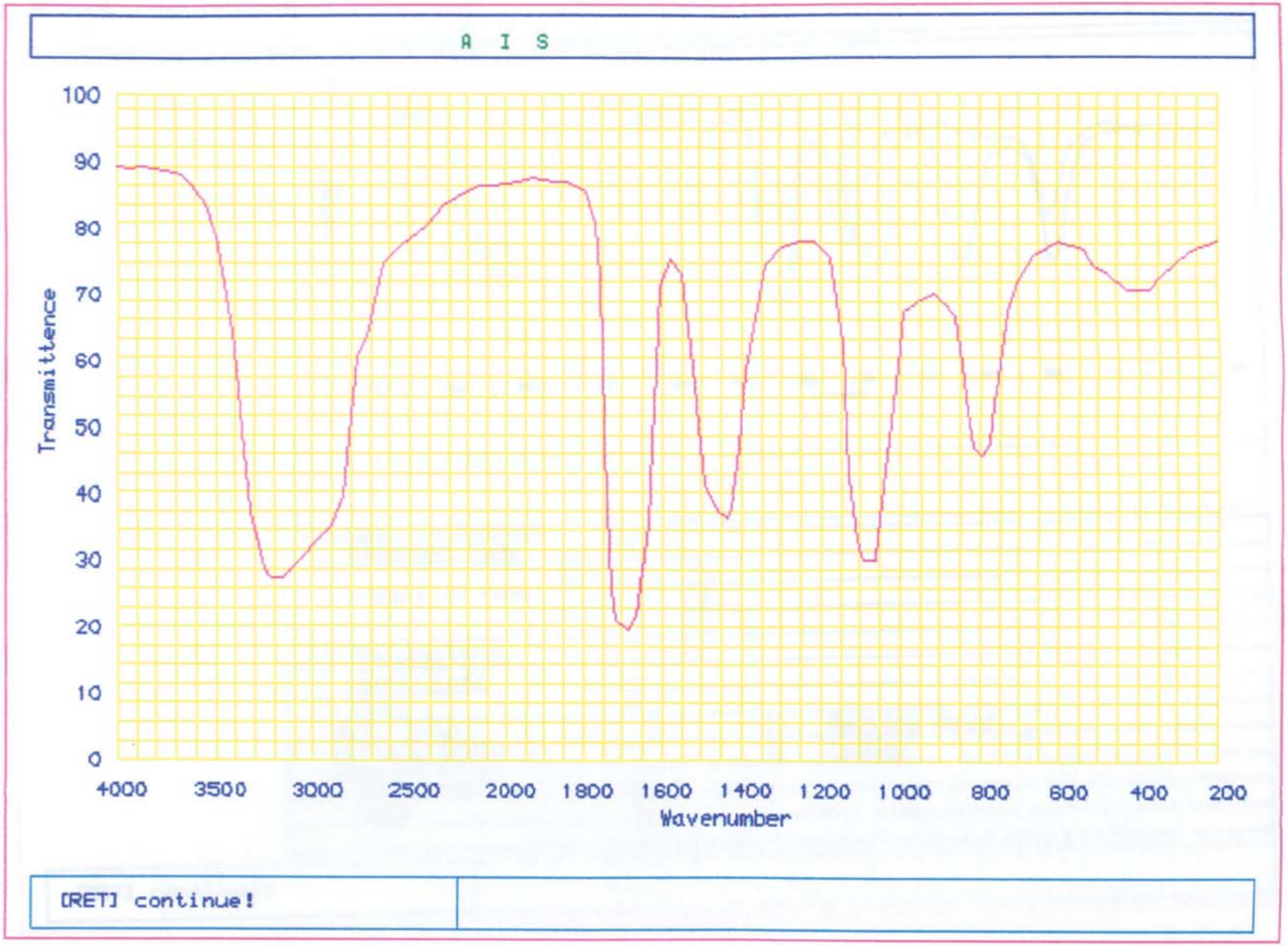
Chemical Name		DHBP		
Cammercial Name	1	Absorber		
	2			
	3			
Manufacturer				
Aston Code		UV001		
Cost		128.0 p/g		
Mol. Weight	500.0		Melting Point	120.0 C
Boiling Point			Colour	Yellowish
Other Descp.				
Note				

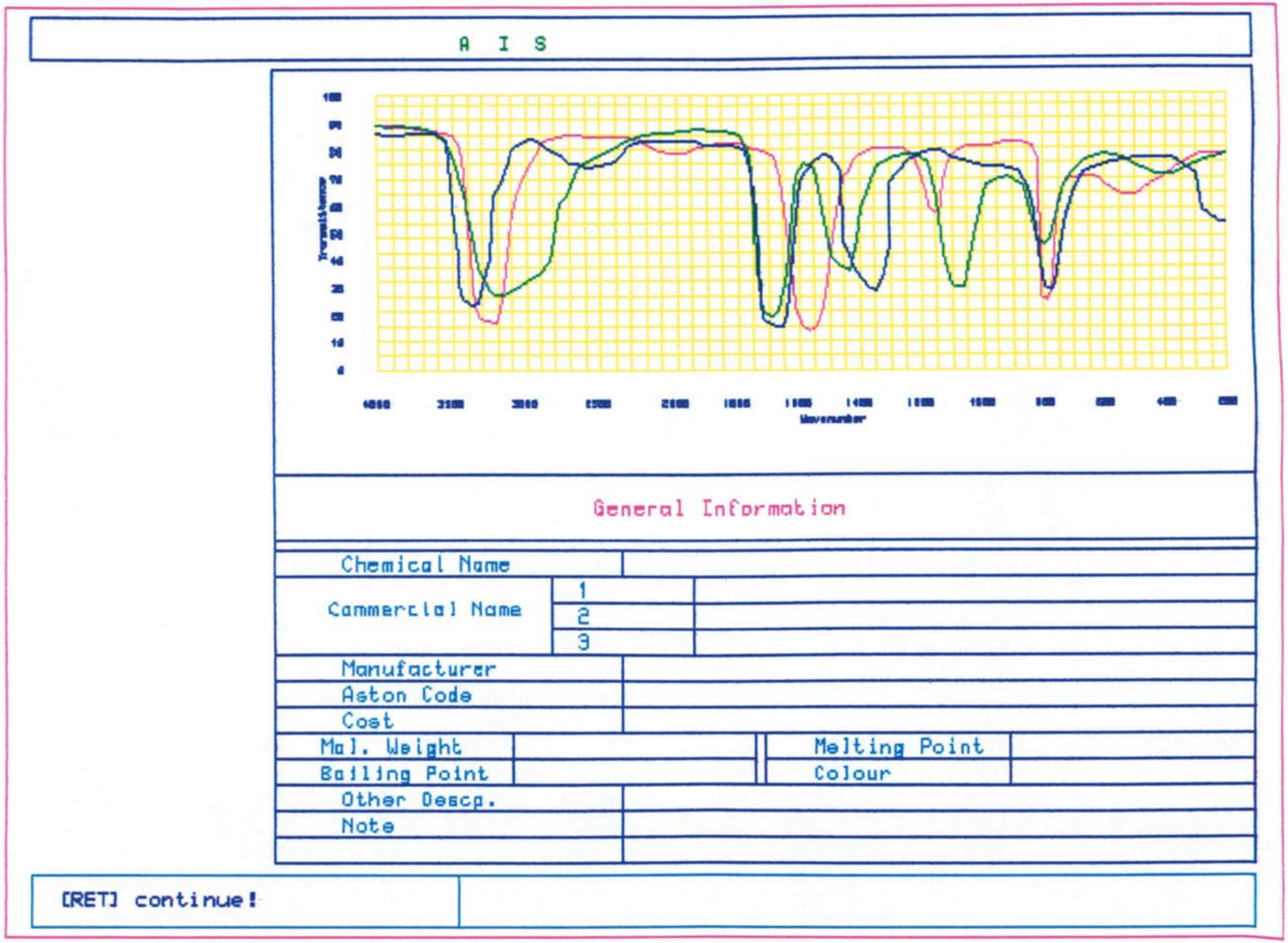
[RET] continue!



General Information

Chemical Name		DHBP		
Commercial Name	1	Absorber		
	2			
	3			
Manufacturer				
Aston Code		UV001		
Cost		128.0 p/g		
Mol. Weight	500.0		Melting Point	120.0 C
Boiling Point			Colour	Yellowish
Other Descp.				
Note				


A I S
Transmittence
100
90
80
70
60
50
40
30
20
10
0
4000
3500
3000
2500
2000
1800
1600
1400
1200
1000
800
600
400
200
Wavenumber
[RET] continue!

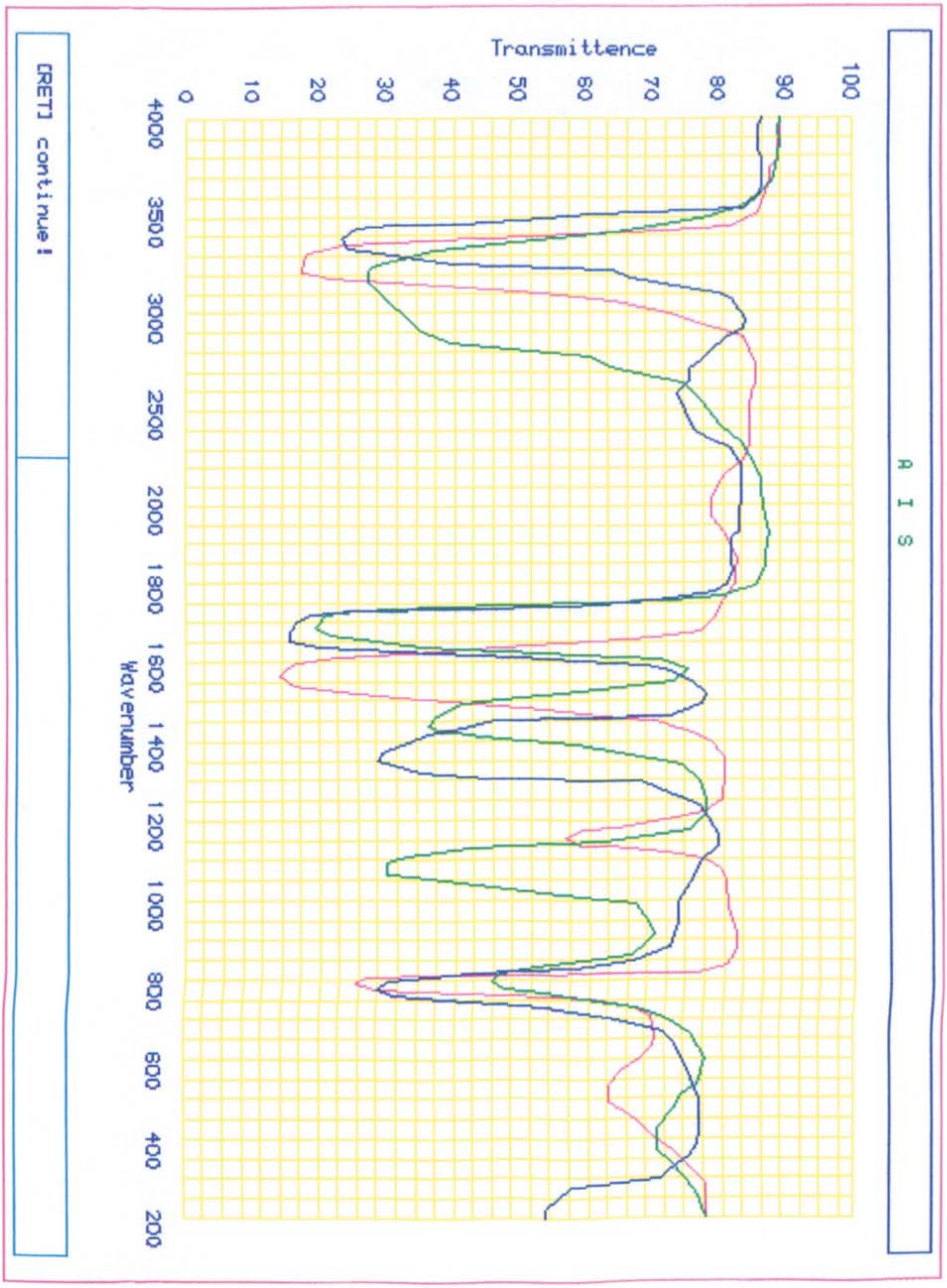


A I S

General Information

Chemical Name			
Cammercial Name	1		
	2		
	3		
Manufacturer			
Aston Code			
Cost			
Mol. Weight		Melting Point	
Boiling Point		Colour	
Other Descp.			
Note			

[RET] continue!



A I S
Transmittence
100
90
80
70
60
50
40
30
20
10
0
4000
3500
3000
2500
2000
1800
1600
1400
1200
1000
800
600
400
200
Wavenumber
[RET] continue!